UNIMOTION

Manual
Closed Loop Stepping System Manual
STDL PN Profinet

1   Introduction

The STDL-xxx-C-PN drives series are equipped with a PROFINET IO field bus and controls the stepper motors with vector technique.

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

All models have the encoder input, which allows the closed-loop motor control thus eliminating stall problems and improving efficiency.

The supported operative modes allow the control of the motor in position, speed and torque. Numerous homing modes are also available.

The free software *STDL Workspace*, running under the Windows platform (Windows 7 or newer), assists the commissioning of the product. For the connection to the PC, the STIL device interface is required.

1.1   Series

The STDL-PN series features two models:

Model	Power supply [V DC]	Motor current [A]	I/O
STDL-040-C-PN	20 – 50	1.0 – 4.5	8 Digital Inputs 3 Digital Outputs 1 Analog Input 1 Encoder Input A, B, I 1 Absolute Encoder Input SSI
STDL-100-C-PN *(coming soon)*		2.0 – 10.0	

1.2   Terms, symbol 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 STDL-xxx-C-PN implies the models STDL-040-C-PN and STDL-100-C-PN.

The terms manual and document have the same meaning, moreover, the words drive, device and product always refer to the STDL-PN 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 the cause of serious damage to people, animals and things.

It draws 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, modes, etc., are always provided in English to avoid confusion or doubts in the interpretation.

To describe the type of data of the registers, parameters, objects, etc., it makes use of abbreviations. The following table describes the meaning besides the range of value allowed for each type:

Abbreviation	Bits	Description	Min.	Max.
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.

Other abbreviations used:

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

1.3   Documents

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

The information contained herein replace any previously issued document.

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 UNIMOTION.

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

UNIMOTION is a registered trademark.

1.4   Contents of the pack

The device is supplied with all connectors and ready to be mounted on the DIN rail.

Technical documentation and software can be downloaded from the UNIMOTION website (see product info).

1.5   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 electronics@unimotion.eu. 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.

A T T E N T I O N
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 for 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 drive housing inaccessible when the drive is powered on. The user must also consider that, because of the capacitors

inside the drive, 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 may be considerably longer.

During an operation, the product generates heat that can raise the temperature of certain parts (the heat sink for example, but not only) to values that 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 drive can generate strong accelerations, with high motor torque. It is therefore essential to never touch the mechanical parts with the drive powered on. The user must provide the application so that this condition is always granted.

Because of an incorrect wiring, incorrect configuration or else, the drive can command to the motor unexpected movements. Before supplying the drive, 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 supply (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, when installation is completed, it is important to execute a complete setting to work test.

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 drive can start sparks and fire. The housing and the components placed nearby the drive 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 must not be used in life support 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 by UNIMOTION.

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

UNIMOTION reserves the right to make changes without prior notice to the products including design, technical specification, manufacturing process and functionality. UNIMOTION expressly declines any responsibility for any damage, whether direct or indirect, arising from the use of these products. The user who disagrees with the *user conditions* of the products, should not use them.

1.6 Warranty

UNIMOTION warrants the products described in this manual against defects in materials or workmanship for 12 months. 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 UNIMOTION. This warranty does not apply if the product's label has been removed or modified.

UNIMOTION 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 is stated herein, should not use or install the software.

2 Installation

The STDL-xxx-C-PN series drives are components. The user is responsible for 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 required to fully understand the features, the setting parameters and the instructions given herein.

The user must apply all the laws and specific rules of the Country and/or application in which the product is used.

The installation must be performed by experienced staff and after having read and understood the instructions included herein.

2.1 Inspection

Verify that the drive is not damaged, the package is intact and all accessories are included.

Furthermore, control that the drive code corresponds to the ordered one, eventual special and customized version included. In case of problems please address to the product's vendor.

2.2 Mechanical Installation

The drive is designed to be mounted vertically on a 35 mm DIN rail.

To mount the drive on the DIN-rail, firstly insert the bracket on the top on the back of the drive, over the top of the DIN-rail, keeping the drive slightly inclined as shown in the picture, then push the drive downward to engage the lower section of the rail. To verify the correct engagement of the drive, try and pull it slightly upward to control that it is fixed in position.

First insert the bracket on the top on the DIN rail

Push the drive downward to engage the lower section of the rail
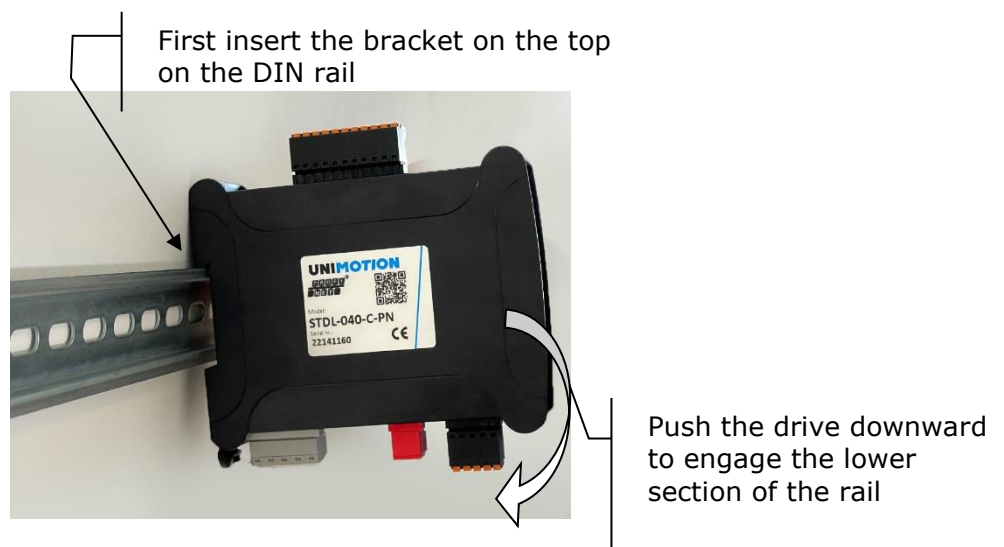
To remove the drive from the DIN-rail, insert a small flat-bladed screwdriver into the hook on the bottom, on the back of the drive. Push the hook downward and pull the drive upward, slightly rotating it, releasing it from the DIN-rail as shown in the picture.


Insert a small flat bladed screwdriver into the hook

Push the screwdriver slightly downward and pull the drive upward, releasing it from the DIN rail

The heat generated by the drive while operating must dissipate towards the surrounding air.

According to the drive calibration and to the running cycle, the space can be also substantially reduced without compromising the correct working of the drive.
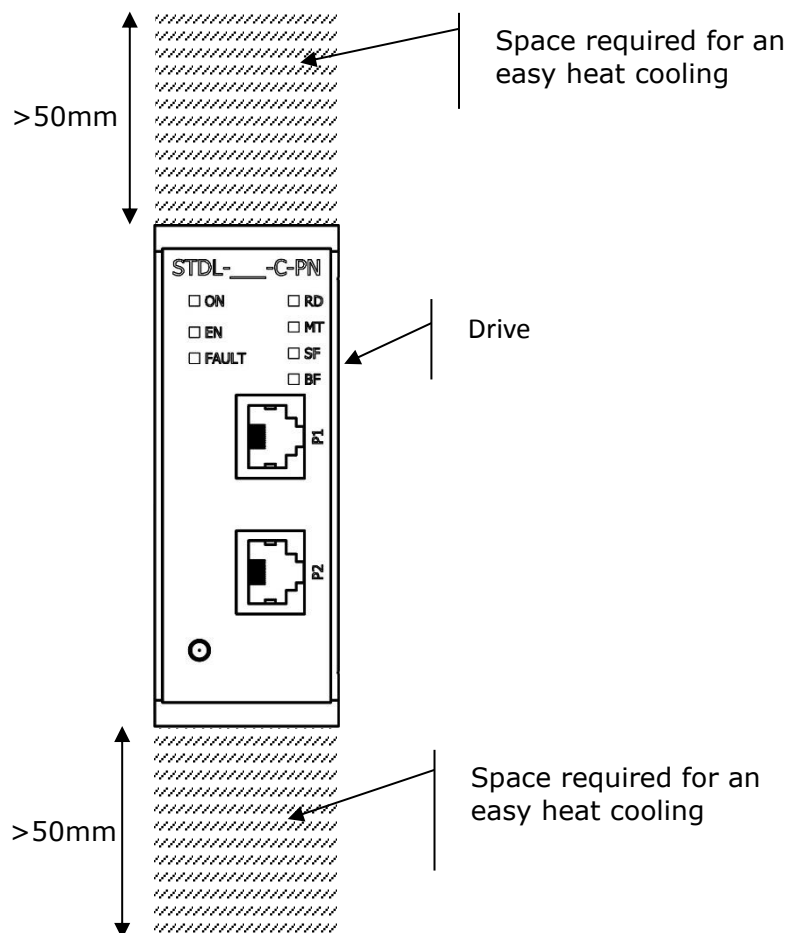
2.3 Chassis setting

To contain the electromagnetic emissions and to better shield the drive, it is essential to give particular attention to the setting of the chassis.

The drive must be placed inside a metal case, preferably iron made, capable of successfully shielding the electromagnetic emissions. The case must be electrically placed to the ground as better described further below.

The filter must be correctly placed in series to the main supply. The ideal position is on the chassis edge to have a short wiring coming out from the main supply. If, on the contrary, the main supply cables run inside the chassis, they can be impacted by electromagnetic interferences making them ineffective or worse, reducing the filter's efficiency.

The filter output earth must be connected to the metallic body of the chassis. It is important that the connection is of short length and made with a large section and low inductance conductor. The point of connection between ground coming from the filter and the chassis constitutes the star center to which all other ground/earth components must be connected. Moreover, the filter's metallic body must be electrically placed in contact with the case.
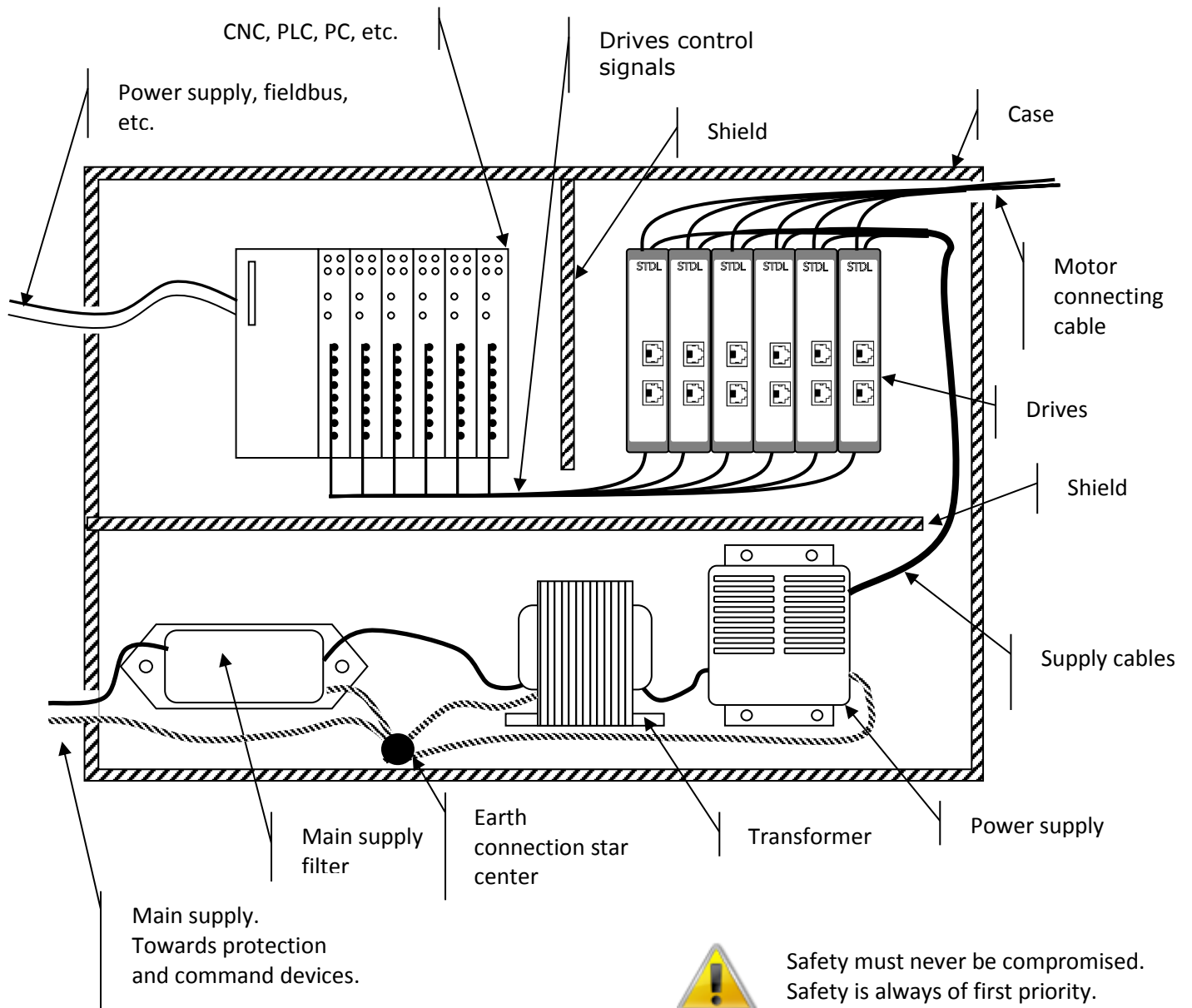
The transformer must be placed close to the filter and must have a shield between the primary and the secondary windings. The shield must be ground connected in the star center obtained inside the case. Furthermore, the transformers metallic body must be electrically connected to the case.

It is advisable to install the power supply near the transformer. If the power supply is of switching type and therefore without a transformer, it is recommended to place it immediately at the output of the mains filter. The power supply earth (usually the negate pole of the filter capacitor) must be connected to the ground in the star center obtained inside the case.

The drive position must be chosen in order so that the motor cables can immediately come out from the case without running long distances inside the case itself. Furthermore, the motor cables must be kept as far as possible from any other conductor.

The numerical control device, PLC or other, which generates the driving signals of the drive must be as far as possible from the drives and the power supply group. Moreover, the signals wiring must be distant from the power supply and motor cables. When the distance from the numerical control device and the drive and/or the power supply is reduced, there must be one or more shielding walls, electrically connected to the case.

The following figure shows a possible setting of the chassis.

CNC, PLC, PC, etc.

Drives control signals

Power supply, fieldbus, etc.

Shield

Case

STDL STDL STDL STDL STDL STDL

Motor connecting cable

Drives

Shield

Supply cables

Main supply filter

Earth connection star center

Transformer

Power supply

Main supply.
Towards protection and command devices.

Safety must never be compromised.
Safety is always of first priority.

2.4   Connectors

The STDL-xxx-C-PN drives series has four connectors that are common to the whole family. The common connectors are reserved for the power supply, the motor connection, the digital I/O and PROFINET bus.

Connector	Function
CN1	Power supply
CN2	Motor
CN3	Digital I/Os
CN4	Motor Encoder
P1, P2	PROFINET Bus

2.4.1   CN1 – Power Supply

⚠️ Reversing polarity connection can permanently damage the drive as well as exceeding the Vpbrk voltage limit.

⚠️ Do not supply the power to the drive before the wiring is complete.

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

CN1 – Power Supply	
Pin	Description
1	+VDC, positive DC supply voltage
2	GND, negative DC supply voltage

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

Model	Symbol	Description	Unit	Value		
				Min	Typ	Max
STDL-xxx-C-PN	Vp	DC supply voltage	V	20		50
	Vpbrk	Voltage causing permanent damage	V	-0.5		60

The 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	Typ	Max
STDL-xxx-C-PN	Vpl	Under voltage protection intervention threshold	V		18	
	Vph	Over voltage protection intervention threshold	V		52	

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

Model	Voltage (V)	Capacity (µF)
STDL-040-C-PN	63	470
STDL-100-C-PN	63	1000

To connect the power supply, the drive and the eventual local capacitor it is necessary to use a conductor with cross-section adequate to the current setting of the drive (anyway, for security reasons it is better to use the maximum output current of the drive). The following table resumes the cable section suggested for each drive:

Model	Cross-section (mm^2)
STDL-040-C-PN	1
STDL-100-C-PN	1.5

The power supply cable can be installed together with the ones which connect the drive to the motor. 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 drive with voltage values near to the agreed maximum 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 tolerance during its sizing so that, in case of mains supply and load fluctuations, the voltage remains however within the allowed operation 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, ensure the correct working and fulfill each safety requirements. To determine the power supply output voltage, it must be considered the maximum mains voltage 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 drive model.

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

The following formula provides a rough indication:

$$Pw = 5 + (1.1 * (Iph * Iph * Rph)) + |((Vrpm * Tnm) / 7)|$$

Where Pw is the power required by the power supply expressed in Watt (W), Iph is the phase current delivered to the motor expressed in effective Ampere (Arms), Rph is the motor phase resistance expressed in ohm (Ω), Vrpm is the rotation speed in rev/minute (RPM) and finally Tnm is the resistant torque of the load expressed in newton/meter (Nm). If, for example, the motor has a phase resistance of 1.5 Ω and is supplied with a current of 3 Arms, and works at a speed of 500 rpm with a load of 2 Nm, the power supply should deliver a power of about 163W ((5 + (1.1*3*3*1.5) + |(500 * 2 / 7)|). Note that during the acceleration and deceleration of the load or at the enabling of the motor the absorption may be higher. For this reason, it is important that the power supply has output capacitors suitable to the size of the chosen drive (see further on).

To limit the peak of the current when enabling the motor, the drive has a function able to gradually increase the phase current up to the nominal value. The ramp time can be set through the parameter 4412_h *CurrentEnableRamp_MTNSTP*.

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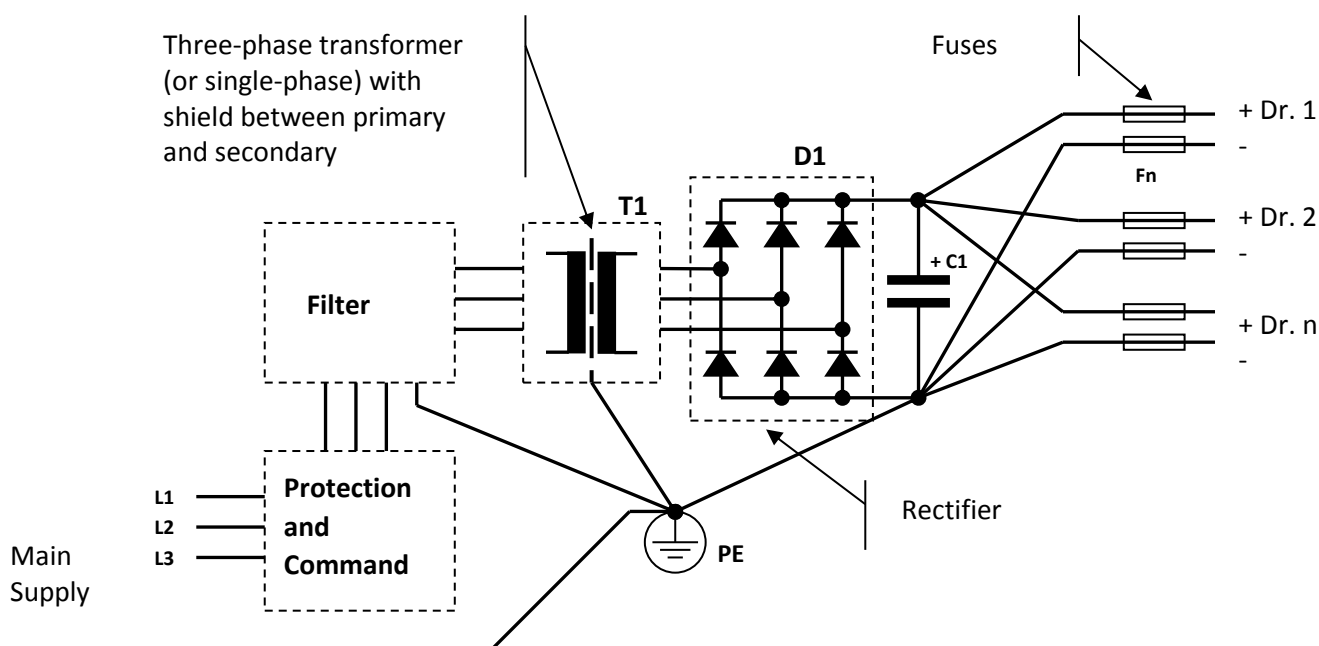
Note that the earth connection must be star-like, where the earth connections of the various components terminate into one single point electrically connected to the electrical cabinet and the earth of the plant.

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 a short circuit or malfunctioning. It is also obligatory to place a fuse on each drive'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 drive and/or present on the electrical network.

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 for advice and to find the suitable filters for your application.

The following table shows the characteristic values of the main components of the power supply according to the number of drives present in the application. The calculations also consider an oscillation of the mains supply voltage +10/−20 %.

Model	Fuses Fn (A T)	Number of drives	SecondaryT1 (V AC)	Power T1 (VA)	Current D1 (Arms)	Voltage C1 (V DC)	Capacity C1 (μF)
STDL-040-C-PN	6,3	1	32	125	25	63	3300
		2		250	25		4700
		3		375	25		5600
		4..5		600	25		8200
		6..8		900	35		10000
STDL-100-C-PN	12,5	1	32	250	25	63	4700
		2		500	25		6800
		3		750	25		8200
		4..5		1100	35		10000
		6..8		1800	50		15000

The capacity values suggested for C1 can also be obtained by placing more capacitors in parallel amongst them. Eventual approximation 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 supply voltage available during the installation of the application. The transformer must have a shield between primary and secondary windings which must be connected to the earth by a short and not inductive connection. The secondary winding voltage is meant without load, with the primary winding supplied at the nominal voltage.

The rectifier, besides supporting the maximum current required by the drive, must be able to tolerate the surge current during the C1 capacitor charge. 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 to be 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 at least 2.

In the configuration with more than one 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 motor's 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 use a single-phase power supply, which is not recommended when the required power is greater than 800 W.

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

2.4.2 CN2 – Motor

The drive regulates the motors phase current with modulating the power supply voltage with PWM technique. If the cable is long, it is important to use such with a low parasitic capacitance between the conductors and between the conductors and the shield.

CN2 – Motor	
Pin	Description
1	A-, negative output phase A
2	A+, positive output phase A
3	B+, positive output phase B
4	B-, negative output phase B
5	Not connected
Note: Inverting the phase A+ with A-, or the phase B+ with B-, the motor rotation direction is inverted.	

The cable cross-section can be dimensioned according to the drive current setting, anyway, it is suggested to choose a cable suitable to withstand the maximum current deliverable from the chosen drive.

It is also advised to connect the motor to the drive with a cable with a length inferior to 10 m. For cables with a greater length, the cable size must be increased to counterbalance the voltage drop.

The cable connecting the drive to the motor can be installed together with the power supply cable, but it must be kept separate from the signal ones.

If you have difficulties in overcoming the electromagnetic compatibility test it is possible to place in series to each phase an inductor with a value between 10 µH and 100 µH, and with current adequate to the set phase current. The inductor must be placed directly at the drive output.

2.4.3   CN3 - I/O Control Signals
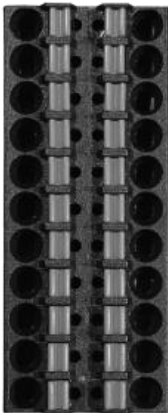
The connection with the control signals is through a 22 contacts removable spring terminal block. The terminal block can be easily oriented through the key, as shown in the picture below.

To insert the cable into the connector, press with a small screwdriver the orange presser and simultaneously insert the wire into the near hole, then release the presser. It is suggested to remove the wire covering to about 8 mm.

Pin 1 → [connector image] ← Pin 2

Key
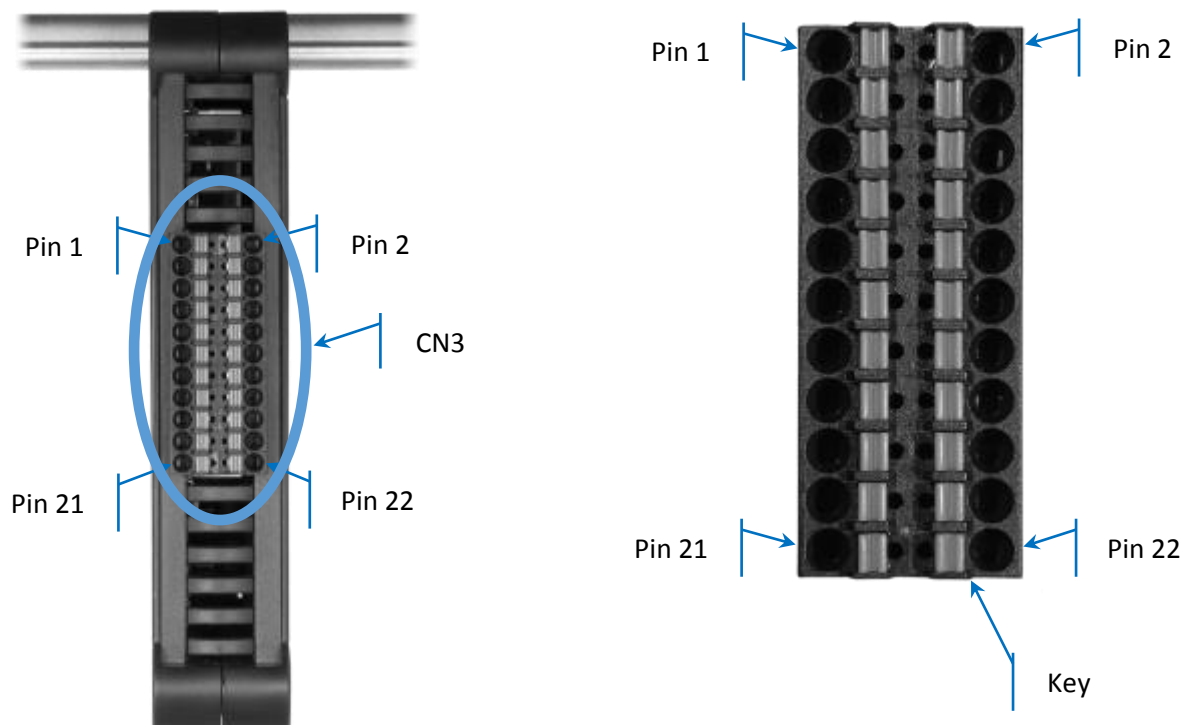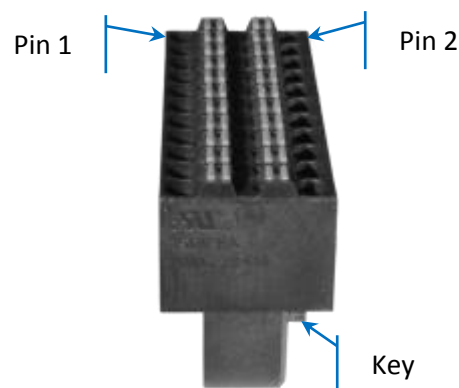
The following table shows the assignment of the signals to the various terminal pins:

CN3 - I/O Control Signals				
Description	Pin		Pin	Description
+24 V Auxiliary Power Supply	1		2	0 V (GND) Auxiliary Power Supply
DI0+ (Digital Input 0)	3		4	DI0- (Digital input 0)
DI1+ (Digital Input 1)	5		6	DI1- (Digital input 1)
DI234COM (common DI2...DI4)	7		8	DI2 (Digital input 2)
DI3 (Digital input 3)	9		10	DI4 (Digital input 4)
DO0+ (Digital Output 0)	11		12	DO0- (Digital Output 0)
DO1+ (Digital Output 1)	13		14	DO1- (Digital Output 1)
AI0+ (Analog Input 0)	15		16	AI0 GND (Analog Input 0)
DO2+ (Digital Output 2)	17		18	DO2- (Digital Output 2)
DI567COM (commonDI5..DI7)	19		20	DI5 (Digital input 5)
DI6 (Digital input 6)	21		22	DI7 (Digital input 7)

2.4.3.1 Auxiliary Power Supply

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

Keeping the logic section of the drive supplied, the signals, the fieldbus and the encoder are powered. The encoder allows keeping track of the motors' position change even if disconnected and moved manually.

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

Symbol	Description	Unit	Value		
			Min	Typ	Max
V24	Auxiliary Power Supply DC voltage	V	20		35
V24brk	Permanent damage voltage	V	-0.5		40

2.4.3.2 Digital inputs

All the digital inputs are optocoupled and have a current regulation circuit which allows to apply voltages between 3 V_{DC} and 28 V_{DC} without the need to add 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 (0 V) 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 negative signal).

The other inputs are organized in two groups of 3 inputs each with a common and can be used both in NPN and PNP configuration. To use a group in NPN configuration simply connect the common of the group to the positive reference and each input to the output of the master controller (PLC, CNC, etc.), while in case of PNP connection connect the common of the group to the GND and each input to the output of the master controller.

The described NPN and PNP connections are shown in the below images:
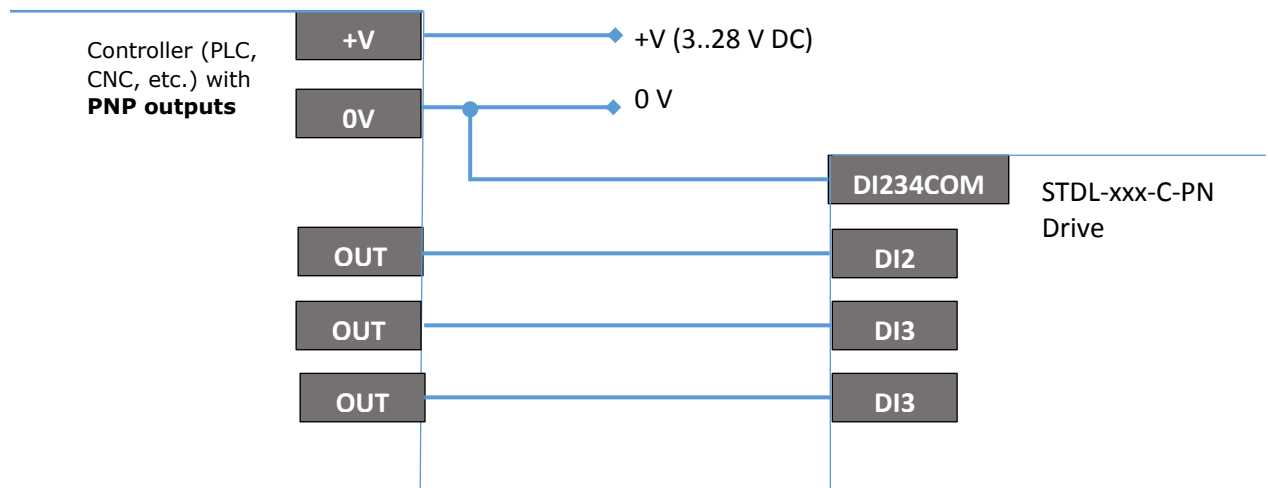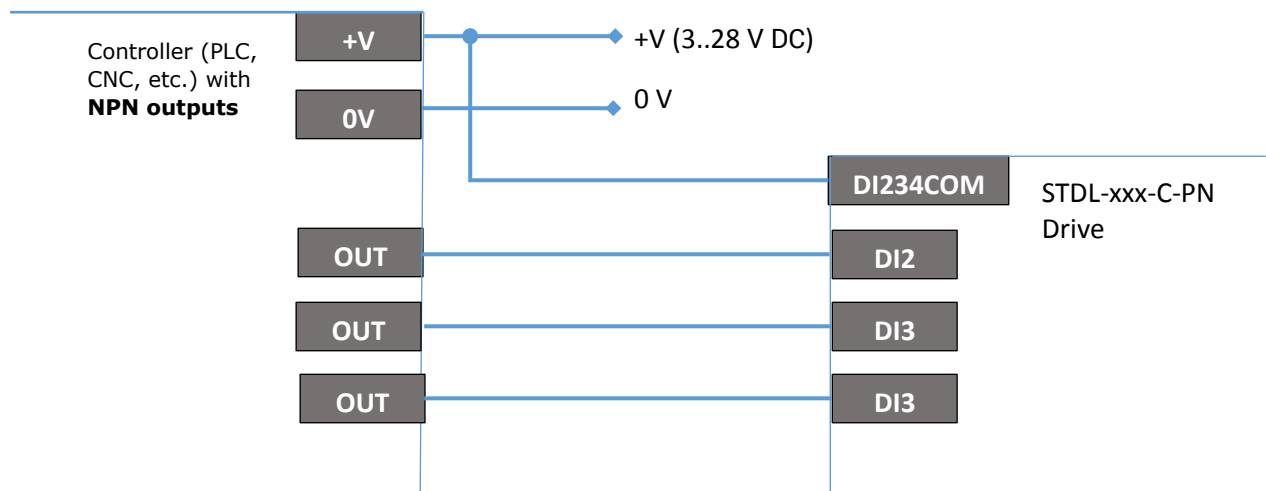
Most of the inputs does not have a dedicated functionality and their use is defined during the configuration of the drive, however, some of the inputs have specific functions that cannot be assigned to other inputs.

The following table shows the inputs with dedicated functions.

Input	Function	
DI0	DI0	Generic input, 100 kHz
	---	Reserved input for future implementation (e.g., counter)
DI1	DI1	Generic input, 100 kHz
	---	Reserved input for future implementation (e.g., counter)
DI2	DI2	Generic input
DI3	DI3	Generic input
DI4	DI4	Generic input
DI5	DI5	Generic input
DI6	DI6	Generic input
DI7	DI7	Generic input

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

Symbol	Description	Unit	Value		
			Min	Typ	Max
Vdi	Active input voltage	V DC	3		
Vdioff	Inactive input voltage	V DC			1
Vdibrk	Digital inputs breakdown voltage	V DC	-30		+30
Idi	Current absorbed by the digital inputs (24 V DC)	mA		5	

2.4.3.3   Digital Outputs

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

For brake wiring, please refer to page 29.

All the outputs are optocoupled and have both connections (+ and −), therefore they can be freely used in NPN and PNP configuration. 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 to add an external recirculation diode.

The following table shows the electrical characteristics of the digital outputs:

Symbol	Description	Unit	Value		
			Min	Typ	Max
Vdo	Digital output operating voltage	V DC	1		30
Vdobrk	Digital output breakdown voltage	V DC	-0.5		37
Vdoz	Zener diode voltage placed in parallel to each output	V DC	37	39	42
Ido	Digital output available current	mA			80
Idobrk	Digital output breakdown current	mA	120		
Pwdo	Digital output dissipate power	mW			400

To use an output in NPN configuration simply connect the − to GND and the + to the input of the master controller (PLC, CNC, etc.), while in case of PNP connection connect the + to the positive reference and the − to the input of the control system.

The described NPN and PNP connections are shown in the below images:
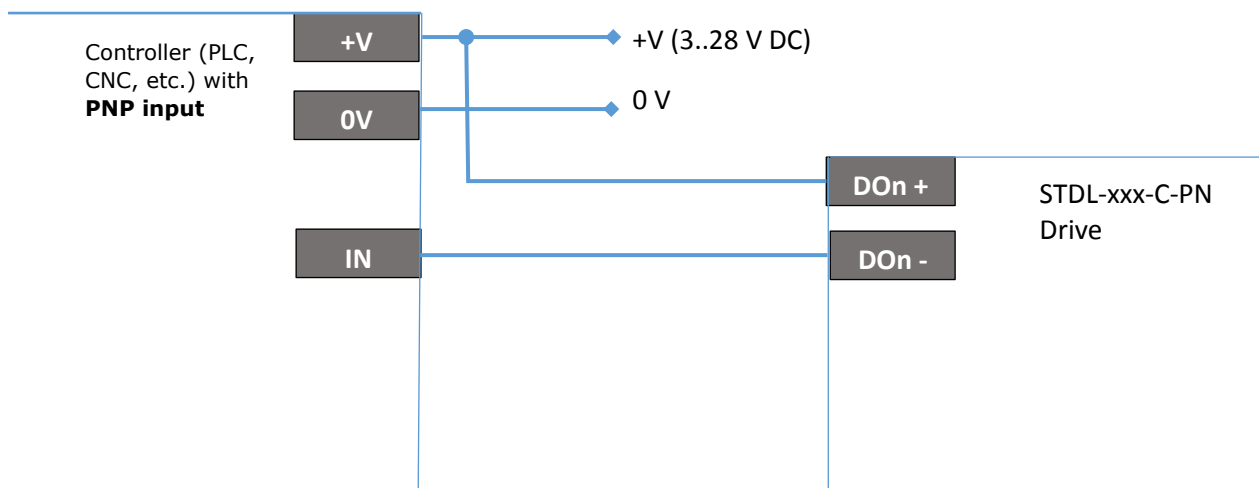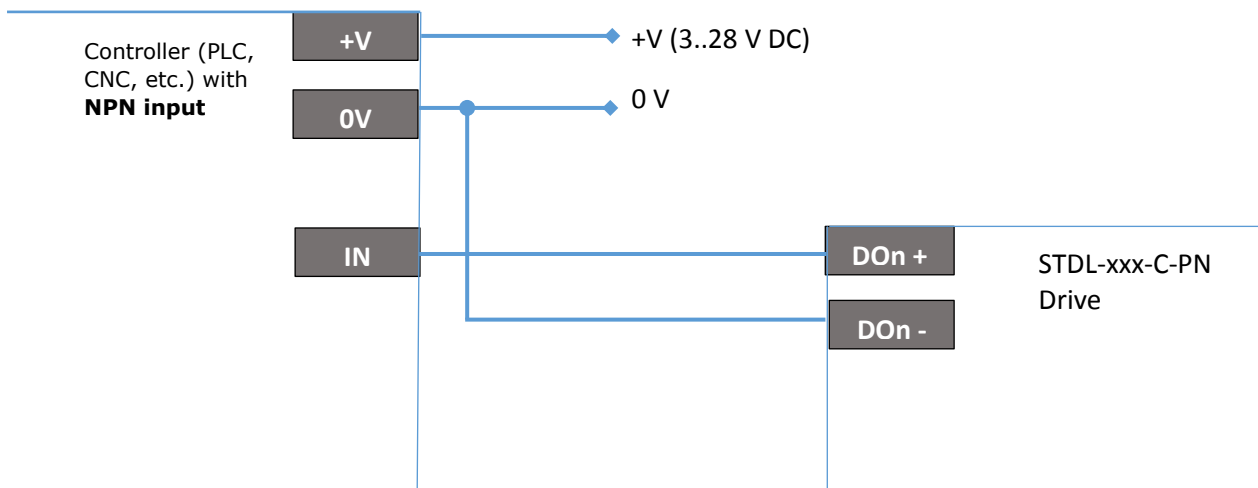
Controller (PLC, CNC, etc.) with **NPN input**

+V → +V (3..28 V DC)

0V → 0 V

IN → DOn +

DOn - → STDL-xxx-C-PN Drive

The brake cannot be connected directly to digital output of the drive!

Brake must be controlled via an additional relay with externally supplied power.

CN3 DOn+ — VCC

Relay

CN3 DOn- — NC | NO → Brake+

Brake-

GND
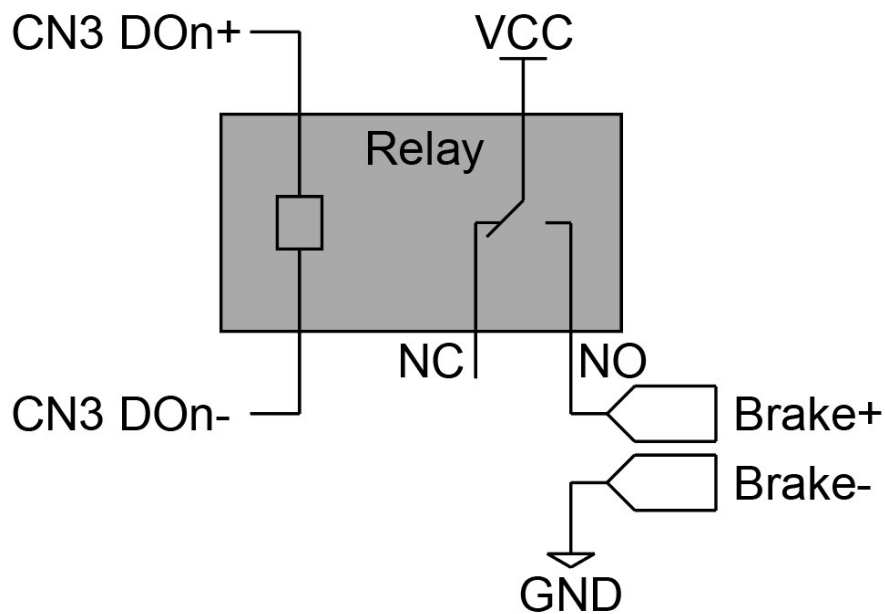
2.4.3.4 Analog inputs

The analog input is able to measure voltage between -10 V and +10 V.

In order to drive the analog input from a control device (PLC, CNC, etc.) simply connect respectively the AIN0+ and AIN0 GND signals of the drive to the output voltage of the control device and to its ground reference. Please take note that the AIN0 GND signal is internally connected to the drive power ground.

Example of connection of the analog input:
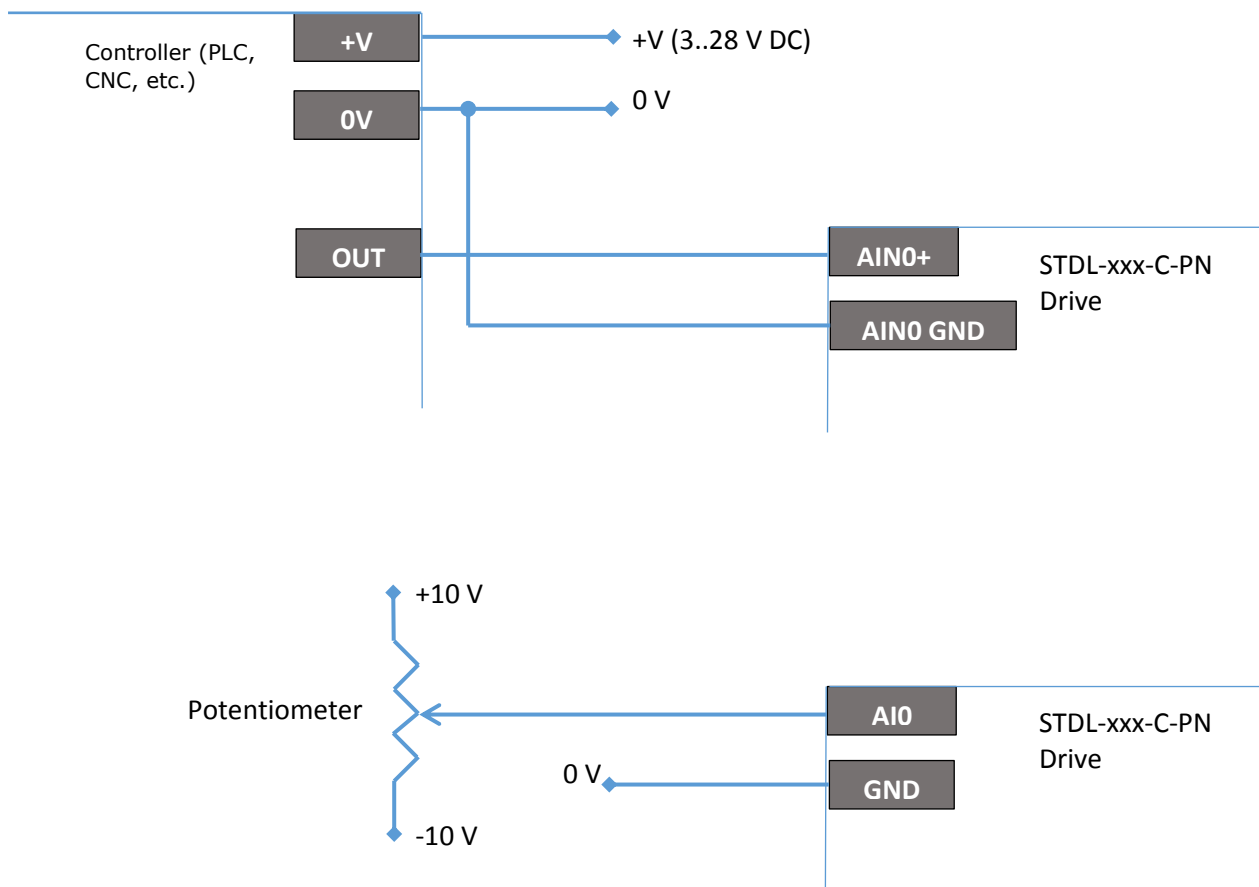
The following table shows the electrical characteristics of the analog input:

Symbol	Description	Unit	Value		
			Min	Typ	Max
Vai	Analog input operating voltage	V DC	-10.2		+10.2
Vaibrk	Analog input breakdown voltage	V DC	-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	
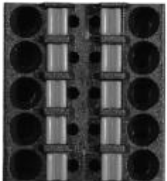
2.4.4 CN4 – Motor Encoder

The 10 contacts removable spring terminal block CN4 is used for connecting the incremental encoder and the encoder with synchronous serial interface (SSI - Synchronous Serial Interface). The terminal block can be easily oriented through the key, as shown in the picture below.

To insert the cable into the connector, press with a small screwdriver on the orange presser and simultaneously insert the wire into the near hole, then release the presser. It is suggested to remove the wire covering to about 8 mm.
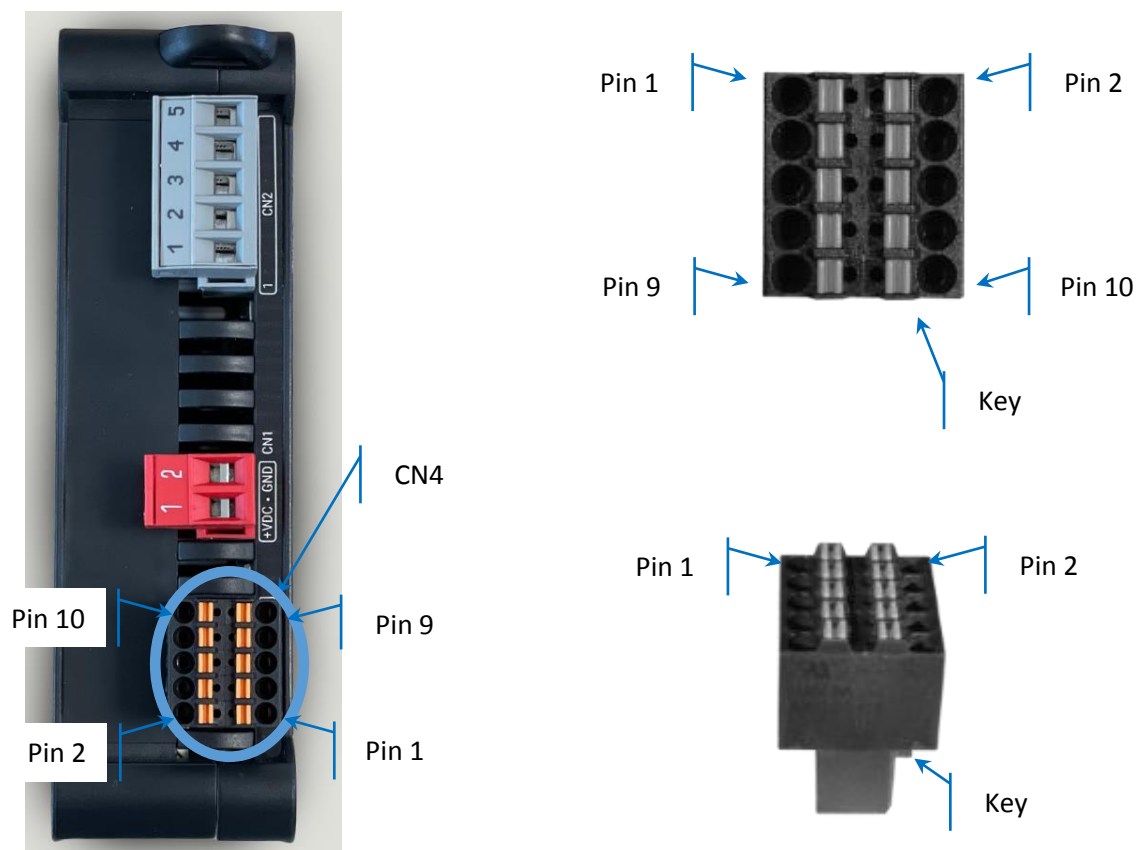
The following table shows the assignment of the signals to the various terminal pins:

CN4 – Motor Encoder and SSI Encoder				
Description	Pin		Pin	Description
+V Encoder Power Supply	1		2	**0V** (GND) Encoder Power Supply
A+ Incremental Encoder	3		4	**A-** Incremental Encoder
B+ Incremental Encoder	5		6	**B-** Incremental Encoder
I+ (Index) Incremental Encoder / **Data+** SSI Encoder	7		8	**I-** (Index) Incremental Encoder / **Data-** SSI Encoder
Clock+ SSI Encoder	9		10	**Clock-** SSI Encoder

The *I (Index)* and *Data* signals must be connected one as an alternative to the other. It follows that when using the SSI encoder, the Index signal of the incremental encoder cannot be used.

For the connection between the drive and the encoder, it is suggested to use a shielded cable, having care to connect the shield on the Pin2 together with the 0 V reference.

To power the Encoder, the drive supplies a voltage of +5 V with a current of 100 mA suitable for the most encoders, however it is also possible to

connect encoders with a different supply voltage, provided that they are externally supplied.

2.4.4.1 Incremental Encoder

It is possible to use any incremental encoder with or without Index (also called zero mark) provided that it has a resolution within the configuration's (register 4332$_h$ *CPR_ENCMTR*).

The signals inputs A, B and I are Line Driver type and usually they do not require terminating resistors. The drive internal circuits also allow the connection of other signals types, as shown in the table below:

Encoder output signals types	Encoder Signal	Drive Signal	Notes
Line Driver	A+	A+	
	A-	A-	
	B+	B+	
	B-	B-	
	I+	I+	
	I-	I-	
TTL/CMOS	A	A+	The inputs A-, B- and I- remain disconnected.
	B	B+	
	I	I+	
Open Collector	A	A+	The inputs A-, B- and I- remain disconnected.
	B	B+	
	I	I+	
Push-Pull	A	A+	⚠️ **ATTENTION, when using an encoder supplied with voltage higher than 5 V with push-pull outputs, it is important to insert in series to each signal a diode (1N4148 for example) with the cathode facing the encoder and the anode connected to the drive, otherwise the drive itself could be damaged.**
	B	B+	
	I	I+	

2.4.4.2 SSI Encoder

It is possible to connect encoders with SSI interface with binary coding and data frame length between 8 and 31 bits. If the frame is longer, the bits after 31 will not be read.

The *Clock* and *Data* signals are of Line Driver type and normally do not require termination resistance.

The following table shows the main features of the SSI interface implemented inside the drive.

Symbol	Description	Unit	Value		
			Min	Typ	Max
SSIclk	Clock frequency	MHz		1.0	
SSIfb	Frame length	bit	8		31
SSImtb	Number of bits for full revolutions	bit	0		24
SSIstb	Number of bits for single revolution	bit	8		31
SSIcd	Position coding			Binary	

2.4.5 PROFINET Bus
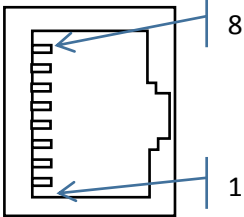
The PROFINET Bus connector has two RJ45 sockets and is used for connection to the PROFINET fieldbus.

☺ Inside the drive there is an IRT switch and it is therefore possible to connect several drives in cascade without the need for external devices.
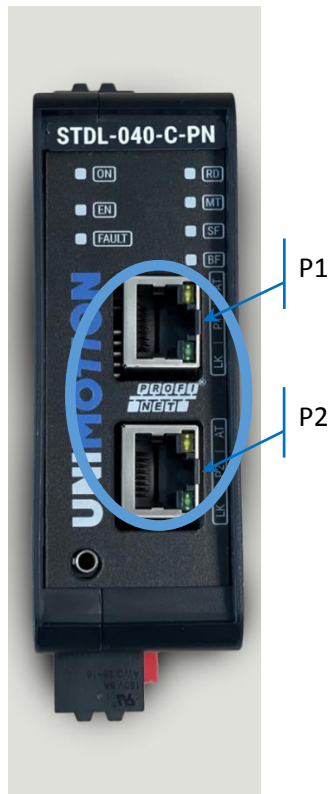
The following table shows the correspondence of signals to the connector's pins:

PROFINET Bus		
Description	Pin	
Shield	8	
Not connected	7	
RXD-	6	
CT-R	5	
CT-T	4	
RXD+	3	
TXD-	2	
TXD+	1	Socket front view

3 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).

3.1 Acyclic access to the registers

The drive's registers 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 drive's registers.

The following image shows an example of how to use the WRREC function block to initialize the *CMC_MTRDT* register.
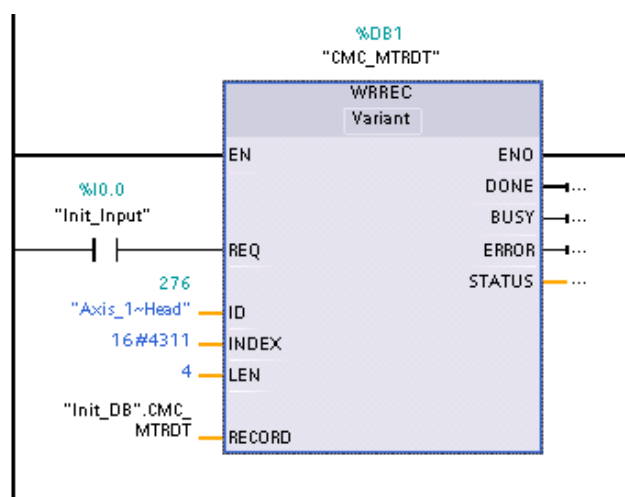
3.2 Cyclic access to the registers

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

The 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

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 does 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.

4   STDL Workspace

The free STDL Workspace running under the Windows platform (Windows 7, Windows 8.1 and e Windows 10 32-bit or 64-bit) allows to diagnose the device and read and write the registers in real-time.

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

When drive is powered only by the STIL device, it receives the auxiliary power (+24 V) and/or the power supply, it automatically restarts.

When the drive is powered only by the STIL device, the power stage, the field bus and the I/O's (digital, analog, encoder, etc.) are not working.

After connecting the STIL device to the STIL IN port on the front of the drive, it is possible to push the *Search* button so that STDL workspace starts scanning the STIL device and the connected drives.

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:
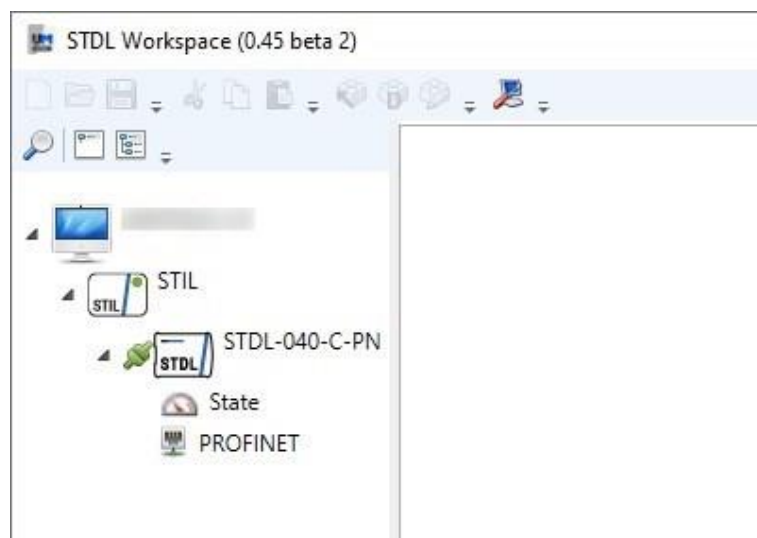
4.1   Firmware update

By double-clicking on the drive's name (STDL-xxx-C-PN 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.

4.2   Status and Diagnostics

In the STDL Workspace, on the left in the tree view list of the connected devices, double-clicking on the voice *Status* visible under the 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 *Vp* shows the voltage value of the power DC bus inside the drive.

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

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

Under the section *Digital I/O,* the status of the digital inputs and outputs is shown. 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 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 reaches 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 replaces the oldest. The column *Time* shows the moment when the error occurred after the power on of the 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.

4.3   Visualization and modification of registers

In STDL Workspace, in the tree of the connected devices on the left, by double-clicking on the entry *PROFINET*, under the drive, you access the tab that allows visualizing and modifying the registers of the device.

There is also an IMPORT function, where you will simply import already prepared file from UNIMOTION without the need to parametrize the motor and the drive will be configured for the used motor.

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

The 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 3 available Presets 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 5.2 Saving and restoring of default values.

The registers visualization and modification area is logically divided into two zones; in the upper area, there are the most common use registers (favourites) while in the lower zone there are all the drive registers ordered by address.

The *Periodic Update* check box, when selected, maintains updated 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 drive status.

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

The registers contained in the favourites 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 register values can be visualized in decimal, hexadecimal or binary format. To change the visualization base, simply place the mouse over the value and click the right mouse 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 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 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.

Export

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 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.

5 Operation

The 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.

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

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

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

5.1 Minimum settings

The minimum settings to be made before enabling the motor involves motor parameters settings (registers from 4311_h to $431B_h$) and the setting of the running and idle current (registers 5312h *CurrentMin_MTRCNF* and 5313_h *CurrentMax_MTRCNF*).

5.1.1 Motor parameters setting

It is most important to set the motor parameters correctly to obtain a smooth movement, the best dynamic performance and the best efficiency.

It follows a detailed description of the registers involved in the motor settings.

5.1.1.1 Downloading and Importing preprepared parameter motor files

Motor files are downloaded via UNIMOTION website. Lower in the download section, click on SOFTWARE and download the complete .zip file containing all the motor files.

After un-zipping the file, connect the drive to the computer via STIL IN and import the file like explained in section 4.3. of this user manual.

5.1.1.2 4313h PolePairs_MTRDT

The register *PolePairs_MTRDT* allows to set the number of motor poles.

The drive uses this information to properly relate the internal position with the one of the motors.

In case of a two-phases stepper motor, each pole gives rise to four full steps, therefore a motor of 200 steps/rev (1.8° step angle) requires to set a value equal to 50 (200 / 4). If, for example, your motor has 100 steps/rev, you will set the value 25 or the value 100 if you are using a motor of 400 steps/rev (0.9° steps angle).

5.1.1.3 4315h Resistance_MTRDT

The register *Resistance_MTRDT* must be compiled with the correct value of the motor phase resistance. Each unit is worth 10 mΩ (i.e., 0.01 Ω) then, for example, to set a value of 3.5 Ω, it is necessary to write the value 350 (3.5 / 0.01) in the parameter

The motor phase resistance is normally specified by the manufacturer and shown in the motor technical datasheet.

Some motors allow more types of phases connection and in this case, it is necessary to verify for which connection the resistance value is specified and adapt it to the phase connection chosen to connect the motor to the drive. The following tables shows the conversion factors to be used:

Phase connection with which the manufacturer has characterized the resistance	Connection chosen for the phases		
	Unipolar	Bipolar Parallel	Bipolar Series
Unipolar	Not supported	0.5	2
Bipolar Parallel	Not supported	1	4
Bipolar Series	Not supported	0.25	1

For example, if the motor has a characteristic resistance of 2.2 Ω in unipolar and is connected to the drive with the phases set in bipolar parallel, the register *Resistance_MTRDT* will have to be compiled with the value 110 (2.2 * 0.5 / 0.01); instead, in case of a bipolar series connection the value to be inserted in the register *Resistance_MTRDT* will be 440 (2.2 * 2 / 0.01).

If the two-phase motor has four wires, it means that the type of phase connection has been fixed during production and the resistance value specified by the manufacturer is therefore the one to be written in the register *Resistance_MTRDT*, without any further processing.

In the event that the value of the resistance is unknown, it is possible to measure it through an ohmmeter. It is suggested to carry out the

measurement with the phases already connected in the chosen configuration, furthermore it is a good idea to average the value through repeated measurements on more motors, if available.

5.1.1.4 4316h Inductance_MTRDT

The register *Inductance_MTRDT* must be filled with the correct value of the motor phase inductance. Each unit is worth 10 µH (i.e., 0.01 mH) therefore to set, for example, a value of 4.2 mH you need to write the value 420 (4.2 / 0.01) in the register.

Some motors allow more types of phase connection and in this case, it is necessary to verify for which connection the inductance value is specified and adapt it to the phase connection chosen to connect the motor to the drive. The following table shows the conversion factors to be used:

Phase connection with which the manufacturer has characterized the inductance	Connection chosen for the phases		
	Unipolar	Bipolar Parallel	Bipolar Series
Unipolar	Not supported	1	4
Bipolar Parallel	Not supported	1	4
Bipolar Series	Not supported	0.25	1

If, for example, the motor has a characteristics inductance of 1.6 mH in unipolar and is connected to the drive with the phases in bipolar parallel, you will have to compile the register *Inductance_MTRDT* with the value 160 (1.6 * 1 / 0.01); instead, if you choose a bipolar series connection the value to be entered will be of 640 (1.6 * 4 / 0.01).

If a two-phase motor has four wires, it means that the type of phase connection has been already fixed during production and the inductance value specified by the manufacturer is therefore the one to be used for the register *Inductance_MTRDT*, without any further processing.

In the event that the value of the inductance is unknown, it is possible to measure it through an inductance meter. We suggest you to carry out the measurement with the phases already connected in the configuration chosen for the drive, furthermore it is a good practice to average the value through repeated measurements on more motors, if available.

5.1.1.5 4317h BackEMF_MTRDT

The register *BackEMF_MTRDT* must be compiled with the value of the counter-electromotive force generated by the motor at a speed of 1000 rpm. Each unit is worth 10 mV (i.e. 0.01 V) therefore, for example, if the motor generates 25 V at 1000 rpm you need to write in the register *BackEMF_MTRDT* the value 2500 (25 / 0.01).

The counter-electromotive force is normally specified by the manufacturer in the motor technical datasheet.

Some motors allow more types of phase connection and in this case, it is necessary to verify for which connection the value of counter-electromotive force is specified and adapt it to the phase connection chosen to connect the motor to the drive. The following table shows the conversion factors to be u:

Phase connection with which the manufacturer has characterized the counter-electromotive force	Connection chosen for the phases		
	Unipolar	Bipolar Parallel	Bipolar Series
Unipolar	Not supported	1	2
Bipolar Parallel	Not supported	1	2
Bipolar Series	Not supported	0.5	1

If, for example, the motor has a counter-electromotive force of 28 V at 1000 rpm in unipolar and is connected to the drive with the phases in bipolar parallel, you will have to compile the register *BackEMF_MTRDT* with the value 2800 (28 * 1 / 0.01); instead, if you choose a bipolar series connection the value to be inserted will be of 5600 (28 * 2 / 0.01).

If a two-phase motor has four wires, it means that the type of phase connection has been already fixed during production and the value of the counter-electromotive force generated by the motor is therefore the one to be used for the register *BackEMF_MTRDT*, without any further processing.

In the event that the value of the counter-electromotive force is unknown, it is possible to measure it through an AC voltmeter connected to one phase of the motor and make it rotate at a speed such to produce a BEMF with a frequency of about 50 Hz. Successively you need to scale the measured voltage value at a speed of 1000 rpm using the formula below:

Vbemf = (Vac * 1000) / Mrpm

Where Vbemf is the value of the counter-electromotive force expressed in V, Vac is the measured voltage expressed in V and Mrpm is the speed at which the motor has been rotated expressed in RPM. If, for example, the motor was rotated at a speed of 60 rpm (to obtain 50 Hz) and the measured voltage value was of 4.85 V AC, the Vbemf value will be equal to 80.83 (4.85 * 1000 / 60) and the value to be inserted in the register *BackEMF_MTRDT* will be (80.83 / 0.01).

Note that the frequency of the BEMF is related to the motor speed through the number of poles, according to the following relationship:

Fhz = Npl * Mrpm / 60

Where Fhz is the frequency of the BEMF expressed in Hz, Npl is the number of motor poles (non-dimensional) and Mrpm is the rotation speed of the

motor. If, for example, we rotate a motor with 50 poles (corresponding to a step angle of 1.8°) at 100 rpm we obtain a BEMF frequency equal to about 83.3 Hz.

It is suggested to average the value through repeated measurements on more motors, if available.

It is essential that the value inserted in the register *BackEMF_MTRDT* corresponds to the counter-electromotive force generated at a speed of 1000 rpm.

5.1.1.6 4318h RatedCurrent_MTRDT

The register *RatedCurrent_MTRDT* must be compiled with the motor rated current. Each unit is worth 10 mArms (i.e., 0.01 Arms) then to set, for example, a value of 4.2 Arms you need to write the value 420 (4.2 / 0.01) in the register.

The value written in the register *RatedCurrent_MTRDT* must take into account the chosen phases connection. According to the connection chosen for the phases and to the configuration chosen by the manufacturer to characterize the rated current, it is necessary to consider one of the scale factors shown in the table below:

Phase connection with which the manufacturer has characterized the current	Connection chosen for the phases		
	Unipolar	Bipolar Parallel	Bipolar Series
Unipolar	Not supported	1.41	0.707
Bipolar Parallel	Not supported	1	0.5
Bipolar Series	Not supported	2	1

For example, if the manufacturer specifies a phase current of 2 A in unipolar connection and you choose to connect the motor to the drive in bipolar parallel, it is necessary to set the register *RatedCurrent_MTRDT* to the value 280 (2 * 1.41 / 0.01).

For example, if the manufacturer specifies a current of 2 A for a bipolar parallel connection and the motor is connected to the drive in bipolar parallel, no conversion is needed and the register *RatedCurrent_MTRDT* can be set with the value 200 (2 / 0.01). Instead, if you choose a bipolar series connection the register will have to be compiled with the value (2 * 0.5 / 0.01).

If a two-phase motor has four wires, it means that the type of phase connection has been already fixed during production and the value of the rated current specified by the manufacturer is therefore the one to insert in the field *Current,* without any further processing.

It is essential <u>that the value in the register *RatedCurrent_MTRDT* corresponds exactly to the rated current of the motor and that this parameter is never used to set the working current.</u> The working current of the motor is set through the register 5313ₕ *CurrentMax_MTRCNF* described later.

5.1.1.7   4319h MaxCurrent_MTRDT

The register *MaxCurrent_MTRDT* must be compiled with the maximum current to which the motor can be supplied. Each unit is worth 10 mArms (i.e., 0.01 Arms) then to set, for example, a value of 5.0 Arms you need to write in the parameter the value 500 (5.0 / 0.01).

In the event that the data is not available in the motor datasheet, it is suggested to use the same value of the register 4318ₕ *RatedCurrent_MTRDT*.

The value written in the register *MaxCurrent_MTRDT* must consider the connection chosen for the phases for the motors which provide more possibilities. See previous chapter (5.1.1.6 4318h RatedCurrent_MTRDT).

5.1.1.8   431Ah RatedTorque_MTRDT

The register *RatedTorque_MTRDT* must be compiled with the static torque value of the motor when supplied at the rated current. Each unit is worth 10 mNm (i.e., 0.01 Nm) then to set, for example, a value of 6.8 Nm you need to write in the parameter the value 6800 (6.8 / 0.01).

The static torque value is normally specified by the manufacturer in the motor datasheet and is often called *Holding Torque*. If the value is expressed in units of measurement different from Nm, it is possible to convert it through the coefficients shown in the table below:

	Newton Centimeter (N-cm)	Newton Meter (N-m)	Pound Force Inch, (lbf-in)	Ounce Force Inch, (ozf-in)
Newton Centimeter (N-cm)	**1 N-cm**	**0.01 N-m**	**0.0885 lbf-in**	**1.42 ozf-in**
Newton Meter (N-m)	**100 N-cm**	**1 N-m**	**8.85 lbf-in**	**142 ozf-in**
Pound Force Inch, (lbf-in)	**11.3 N-cm**	**0.113 N-m**	**1 lbf-in**	**16 ozf-in**
Ounce Force Inch, (ozf-in)	**0.706 N-cm**	**0.00706 N-m**	**0.0625 lbf-in**	**1 ozf-in**

Some motors allow more types of phase connection and in this case, it is necessary to verify for which connection the rated torque is specified and

adapt it to the phase connection chosen to connect the motor to the drive. The following table shows the conversion factors to be used:

Phase connection with which the manufacturer has characterized the rated static torque	Connection chosen for the phases		
	Unipolar	Bipolar Parallel	Bipolar Series
Unipolar	Not supported	1.41	1.41
Bipolar Parallel	Not supported	1	1
Bipolar Series	Not supported	1	1

For example, if a motor has a rated static torque of 3.1 Nm in unipolar and is connected to the drive with the phases in bipolar parallel or in bipolar series, you will have to compile the register *RatedTorque_MTRDT* with the value 437 (3.1 * 1.41 / 0.01).

If a two-phase motor has four wires it means that the type of phase connection has been already fixed during production and the value of *Holding Torque* specified by the manufacturer is therefore the one to be used, without any further processing.

In the event that the value of the motor torque is unknown, it is possible to measure it through a torquemeter with the phases of the motor supplied at the rated current. It is suggested to execute the measurement with the phases already connected in the configuration chosen for the drive, furthermore it is a good idea to average the value through repeated measurements on more motors, if available.

It is essential that the value in the register *RatedTorque_MTRDT* corresponds exactly to the rated torque of the motor.

5.1.1.9   431Bh MaxSpeed_MTRDT

The register *MaxSpeed_MTRDT* must be compiled with the maximum speed that the motor can reach. Each unit is worth 0.1 rpm therefore to set, for example, a value of 600 rpm it is necessary to write the value 6000 (600 / 0.1) in the register.

5.1.2 Running and idle current configuration

The drive allows 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 (register 43A1$_h$ *Mode_CRRG*).

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 description of the parameters relating to the motor operating current follows.

5.1.2.1   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 (register 4318$_h$ *RatedCurrent_MTRDT*) 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 4 Arms, the idle current will be equal to 1.2 Arms (30 % of 4 A).

5.1.2.2   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 current (register 4318$_h$ *RatedCurrent_MTRDT*) 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 4 Arms, the idle current will be equal to 3.2 Arms (80 % of 4 A).

5.1.2.3   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 1 ms, therefore setting for example the value 500, the drive will wait for 500 ms from the motor stop before changing the phase current.

5.2 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:

Name	Mnemonic
PDO Mapping	Note
Maximum	Unit
	Descrizione

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 an 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 STDL Workspace commissioning software.

The registers can be also be restored to the default value writing an appropriate key in the 4121$_h$ *RestoreAllDefaultParameters_RDP* or using the STDL Workspace commissioning software.

UNIMOTION

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.

5.3 Motor Holding Brakes

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

The brake cannot be connected directly to digital output of the drive!

Brake must be controlled via an additional relay with externally supplied power.
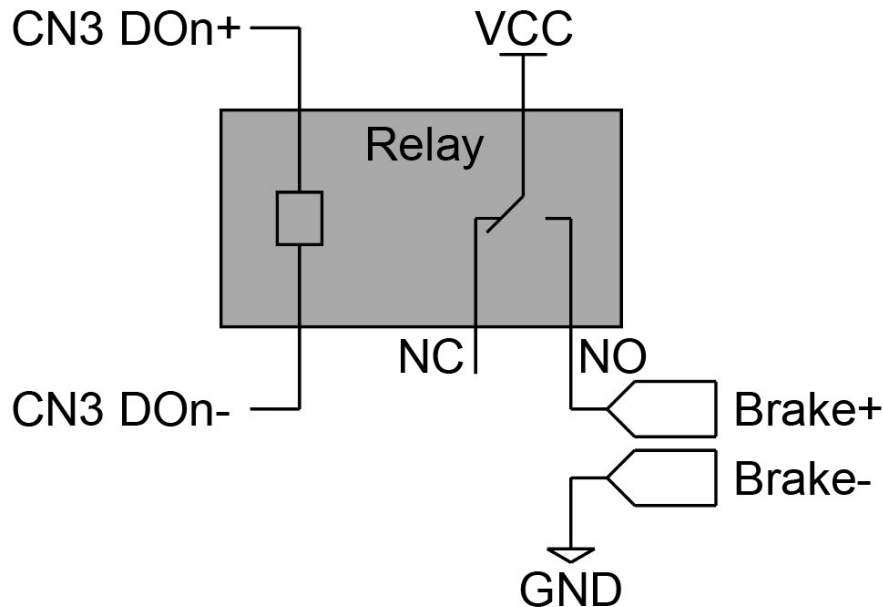
Through the register *Option_HBRKS* it is possible to set the drive to handle automatically the brake upon enabling the motor, taking into account the brake's characteristic engaging and disengaging time.

With the brake control enabled, by enabling the motor the drive activates the digital output immediately predisposed to control the break as to release it and at the same time supplies the motor to maintain it in 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 from the operating state immediately 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:

Release Time
(Release_Time_HBRKS)

Application Time
(Application_Time_HBRKS)
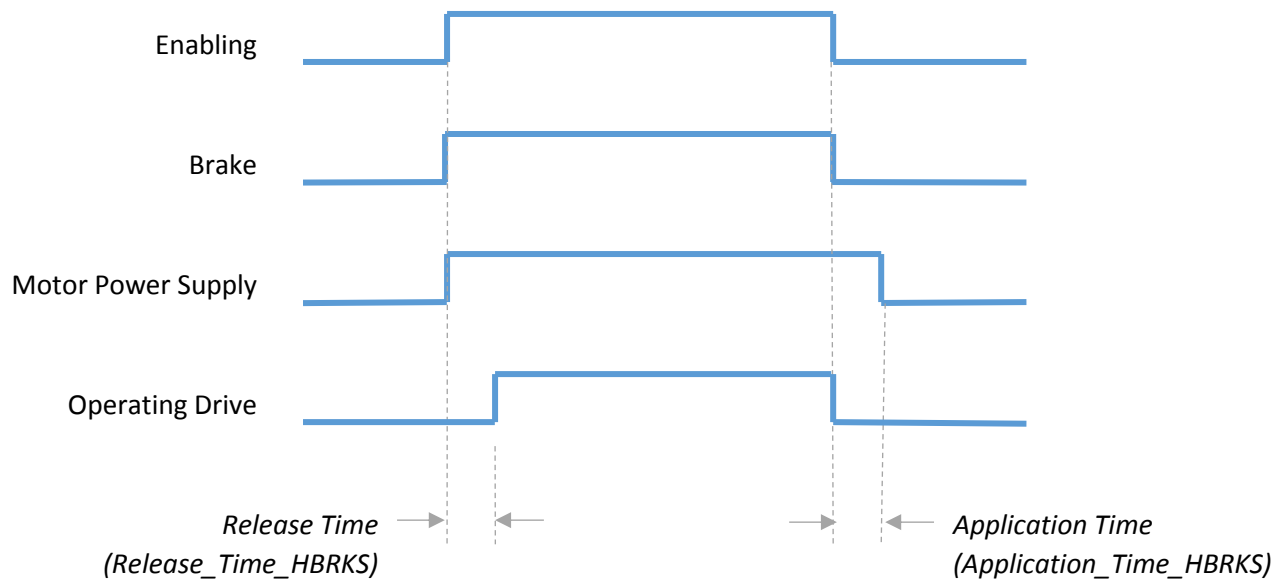
The register *Option_HBRKS* is also useful to enable and set the manual control of the brake that can intervene in an exclusive way, with respect to the control made by the drive, or shared.

The manual control can be performed via a digital input that can be set through the register *Holding_Brake_DIA* or through the bit 0 of the register *Control_HBRKC.*

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

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

5.4   SSI Encoder

The drive is able to read rotary and linear encoders with synchronous serial interface (SSI - Synchronous Serial Interface).

Both mono-turn and multi-turn binary encoders can be connected and the total number of bits of the frame, of the multi-turn part and of the single-turn part can be freely set independently. It is also possible to indicate the position of the most significant bit (MSB) inside the frame to correctly read encoders with data center alignment (tree structure) or aligned to the right or left of the frame.

The SSI encoder can be used to initialize the position of the motor (at the power on or on request) and to verify the positioning through the Following Error in real time.

However, when using the SSI encoder, it is still possible to continue and use the incremental encoder at the same time, for example for a redundant position control or to use the closed-loop control mode.

The SSI encoder can be mounted on the motor shaft or in a different kinematic position, even after a reduction ratio. Using the *PositionOffset_ENCSSI*, *ScalingFactor_N_ENCSSI* and *ScalingFactor_D_ENCSSI* registers, it is possible to bring back the quote read by the SSI encoder to the motor position declared in 0.0001 rev. The calculation made by the drive is as follows:

Position_ENCSSIV = (*PositionRaw_ENCSSIV* - *PositionOffset_ENCSSI*) * *ScalingFactor_N_ENCSSI* / *ScalingFactor_D_ENCSSI*

The SSI encoder can be supplied externally or through the +5 Vdc output provided on the drive for this purpose.

The drive is able to detect error conditions such as the cut or short-circuit of the connecting cable with the SSI encoder. The error control can be activated by setting bit 1 of the *Configuration_ENCSSI* register to 1.

The drive is able to read a wide variety of SSI encoders thanks to the possibility to freely configure the data frame structure. For example, it is possible to use the SSI encoders with the data organized in the center (tree structure), aligned to the left or to the right.

Through the *FrameLength_ENCSSI*, *MSB_Offset_ENCSSI*, *TurnsBits_ENCSSI* and *SingleTurnBits_ENCSSI* registers, it is possible to configure respectively the total length of the frame, the position of the most significant bit (MSB), the number of bits used for the revolutions counting and the number of bits used for the single-turn position.

The following example shows an SSI encoder with a 25 bits length frame with the data aligned at the center and composed of 8 bits for the full revolutions counting and 12 bits for the single-turn position measurement.

clock																									

data	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	0	0	0	0	n7	n6	n5	n4	n3	n2	n1	n0	p12	p11	p9	p8	p7	p6	p5	p4	p3	p2	p1	p0	0
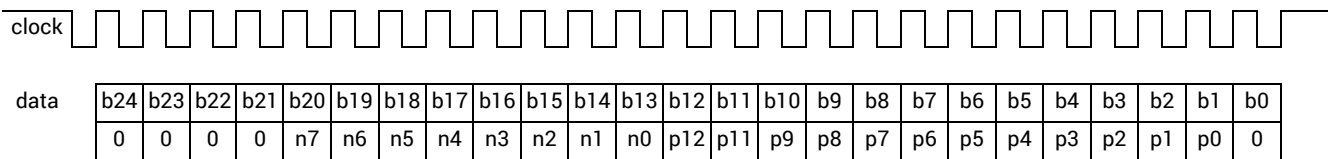
b..   Number of bits in the frame

n..   bit for the full revolutions counting

p..   bit for the single-turn position measurement

To correctly read an SSI encoder with the data frame organized as shown above, the following registers must be configured:

FrameLength_ENCSSI = 25

TurnsBits_ENCSSI = 8

SingleTurnBits_ENCSSI = 12

MSB_Offset_ENCSSI = 4

Setting the bit 4 of the *Configuration_ENCSSI* register to 1, the position, expressed in 0.0001 rev read by the SSI encoder, is copied into the *Position_actual_value* register at each transition from 0 to 1 of the bit 0 of the *Status_ENCSSIV* register and at each transition in the *Operational* state of the drive.

The *Position_actual_value* register can be initialized with the position expressed in 0.0001 rev read by the SSI encoder also through the *Homing* operative mode setting the *Homing_method* register = −80.

For example, if you want to use the SSI encoder to monitor the following error (*Following_error_actual_value*) you need to configure the *Feedback_MTNSTP* register = 0x12;

The *Data* signal of the SSI encoder and the *I (Index)* signal of the incremental encoder share the same connection, therefore when using the SSI encoder, the Index signal of the incremental encoder cannot be used and the bit 0 of the *Configuration_ENCMTR* register must be set to 0.

5.5 Closed-loop

STDL-xxx-C-PN series are able to control the closed-loop motor 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 perform the closed-loop control the drive needs to know the motor position in real time, therefore the motor must be equipped with an Encoder. This latter must ensure a delay of less than 5 µs between the real motor position and the one acquired at the sampling point, otherwise the maximum speed allowed by the closed-loop control is reduced.

In order for the drive to operate properly in closed-loop, it is necessary to accurately configure the characteristics of the motor, the encoder and finally enable the closed-loop control mode by setting the bits 1..0 of the *Feedback_MTNSTP* register to 1.

5.6 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 32 bit (double word) and it is read-only. The following table shows the meaning of the bits contained in the register:

	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	**bit 26..24 / Operating mode:** 001 Position, 010 Velocity, 011 Torque, 111 Homing. The bit combinations not in the table must be considered as non-selected operating mode.	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 1 / bit 0 / Description:** 0 0 Disabled, 0 1 Disabled, 1 0 Disabled, 1 1 Enabled	

The OperationMode sub-table (bits 26..24):

bit 26..24	Operating mode
001	Position
010	Velocity
011	Torque
111	Homing

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

Enable sub-table (bits 1, 0):

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

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

The *ControlDWord* registers also has a dimension of 32 bit (double word). The meaning of the bits is summarized 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	bit 26..24 — Operating mode; 000 None; 001 Position; 010 Velocity; 011 Torque; 111 Homing. The bit combinations not in the table must be avoided.	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		

Operating mode table (bits 26..24):

bit 26..24	Operating mode
000	None
001	Position
010	Velocity
011	Torque
111	Homing

Enable table (bits 1 and 0):

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

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

5.6.1 Motor Enabling

The motor is enabled setting the bit 1 (Enable) and bit 0 (Enable) of the *ControlDWord* register to 1. If there are no active faults and the 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 bits 26..24 (Operating mode) have the value 001).

The enabling of the motor is confirmed by the bit 1 (Enable) and bit 0 (Enable) of the *StatusDWord* which both take value 1.

To disable the motor, you just need to set the bit 1 (Enable) and bit 0 (Enable) of the *ControlDWord* register to 0.

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

5.6.2 Emergency Stop (QuickStop)

By setting the bit 2 (QuickStop) 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 bit 2 (QuickStop) of the *StatusDWord* register assumes the value 1 each time the drive executes the emergency stop.

By setting the bit 2 (QuickStop) of the *ControlDWord* register to 0, the emergency stop request is removed and if the procedure is completed (the motor has stopped) the 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 bit 2 (QuickStop) of the *StatusDWord* register is set to 1 and the bit 2 (QuickStop) 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 bit 4 (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 bit 4 (FaultAcknowledge) of *ControlDWord* to 1. Note that the drive uses the rising edge of bit 4 (transition from 0 to 1) to exit the emergency condition. If bit 4 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.

5.6.3 Motor Stop (Halt)

By setting the bit 3 (Halt) of the *ControlDWord* register to 1, the motor stops with the deceleration ramp defined by the *Deceleration* register.

The bit 3 (Halt) 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 bit 3 to 1.

5.6.4 Fault and Warning

When the drive detects a fault condition it sets the bit 4 (Fault) 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 bit 4 (FaultAcknowledge) of *ControlDWord* to 1. If the fault condition persists the drive ignores the bit 4 and remains in the fault status. Note that the drive uses the rising edge of bit 4 (transition from 0 to 1) to exit the fault condition. If the bit 4 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 required to switch off and on the device.

When the drive detects warning conditions, it signals them by activating the bit 5 (Warning) of *StatusDWord*. The warning event does not compromise the correct functioning of the drive immediately that, therefore, remains in the operating state.

To accept the warning signal, it is necessary to set the bit 5 (WarningAcknowledge) of *ControlDWord* to 1. If the warning condition persists the drive ignores bit 5. Note that the drive uses the rising edge of the bit (transition from 0 to 1) to cancel the warning signal. If bit 4 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.

5.6.5 Real-time or delayed target process

The 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 bit 8 (TriggerMode) 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 changes to reach the new value.

Instead, by setting the bit 8 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 *TargetPosition* register, 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 bit 4 (FaultAcknowledge) 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 bit 9 (Trigger) of *ControlDWord*.

The rising edge (transition from 0 to 1) of the bit 9 (TriggerAcknowledge) of *StatusDWord* register confirms the application of the targets.

5.6.6 Absolute or relative positioning

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

When the bit 10 (AbsoluteRelative) of *ControlDWord* is set to 0, the position is interpreted in absolute value while when the bit 0 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 (bit 8 (TriggerMode) = 0 of *ControlDWord*), it is not possible to use the relative positioning and the bit 10 of *ControlDWord* is ignored.

5.6.7 Target achieved and motor halted

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

The bit 13 (TorqueReacher) 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 bit 14 (VelocityReacher) 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 bit 15 (PositionReached) 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 bit 12 (Standstill) 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*.

5.6.8 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 bit 18 (VelocityError) 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 bit 19 (PositionError) 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 bit 18 (VelocityErrorLatch) and bit 19 (PositionErrorLatch) 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 bit 18 and bit 19 of *ControlDWord* to 0.

5.6.9 Operating mode

The setting of the operating mode is by the bit 26, bit 25 and bit 24 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 drive confirms the active operating mode by the bit 26, bit 25 and bit 24 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*

5.7 Operating Modes

The following paragraphs describe the different operating modes. All the examples assume that the drive is started with the default values and successively configured with the minimum setting described in the previous chapter 5.1 Minimum settings.

5.7.1 Position (pp)

In this mode the 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 bit 26..24 (Operating mode) of *ControlDWord* to the value 1 (bit 26 = 0, bit 25 = 0 and bit 24 = 1).

The position, velocity and torque targets can be processed in real or delayed time whenever the trigger is activated, as described in chapter 5.6.5 *Real-time or delayed target process*, furthermore the master controller can command absolute or relative positioning, as described in chapter 5.6.6 *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 chapter 5.6.7 *Target achieved and motor* and 5.6.8 *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 400 rpm, an acceleration equal to 2000 rpm/s and a deceleration of 3500 rpm/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*

5.7.2 Velocity (pv)

In this mode the 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 bit 26..24 (Operating mode) of *ControlDWord* to the value 2 (bit 26 = 0, bit 25 = 1 and bit 24 = 0).

The velocity and torque targets can be processed in real or delayed time whenever the trigger is activated, as described in chapter 5.6.5 *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 chapter 5.6.7 *Target achieved and motor* and 5.6.8 *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 640.5 rpm, with an acceleration equal to 500 rpm/s. To decelerate it will use a ramp of 1000 rpm/s while the maximum torque is set at the 100 % of the rated value.

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

Note that in the *Velocity* mode, the *TargetPosition* register is ignored.

5.7.3 Torque (tq)

The *Torque* 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 through the *TargetVelocity* register.

To select the Torque mode, it is sufficient to set the bits 26..24 (Operating mode) of *ControlDWord* to the value 3 (bit 26 = 0, bit 25 = 1 and bit 24 = 1).

The torque target can be processed in real or delayed time whenever the trigger is activated, as described in chapter 5.6.5 *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 5.6.7 *Target achieved and motor* .

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 (PLC output)	IN (PLC input)
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.

5.7.4 Homing (hm)

Through the Homing mode, the 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 bits 26..24 (Operating mode) of *ControlDWord* to the value 7 (bit 26 = 1, bit 25 = 1 and bit 24 = 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 bit 9 (Trigger) of *ControlDWord).*

If the homing ends successfully, the bit 10 (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 drive must be associated with the corresponding digital input so that the drive 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 the negative limit switch is inactive, the motor will start rotating in the counterclockwise direction up to the limit switch, then reverse and home at the first index (Z or I pulse) outside the negative limit switch. At the start, if the negative limit switch is active, the motor will start rotating in the clockwise direction until it leaves the limit switch (becomes inactive), then home at the first index (Z or I pulse) outside the negative limit switch.	●			●
2	At the start, if the positive limit switch is inactive, the motor will rotate in the clockwise direction up to the limit switch, then reverse and home at the first index (Z or I pulse) outside the positive limit switch. At the start, if the positive limit switch is active, the motor will rotate in the counterclockwise direction until it leaves the limit switch, then home at the first index (Z or I pulse) outside the positive limit switch.	●			●
3	At the start, if the home switch is inactive, the motor will rotate in the clockwise direction up to home switch, then reverse and home at the first index (Z or I pulse) outside the home switch. At the start, if the home switch is active, the motor will rotate in the counterclockwise direction until it leaves the switch, then home at the first index (Z or I pulse) outside the home switch.			●	●
4	At the start, if the home switch inactive, the motor will rotate in the clockwise direction up to home switch, then home at the first index (Z or I pulse) inside the home switch. At the start, if the home switch is active, the motor will rotate in the counterclockwise direction until it leaves the switch, then reverse and home at the first index (Z or I pulse) inside the home switch.			●	●
5	At the start, if the home switch is active, the motor will rotate in the clockwise direction until it leaves the switch, then home at the first index (Z or I pulse) outside the home switch. At the start, if the home switch is inactive, the motor will rotate in the counterclockwise direction up to find the switch, then reverse and home at the first index (Z or I pulse) outside the home switch.			●	●
6	At the start, if the home switch is active, the motor will rotate in the clockwise direction until it leaves the switch, then			●	●

	reverse and home at the first index (Z or I pulse) inside the home switch. At the start, if the home switch is inactive, the motor will rotate in the counterclockwise direction up to find the switch, then reverse and home at the first index (Z or I pulse) inside the home switch.				
7	At the start, if the home switch is inactive, the motor will rotate in the clockwise direction up to find the home switch or the positive limit switch. In case of home switch, reverse and home at the first index (Z or I pulse) outside the switch. In case of positive limit switch, reverse up to find the home switch, continue up to leave the home switch, then home at the first index (Z or I pulse) outside the switch. At the start, if the home switch is active, the motor will rotate in the counterclockwise direction up to leave the home switch, then home at the first index (Z or I pulse) outside the switch.	●		●	●
8	At the start, if the home switch is inactive, the motor will rotate in the clockwise direction up to find the home switch or the positive limit switch. In case of home switch, it homes at the first index (Z or I pulse) 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 home at the first index (Z or I pulse) inside the switch. At the start, if the home switch active, the motor will rotate in the counterclockwise direction up to leave the home switch, then reverse and home at the first index (Z or I pulse) inside the switch.	●		●	●
9	At the start, if the home switch is inactive, the motor will rotate in the clockwise direction up to find the home switch or the positive limit switch. In case of home switch, continue up to leave it then reverse and home at the first index (Z or I pulse) inside the switch. In case of positive limit switch, reverse up to find the home switch, then home at the first index (Z or I pulse) inside the switch. At the start, if the home switch is active the motor will rotate in the clockwise direction up to leave the home switch, then reverse and home at the first index (Z or I pulse) inside the home switch.	●		●	●
10	At the start, if the home switch is inactive the motor will rotate in the clockwise direction up to find the home switch or the positive limit switch. In case of home switch, continue up to leave it then home at the first index (Z or I pulse) 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 home at the first index (Z or I pulse) outside the switch. At the start, if the home switch is active the motor will rotate in the clockwise direction up to leave the home switch, then home at the first index (Z or I pulse) outside the home switch.	●		●	●

11	At the start, if the home switch is inactive the motor will rotate in the counterclockwise direction up to find the home switch or the negative limit switch. In case of home switch, reverse and home at the first index (Z or I pulse) 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 home at the first index (Z or I pulse) outside the switch. At the start, if the home switch is active the motor will rotate in the clockwise direction up to leave the home switch, then home at the first index (Z or I pulse) outside the home switch.	●	●	●
12	At the start, if the home switch is inactive the motor will rotate in the counterclockwise direction up to find the home switch or the negative limit switch. In case of home switch, home at the first index (Z or I pulse) 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 home at the first index (Z or I pulse) inside the switch. At the start, if the home switch is active the motor will rotate in the clockwise direction up to leave the home switch, then reverse and home at the first index (Z or I pulse) inside the home switch.	●	●	●
13	At the start, if the home switch is inactive the motor will rotate in the counterclockwise direction up to find the home switch or the negative limit switch. In case of home switch, continue up to leave it then reverse and home at the first index (Z or I pulse) inside the switch. In case of negative limit switch, reverse up to find the home switch, then home at the first index (Z or I pulse) inside the switch. At the start, if the home switch is active the motor will rotate in the clockwise direction up to leave the home switch, then reverse and home at the first index (Z or I pulse) inside the home switch.	●	●	●
14	At the start, if the home switch is inactive the motor will rotate in the counterclockwise direction up to find the home switch or the negative limit switch. In case of home switch, continue up to leave it then home at the first index (Z or I pulse) 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 home at the first index (Z or I pulse) outside the switch. At the start, if the home switch is active the motor will rotate in the counterclockwise direction up to leave the home switch, then home at the first index (Z or I pulse) outside the switch.	●	●	●
17	At the start, if the negative limit switch is inactive the motor will rotate in the counterclockwise direction up to find the limit switch, then reverse and home at the active/inactive switch transition.	●		

	At the start, if the negative limit switch is active the motor will rotate in the clockwise direction with homing at the active/inactive switch transition.				
18	At the start, if the positive limit switch is inactive the motor will rotate in the clockwise direction up to the limit switch, then reverse and home at the active/inactive switch transition. At the start, if the negative limit switch is active the motor will rotate in the counterclockwise direction with homing at the active/inactive switch transition.	●			
19	At the start, if the home switch is inactive the motor will rotate in the clockwise direction up to the home switch, then reverse and home at the active/inactive switch transition. At the start, if the home switch is active the motor will rotate in the counterclockwise direction with homing at the active/inactive switch transition.			●	
20	At the start, if the home switch is inactive the motor will rotate in the clockwise direction up to the home switch, then home at the inactive/active switch transition. At the start, if the home switch is active the motor will rotate in the counterclockwise direction up to leave the switch, then reverse and home at the inactive/active switch transition.			●	
21	At the start, if the home switch is active the motor will rotate in the clockwise direction up to leave the switch, then home at the active/inactive switch transition. At the start, if the home switch is inactive the motor will rotate in the counterclockwise direction up to find the switch, then reverse and home at the active/inactive switch transition.			●	
22	At the start, if the home switch is active the motor will rotate in the clockwise direction up to leave the switch, then reverse and home at the inactive/active switch transition. At the start, if the home switch is inactive the motor will rotate in the counterclockwise direction up to find the switch, then home at the inactive/active switch transition.			●	
23	At the start, if the home switch is inactive the motor will rotate in the clockwise direction up to find the home switch or the positive limit switch. In case of home switch, reverse and home 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 home at the active/inactive switch transition. At the start, if the home switch is active the motor will rotate in the counterclockwise direction up to leave the switch, then home at the active/inactive switch transition.	●		●	
24	At the start, if the home switch is inactive the motor will rotate in the clockwise direction up to find the home switch or the positive limit switch. In case of home switch, home 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 home at the inactive/active switch transition. At the start, if the home switch is active the motor will rotate in the counterclockwise direction up to leave the home switch, then reverse and home at the inactive/active switch transition.				
25	At the start, if the home switch is inactive the motor will rotate in the clockwise direction up to find the home switch or the positive limit switch. In case of home switch, continue up to leave it, then reverse and home 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 the home switch is active the motor will rotate in the clockwise direction up to leave the home switch, then reverse and home at the inactive/active switch transition.	●		●	
26	At the start, if the home switch is inactive the motor will rotate in the clockwise direction up to find the home switch or the positive limit switch. In case of home switch, continue up to leave it, then home 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 home at the active/inactive switch transition. At the start, if the home switch is active the motor will rotate in the clockwise direction up to leave the home switch, then home at the active/inactive switch transition.	●		●	
27	At the start, if the home switch is inactive the motor will rotate in the counterclockwise direction up to find the home switch or the negative limit switch. In case of home switch, reverse and home 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 home at the active/inactive switch transition. At the start, if the home switch is active the motor will rotate in the clockwise direction up to leave the home switch, then home at the active/inactive switch transition.		●	●	
28	At the start, if the home switch is inactive the motor will rotate in the counterclockwise direction 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 home at the inactive/active switch transition. At the start, if the home switch is active the motor will rotate in the clockwise direction up to leave the home switch, then reverse and home at the inactive/active switch transition.		●	●	

29	At the start, if the home switch is inactive the motor will rotate in the counterclockwise direction up to find the home switch or the negative limit switch. In case of home switch, continue up to leave it then reverse and home at the inactive/active switch transition. In case of negative limit switch, reverse up to find the home switch, then home at the inactive/active switch transition. At the start, if the home switch is active the motor will rotate in the counterclockwise direction up to leave the home switch, then reverse and home at the inactive/active switch transition.	●	●		
30	At the start, if the home switch is inactive the motor will rotate in the counterclockwise direction up to find the home switch or the negative limit switch. In case of home switch, continue up to leave it then home 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 home at the active/inactive switch transition. At the start, if the home switch is active the motor will rotate in the counterclockwise direction up to leave the home switch, then home at the active/inactive switch transition.	●	●		
33	The motor will rotate in the counterclockwise direction and home at the first index (Z or I pulse) found.				●
34	The motor will rotate in the clockwise direction and home at the first index (Z or I pulse) found.				●
37	Homing at the actual position.				

6 Errors and diagnostics

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

6.1 LED *On*

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

Led *On*	
Off	Power to drive is not supplied
On	Power to drive is supplied

6.2 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 and stopped
Flashing	Motor enabled and in rotation

6.3 LED *Fault*

The red Fault LED is used to indicate the presence of errors in the drive, 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.

6.4 Errors

The drive is able to detect many fault and error conditions and to intervene by stopping or disabling the motor.

The error condition is signaled by the red LED placed on the front panel of the drive, through the digital outputs and the fieldbus.

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

Class	Description	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 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 driver:

Error code	Class	LED Flash	bit	Description / Sub-code
2310$_h$	6	4	3	Detected motor over current

				Sub-code	Description	
				0100$_h$	During PWM modulation	
				0101$_h$	Towards ground	
				0102$_h$	Towards positive power bus	
				0103$_h$	Short circuit between phase A+ and ground	
				0104$_h$	Short circuit between phase A- and ground	
				0105$_h$	Short circuit between phase B+ and ground	
				0106$_h$	Short circuit between Phase B- and ground	
				0109$_h$	Abnormal current between phase A+ and ground	
				010A$_h$	Abnormal current between phase A- and ground	
				010B$_h$	Abnormal current between phase B+ and ground	
				010C$_h$	Abnormal current between phase B- and ground	
23A0$_h$	6	5	5	Detected motor phase open (not connected or interrupted)		
				Sub-code	Description	
				0201$_h$	Phase A	
				0202$_h$	Phase B	
23B2$_h$	6	8	4	Detected over current on the encoder power supply +V		
3210$_h$	6	2	1	Power supply voltage higher than the maximum allowed		
3220$_h$	6	1	0	Power supply voltage lower than the minimum necessary for operating		
4210$_h$	4	3	2	Power stage temperature higher than the maximum allowed value		
5592$_h$	8	13	24	Invalid device descriptor		
5594$_h$	8	13	24	Invalid device configuration		
55A0$_h$	8	13	31	Inappropriate installed firmware		
55A2$_h$	8	13	31	Incompatible installed firmware revision		
6200$_h$	8	15	31	Firmware error		
63A0$_h$	6	10	16	Conflicts in the configuration		
				Sub-code	Description	
				0701$_h$	Selected *Torque mode* without encoder feedback (present or configured).	
				0702$_h$	Selected current proportional to the load (register *Mode_CRRG*) without encoder feedback (present o configured).	
				0704$_h$	Activated encoder feedback without having activated or configured the encoder	

				Sub-code	Description
				0705h	Activated the following error control without having activated or configured the encoder selected as feedback source.
				0706h	Activated the incremental encoder Index management and contemporaneously the SSI encoder.
7282h	8	14	11	Detected invalid offset values	
8100h	6	7	28	Bus communication error	
8611h	6	6	18	Following error. The indicated error class expresses the default value. The class can be modified through the register *Following_Error_ERRCS*.	
8612h	2	6	19	Limit switch activation	

Sub-code	Description
0801h	Positive limit activation
0802h	Negative limit activation
0803h	Positive movement with positive limit switch active
0804h	Negative movement with negative limit switch active

8613h	2	6	17	Homing error

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

7 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, the register *Resistance_MTRDT* which allows to set the motor phase resistance, cannot be modified if the drive is enabled.

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.

7.1 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 100 ms.

Address		Name	Mnemonic
4111ₕ	*SaveAllParameters_SPF*		

	Data Type	Access Type	PDO Mapping	Note
	u32	Read/Write		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.

7.2   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 100 ms.

Address	Name			Mnemonic
4121ₕ	*RestoreAllDefaultParameters_RDP*			
	Data Type	Access Type	PDO Mapping	Note
	u32	Read/Write		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.

7.3   Motor Data

Address	Name			Mnemonic
4311ₕ	*CMC_MTRDT*			
	Data Type	Access Type	PD Mapping	Note
	u32	Read/Write		🖫 It cannot be modified.
	Default Value	Minimum	Maximum	Unit
	0			
	Description			
	Reserved, do not use.			

Address	Name			Mnemonic
4312ₕ	*Type_MTRDT*			
	Data Type	Access Type	PD Mapping	Note
	u8	Read only		🖫 Valid only if *CMC_MTRDT* = 0
	Default Value	Minimum	Maximum	Unit
	12			
	Description			
	It indicates the type of motor connected to the drive according to the table below:			

12	Two-phase Stepper Motor

Address 4313ₕ	PolePairs_MTRDT		Name		Mnemonic
	Data Type u8	Access Type Read/Write	PD Mapping		Note 🖫 Valid only if *CMC_MTRDT* = 0. It cannot be modified with the drive enabled.
	Default Value 50	Minimum 50	Maximum 50		Unit
	Description It allows to set the number of poles of the motor. The number of poles is the number of electrical cycles included in a complete motor revolution. For example, a 200 steps/revolution two-phase stepper motor has a number of poles equal to 50.				

Address 4314ₕ	Wiring_MTRDT		Name		Mnemonic
	Data Type u8	Access Type Read only	PD Mapping		Note 🖫 Valid only if *CMC_MTRDT* = 0
	Default Value 0	Minimum	Maximum		Unit
	Description Motor connection. Currently not used.				

Address 4315ₕ	Resistance_MTRDT		Name		Mnemonic
	Data Type u16	Access Type Read/Write	PD Mapping		Note 🖫 Valid only if *CMC_MTRDT* = 0. It cannot be modified with the drive enabled.
	Default Value 100	Minimum	Maximum 60000	Unit 10 mΩ (Ex. 240 = 2.4 Ω)	
	Description It allows to set the phase resistance of the motor connected to the drive. The set value must take into account the type of phase connection (series or parallel), if the motor allows multiple configurations.				

Address 4316ₕ	Inductance_MTRDT		Name		Mnemonic
	Data Type u16	Access Type Read/Write	PD Mapping		Note 🖫 Valid only if *CMC_MTRDT* = 0. It cannot be modified with the drive enabled.
	Default Value 300	Minimum	Maximum 60000	Unit 10 µH (Ex. 320 = 3.2 mH)	
	Description It allows to set the phase inductance of the motor connected to the drive.				

The set value must take into account the type of phase connection (series or parallel), if the motor allows multiple configurations.

> For optimal operation, it is very important to carefully set this register so that it indicates the real inductance of the connected motor.

Address	Name				Mnemonic
4317h	*BackEMF_MTRDT*				
	Data Type	Access Type	PD Mapping		Note
	u16	Read/Write			⊟ Valid only if *CMC_MTRDT* = 0. It cannot be modified with the drive enabled.
	Default Value	Minimum	Maximum		Unit
	2500		60000		10 mV/1000 rpm (Ex. 4500 = 45 V/1000 rpm)
					Description

It allows to set the BEMF of the motor connected to the drive.

The set value must take into account the type of phase connection (series or parallel), if the motor allows multiple configurations.

Address	Name				Mnemonic
4318h	*RatedCurrent_MTRDT*				
	Data Type	Access Type	PD Mapping		Note
	u16	Read/Write			⊟ Valid only if *CMC_MTRDT* = 0. It cannot be modified with the drive enabled.
	Default Value	Minimum	Maximum		Unit
	100	10	60000		10 mArms (Ex. 420 = 4.2 Arms)
					Description

It allows to set the rated current of the motor connected to the drive.

The set value must take into account the type of phase connection (series or parallel), if the motor allows multiple configurations.

> For optimal operation, it is very important to carefully set this register to the rated current of the motor.

> Do not use this register to modify the motor current. The register *RatedCurrent_MTRDT* must always be set with the rated current value indicated by the motor manufacturer. To modify the running or idle current of the motor, please use the registers *CurrentMax_MTRCNF* and *CurrentMin_MTRCNF* instead.

If the set current value exceeds the maximum device capacity, the latter shall prevail.

Address	Name			Mnemonic
4319h	*MaxCurrent_MTRDT*			

Data Type	Access Type	PD Mapping	Note
u16	Read/Write		🖫 Valid only if *CMC_MTRDT* = 0. It cannot be modified with the drive enabled.

Default Value	Minimum	Maximum	Unit
130	10	60000	10 mArms (Ex. 550 = 5.5 Arms)

Description

It allows to set the maximum current applied to the motor for short periods.

The set value must take into account the type of phase connection (series or parallel), if the motor allows multiple configurations.

In case of the UNIMOTIONs motor, the register *MaxCurrent_MTRDT* can be set at a value equal to the 130 % of the parameter *RatedCurrent_MTRDT.* For different motors, it is necessary to ask the maximum permissible current to the motor's manufacturer.

If the set current value exceeds the maximum device capacity, the latter shall prevail.

Address	Name			Mnemonic
431Ah	*RatedTorque_MTRDT*			

Data Type	Access Type	PD Mapping	Note
u16	Read/Write		🖫 Valid only if *CMC_MTRDT* = 0. It cannot be modified with the drive enabled.

Default Value	Minimum	Maximum	Unit
50	1	60000	10 mNm (Ex. 180 = 1.8 Nm)

Description

It allows to set the rated torque of the motor connected to the drive.

> For optimal operation, it is very important to carefully set this register so that it indicates the motor rated torque when supplied at the rated current set through the register *RatedCurrent_MTRDT*.

Address	Name			Mnemonic
431Bh	*MaxSpeed_MTRDT*			

Data Type	Access Type	PD Mapping	Note
u16	Read/Write		🖫

Default Value	Minimum	Maximum	Unit
30000	1	30000	0.1 rpm (Ex. 668 = 66.8 rpm)

Description

The register sets the maximum motor speed.

This value can never be exceeded and prevails over any other setting, whatever the chosen operating mode (also *Torque mode*).

7.4 Motor Incremental Encoder Data

Address	Name		Mnemonic
4331ₕ	*Configuration_ENCMTR*		
Data Type	Access Type	PD Mapping	Note
u8	Read/Write		🖫 It cannot be modified with the drive enabled.
Default Value	Minimum	Maximum	Unit
0			

Description

It allows you to set the encoder management mode. The register is split into 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	Non used	
3	Power Supply	
	0	Encoder power supply off
	1	Encoder power supply on
7..4	Filter If the encoder signals are affected by electric noise, it is possible to use the digital filter present in the drive to eliminate the pulses shorter than a predefined value.	
	3	250 ns
	5	500 ns
	7	1 µs
	9	2 µs
	12	4 µs
	15	8 µs
	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 1 μs, you can apply a maximum frequency of 500 kHz (1+1=2 μs i.e., 500 kHz).

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

Address 4332ₕ	CPR_ENCMTR		Name		Mnemonic
	Data Type u16	Access Type Read/Write	PD Mapping		Note 💾 It cannot be modified with the drive enabled.
	Default Value 0	Minimum	Maximum		Unit Encoder pulses per revolution
	Description				

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

By setting the value to 0 the encoder input is disables. The register *CPR_ENCMTR* must be configured with the value 0 if the encoder is not present or used.

The drive is able to count every pulse edge in order to obtain a resolution 4 times higher than the encoders 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 UNIMOTIONs motor code.

Address 4333ₕ	PhasingTime_ENCMTR		Name		Mnemonic
	Data Type u16	Access Type Read/Write	PD Mapping		Note 💾
	Default Value 500	Minimum	Maximum		Unit 1 ms (1500 = 1.5 ms)
	Descrizione				

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.

7.5 SSI Encoder Data

Address				Name	Mnemonic
4351ₕ	Configuration_ENCSSI				

	Data Type	Access Type	PDO Mapping	Note
	u8	Read/Write		🖫 It cannot be modified with the drive enabled.

	Default Value	Minimum	Maximum	Unit
	0			

Description

It allows to enable the SSI encoder and set its 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 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 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	Read/Write		🖫 It cannot be modified with the drive enabled.	
	Default Value	Minimum	Maximum	Unit	
	25	8	31	bit	
	Description				
	It allows to set the number of bits that make up the data frame of the SSI encoder. The *start* bit must not be counted.				

Address	Name			Mnemonic	
4353ₕ	TurnsBits_ENCSSI				
	Data Type	Access Type	PDO Mapping	Note	
	u8	Read/Write		🖫 It cannot be modified with the drive enabled.	
	Default Value	Minimum	Maximum	Unit	
	12		24	bit	
	Description				
	It allows to set the number of bits reserved for the full revolutions counting. In case of linear SSI encoder, set the register to 0.				

Address	Name			Mnemonic	
4354ₕ	SingleTurnBits_ENCSSI				
	Data Type	Access Type	PDO Mapping	Note	
	u8	Read/Write		🖫 It cannot be modified with the drive enabled.	
	Default Value	Minimum	Maximum	Unit	
	13	8	31	bit	
	Description				
	It allows to set the number of bits reserved for the position on the revolution. In case of linear encoder, set the 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	Read/Write		🖫 It cannot be modified with the drive enabled.	
	Default Value	Minimum	Maximum	Unit	
	0		20	bit	
	Description				
	It allows to set the position of the most significant bit (MSB) inside the frame.				

The position is counted from left to right with the first bit of the frame that has position 0.

For example, if the most significant bit of the quota falls on the third bit of the frame (starting from left and not counting the *start* bit), the *MSB_Offset_ENCSSI* register must be set to the value 3.

Address	Name			Mnemonic	
4356h	*FormingTime_ENCSSI*				
	Data Type	Access Type	PDO Mapping	Note	
	u16	Read/Write		🖫 It cannot be modified with the drive enabled.	
	Default Value	Minimum	Maximum	Unit	
	150			1 us (150 = 150 µs)	
	Description				
	It allows to set the time required by the SSI encoder to prepare a new quota. The drive waits at least this time between one reading and the next.				

Address	Name			Mnemonic	
4357h	*SetupTime_ENCSSI*				
	Data Type	Access Type	PDO Mapping	Note	
	u16	Read/Write		🖫 It cannot be modified with the drive enabled.	
	Default Value	Minimum	Maximum	Unit	
	1200			1 ms (1200 = 1.2 ms)	
	Description				
	It allows to set the time required by the SSI encoder to complete the initializations and provide the first valid quota. After the SSI encoder is enabled and after power has been supplied (if the SSI encoder is configured with bit 3 of the *Configuration_ENCSSI* register to 1), the 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	Read/Write		🖫 It cannot be modified with the drive enabled.	
	Default Value	Minimum	Maximum	Unit	
	10000				
	Description				
	It allows to set the numerator of the formula used to calculate the position expressed in 0.0001 rev, 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	Read/Write		🖫 It cannot be modified with the drive enabled.
	Default Value	Minimum	Maximum	Unit
	10000			
				Description

It allows to set the denominator of the formula used to calculate the position expressed in 0.0001 rev, 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	Read/Write		🖫 It cannot be modified with the drive enabled.
	Default Value	Minimum	Maximum	Unit
	10000			
				Description

It allows to set the offset subtracted from the quota read by the SSI encoder to calculate the position express in 0.0001 rev.

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	Read/Write		🖫 It cannot be modified with the drive enabled.
	Default Value	Minimum	Maximum	Unit
	500			1 ms (1500 = 1.5 ms)

		Description

It allows to set the time required to phase the encoder.

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

Following the enabling, the motor could move and it is necessary to wait for the load to stabilize before proceeding with the encoder phasing. The *PhasingTime_ENCSSI* register allows to set this time.

7.6 Holding Brake Setup

Address		Name			Mnemonic
4361ₕ	*Option_HBRKS*				
	Data Type	Access Type	PDO Mapping		Note
	u16	Read/Write	💾		
	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 drive.
	1	Handled, the holding brake is handled through the drive (connected to one output).
	When the bit is set to 1 upon activation, the drive waits for the brake disengaging time specified through the register *Release_Time_HBRKS* before going into the operating status. Upon disabling the drive waits for the *Application_Time_HBRKS* before switching off the motor.	
1	Automatic/Manual	
	0	Automatic; the drive commands 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	

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 drive (if active through bit 1 = 0).

In the following description, "active signal" means the active state of the digital input chosen for the manual brake control or the bit 0 of the register *Control_HBRKC* set to 1 (the two signals are in logic OR between them).

8..11	0	None
	1	Release; with an active signal, the brake is always released independently from the drive control.
	2	Engage; with an active signal, the brake is always engaged independently from the drive control.
	3	Release/Engage; with an active signal, the brake is always released. With an inactive signal, the brake is always engaged independently from the drive control.
	4	Shared Release; on the signal rising edge (transition from inactive to active), the brake is released. Also, the drive 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 drive 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 drive can release or engage the brake when necessary.

Address			Name		Mnemonic
4362h	*Application_Time_HBRKS*				
	Data Type	Access Type	PDO Mapping		Note
	u16	Read/Write	🖫		
	Default Value	Minimum	Maximum		Unit
	200		10000	ms (ex. 250 = 250 ms)	

		Description
	It allows you to set the time required for the brake to completely engage to ensure the maximum braking torque.	
	When the bit 0 of the *Option_HBRKS* register is set to 1 and the motor is requested to be disabled, the drive waits for the time specified in the register *Application_Time_HBRKS*, after the brake has been engaged, before disconnecting the motor.	

Address	Name			Mnemonic	
4363ₕ	*Release_Time_HBRKS*				
	Data Type	Access Type	PDO Mapping		Note
	u16	Read/Write		💾	
	Default Value	Minimum	Maximum		Unit
	200		10000	ms (ex. 250 = 250 ms)	
	Description				
	It allows you to set the time necessary for the brake to completely disengage to ensure the minimum braking torque. When the bit 0 of the register *Option_HBRKS* is set to 1 and the motor is requested to be enabled, the drive 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	Name			Mnemonic	
4364ₕ					
	Data Type	Access Type	PDO Mapping		Note
	u32	Read/Write		💾	
	Default Value	Minimum	Maximum		Unit
	Description				
	Reserved, do not use.				

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

7.7 Current Regulation

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

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 drive supplies the motor a current 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	Name		Mnemonic
43A2ₕ	*Gain_CRRG*		

	Data Type	Access Type	PD Mapping	Note
	u8	Read/Write	RPD	💾

	Default Value	Minimum	Maximum	Unit
	100	50	200	% (Ex. 120 = 120 %)

Description

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

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

7.8 Motion Setup

Address	Name		Mnemonic
4411ₕ	*Feedback_MTNSTP*		

	Data Type	Access Type	PD Mapping	Note
	u8	Read/Write	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 It allows you to enable closed-loop operation using the motors encoder for feedback. | Value	Description
0	Disabled
1	Closed loop (feedback enabled)
2	Following control | It allows to enable the closed-loop operation using the motors 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 the 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	
4412ₕ	*CurrentEnableRamp_MTNSTP*				
	Data Type	Access Type	PD Mapping	Note	
	u16	Read/Write		🖫	
	Default Value	Minimum	Maximum	Unit	
	10		2000	ms (ex. 250 = 250 ms).	
	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.				

7.9 Position Loop

Address	Name			Mnemonic	
4441ₕ	*Kp_PSTNLP*				
	Data Type	Access Type	PD Mapping	Note	
	u16	Read/Write	RPD	🖫	
	Default Value	Minimum	Maximum	Unit	
	300	2	3000		
	Description				
	It allows to set the proportional gain of the position control loop. Too low value, can causes a great position error while an excessive value can make the system unstable.				

7.10 Error Class Setup

Address	Name			Mnemonic	
4481ₕ	*Following_Error_ERRCS*				
	Data Type	Access Type	PD Mapping	Note	
	u8	Read/Write		🖫	
	Default Value	Minimum	Maximum	Unit	
	6				
	Description				
	It allows you to define the emergency class generated in case of the following error. The following are the values that can be used:				

Class	Description	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.

7.11 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 *Controlword.* Please note that the same input can be associated with more functions and actions.

Address	Name			Mnemonic
4812ₕ	*FaultReset_DIA*			

	Data Type	Access Type	PD Mapping	Note
	u16	Read/Write	💾	
	Default Value	Minimum	Maximum	Unit
	0502ₕ			

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 the *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
4813ₕ	*PositiveLimit_DIA*			

	Data Type	Access Type	PD Mapping	Note
	u16	Read/Write	💾	
	Default Value	Minimum	Maximum	Unit
	0207ₕ			

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
4814h	*NegativeLimit_DIA*				
	Data Type	Access Type	PD Mapping		Note
	u16	Read/Write		💾	
	Default Value	Minimum	Maximum		Unit
	0206h				

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 the *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
4815h	*Home_DIA*				
	Data Type	Access Type	PD Mapping		Note
	u16	Read/Write		💾	
	Default Value	Minimum	Maximum		Unit
	0205h				

		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 the *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	Read/Write		💾

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 the *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

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

Address	Name	Mnemonic
481Eh	*TouchProbe_1_DIA*	

Data Type	Access Type	PDO Mapping	Note
u16	Read/Write	💾	

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 state		Low byte, input number	
Value	Description	Value	Description
2	Active Input	0	Digital Input 0
		1	Digital Input 1
		2	Digital Input 2
		3	Digital Input 3
		4	Digital Input 4
		5	Digital Input 5
		6	Digital Input 6
		7	Digital Input 7

Address	Name	Mnemonic
481Fh	*TouchProbe_2_DIA*	

Data Type	Access Type	PDO Mapping	Note
u16	Read/Write	💾	

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 state		Low byte, input number	
Value	Description	Value	Description
2	Active Input	0	Digital Input 0
		1	Digital Input 1
		2	Digital Input 2
		3	Digital Input 3
		4	Digital Input 4
		5	Digital Input 5
		6	Digital Input 6
		7	Digital Input 7

7.12 Digital Outputs Assignment

Address	Name				Mnemonic
4831$_h$	*Source_DOA_0*				

	Data Type	Access Type	PD Mapping	Note	
	u16	Read/Write		💾	

	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 a 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	Read/Write	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

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

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

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

Address	Name				Mnemonic
4851h	*Source_DOA_2*				
	Data Type	Access Type	PD Mapping		Note
	u16	Read/Write	RPD	💾	
	Default Value	Minimum	Maximum		Unit
	006Ah				

Description

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

Value	Description
0000h	General Purpose
0008h	Active
8008h	Inactive
0040h	Fault
8040h	No Fault
0041h	Operational enabled
8041h	No Operational enabled
0042h	Quick stop active
8042h	No Quick stop active
0044h	Holding Brake
8044h	No Holding Brake
0050h	Positive Movement
8050h	No Positive Movement
0051h	Negative Movement

8051ₕ	No Negative Movement
0060ₕ…006Fₕ	*Statusword* bit n = 1 (0060ₕ=bit0, 0061ₕ=bit1, etc.)
8060ₕ…806Fₕ	*Statusword* bit n = 0 (8060ₕ=bit0, 8061ₕ=bit1, etc.)

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

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

7.13 Device Status

Address	Name			Mnemonic
5141ₕ	*BridgeTemperature_DVSTS*			
	Data Type	Access Type	PD Mapping	Note
	i16	Read only	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
5143ₕ	*PowerVoltage_DVSTS*			
	Data Type	Access Type	PD Mapping	Note
	u16	Read only	TPD	
	Default Value	Minimum	Maximum	Unit
				0.1 Vdc (Ex. 482 = 48.2 Vdc)
	Description			
	It indicates the voltage of the power bus.			

7.14 Digital Inputs Value

Address	Name			Mnemonic
5211h	*Inputs_DIV*			
	Data Type	Access Type	PD Mapping	Note
	u16	Read only	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.

7.15 Digital Outputs Value

Address	Name				Mnemonic
5231ₕ	Outputs_DOV				

	Data Type	Access Type	PD Mapping		Note
	u16	Read/Write	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.

> ℹ️ If the output is assigned to the drives' 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
5232ₕ	SetOutput_DOV				

	Data Type	Access Type	PD Mapping		Note
	U8	Write only			
	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 81ₕ the bit 1 of the register *Outputs_DOV* will be set to 0, writing instead the value 01ₕ the same bit will be set to 1. Instead writing the register *SetOutput_DOV* with value 00ₕ 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. <table><tr><td>Value</td><td>Description</td></tr><tr><td>0</td><td>Bit 0</td></tr><tr><td>1</td><td>Bit 1</td></tr><tr><td>…</td><td>Bit n</td></tr><tr><td>15</td><td>Bit 15</td></tr></table>
4..6	Not used, set to 0
7	Logic level to set in the bit of the register *Outputs_DOV* <table><tr><td>Value</td><td>Description</td></tr><tr><td>0</td><td>Bit = 1</td></tr><tr><td>1</td><td>Bit = 0</td></tr></table>

If the selected bit of the register *Outputs_DOV* corresponds to an output assigned to the drives' 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.

7.16 Analog Inputs Value

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

7.17 Analog Outputs Value

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

7.18 Motor Configuration

Address	Name			Mnemonic
5312h	*CurrentMin_MTRCNF*			
	Data Type	Access Type	PD Mapping	Note
	u16	Read/Write	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 4 Arms and a minimum current equal to 25 %, the drive will never supply less than 1 Arms 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	Read/Write	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 4 Arms and a minimum current equal to 75 %, the drive will never supply less than 3 Arms 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	Read/Write		💾	
	Default Value	Minimum	Maximum	Unit	
	500	2	10000	1 ms (Ex. 3500 = 3.5 s)	
	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.				

7.19 Motor Value

Address	Name			Mnemonic	
531B$_h$	Current_MTRV				
	Data Type	Access Type	PDO Mapping	Note	
	i16	Read only	TPD		
	Default Value	Minimum	Maximum	Unit	
				10 mArms (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	Read only	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.				

7.20 Motor Incremental Encoder Value

Address	Name				Mnemonic
5331h	Status_ENCMTRV				

Data Type	Access Type	PDO Mapping	Note
u8	Read only	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
5332h	PositionRaw_ENCMTRV				

Data Type	Access Type	PDO Mapping	Note
u32	Read only	TPDO	

Default Value	Minimum	Maximum	Unit

Description

It indicates the cyclical position on the revolution of the motor incremental encoder.

Address	Name				Mnemonic
5333h	Frequency_ENCMTRV				

Data Type	Access Type	PDO Mapping	Note
u32	Read only	TPDO	

Default Value	Minimum	Maximum	Unit
			Hz (Ex. 4365 = 4.365 kHz)

Description

It indicates the frequency of the pulses generated by the motors' incremental encoder.

7.21 3338h SSI Encoder Value

Address				Name			Mnemonic
5339h	*Status_ENCSSIV*						
		Data Type		Access Type	PDO Mapping		Note
		u8		Read only	TPDO		
		Default Value		Minimum	Maximum		Unit
							10 mArms (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 ℹ 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 bit 3 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	Read only	TPDO	
	Default Value	Minimum	Maximum	Unit
				0.0001 rev (Ex. 5000 = 0.5000 rev)
	Description			
	It indicates the position expressed in 0.0001 rev obtained from the SSI encoder.			
	The formula used to obtain the position expressed in 0.0001 rev 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	Read only	TPDO	
	Default Value	Minimum	Maximum	Unit
	Description			
	It indicates the quota read by the SSI encoder.			

7.22 Holding Brake Control

Address	Name				Mnemonic
5361ₕ	*Control_HBRKC*				
	Data Type	**Access Type**	**PDO Mapping**		**Note**
	u8	Read/Write	RPD	💾	
	Default Value	**Minimum**	**Maximum**		**Unit**
	0				

Description

It allows you to manually control the holding brake.

The following table resumes the use of the register bits.

Bit	Description
0	Manual control It allows you to manually control the holding brake. <table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Inactive manual control signal</td></tr><tr><td>1</td><td>Active manual control signal</td></tr></table> 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 7.6 Holding Brake Setup. The bit 0 is in logic OR with the digital input chosen for the manual control of the brake.
1..7	Not used, set to 0.

Address			Name		Mnemonic
5362_h	*Status_HBRKC*				

	Data Type	Access Type	PDO Mapping		Note
	u8	Read only	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 you to know if the *Release_Time_HBRKS* has passed and the brake is then in the released state. **Value / Description** 0 — The brake is not released 1 — The brake is released
2	Engaged It allows you to know if the *Application_Time_HBRKS* has passed and the brake is then in the engaged state. **Value / Description** 0 — The brake is not engaged 1 — The brake is engaged

7.23 Profile Torque Configuration

Address	Name					Mnemonic
5471ₕ	*Mode_PTCNF*					
	Data Type	Access Type	PD Mapping			Note
	u8	Read/Write	RPD	💾		
	Default Value	Minimum	Maximum			Unit
	0					

Address: 5471_h

	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 any load. Setting the bit 0 of the register *Mode_PTCNF* to 1, the drive limits the motor maximum speed to the value set through the register *TargetVelocity*.

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.	

7.24 StatusDWord and ControlDWord

Address	Name		Mnemonic
6011ₕ	*StatusDWord*		

	Data Type	Access Type	PD Mapping		Note
	u16	Read only	TPD		

	Default Value	Minimum	Maximum		Unit

Description

It allows you to know the device status.

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

<table>
<tr><td colspan="4" align="center">StatusDWord</td></tr>
<tr><td>bit</td><td>Description</td><td>Bit</td><td>Description</td></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><td>bit 26..24</td><td colspan="2">Operation Mode</td></tr><tr><td></td><td>001</td><td>Position</td></tr><tr><td></td><td>010</td><td>Velocity</td></tr><tr><td></td><td>011</td><td>Torque</td></tr><tr><td></td><td>111</td><td>Homing</td></tr></table>
The bit combination 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>Enable</td></tr>
<tr><td>16</td><td>Reserved, ignore the value</td><td>0</td><td><table><tr><td>bit1</td><td>bit 0</td><td>Description</td></tr><tr><td>0</td><td>0</td><td>Disabled</td></tr><tr><td>0</td><td>1</td><td></td></tr><tr><td>1</td><td>0</td><td></td></tr><tr><td>1</td><td>1</td><td>Enabled</td></tr></table></td></tr>
</table>

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	Read/Write	RPD		
	Default Value	Minimum	Maximum		Unit

Description

It allows you 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	<table><tr><td>bit 26..24</td><td>Operation Mode</td></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 bits combination not in the table must be avoided.	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	<table><tr><td>bit 1</td><td>bit 0</td><td>Description</td></tr><tr><td>0</td><td>0</td><td>Disabled</td></tr><tr><td>0</td><td>1</td><td></td></tr><tr><td>1</td><td>0</td><td></td></tr><tr><td>1</td><td>1</td><td>Enabled</td></tr></table>

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

7.25 Errors

Address	Name				Mnemonic
6081ₕ	*ErrorCode*				
	Data Type	Access Type	PD Mapping		Note
	u16	Read only	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	Read only	TPD		
	Default Value	Minimum	Maximum		Unit
	Description				
	It contains additional information on the last error occurred in the device.				

7.26 Actual Motion Values

Address	Name			Mnemonic
6111ₕ	*ActualTorque*			
	Data Type	Access Type	PD Mapping	Note
	i16	Read only	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	Read only	TPD	
	Default Value	Minimum	Maximum	Unit
				0.1 rpm (Ex. 3850 = 385.0 rpm)
	Description			
	The register provides the actual velocity value. The value is given in 0.1 rpm.			

Address	Name			Mnemonic
6113ₕ	*ActualPosition*			
	Data Type	Access Type	PD Mapping	Note
	i32	Read only	TPD	
	Default Value	Minimum	Maximum	Unit
				0,0001 rev (Ex. 5000 = 0.5000 rev)
	Description			
	The register provides the actual position value. The value is given in 1/10000 of revolution.			

Address	Name			Mnemonic
6122ₕ	*ActualPositionError*			
	Data Type	Access Type	PD Mapping	Note
	i32	Read only	TPD	
	Default Value	Minimum	Maximum	Unit
				0,0001 rev (Ex. 20000 = 2.0000 rev)
	Description			
	This register allows you to read the following position error value. The value is given in 1/10000 of revolution.			

7.27 Target Motion Values

Address	Name			Mnemonic
6211h	**TargetTorque**			
	Data Type	Access Type	PD Mapping	Note
	i16	Read/Write	RPD	💾
	Default Value	Minimum	Maximum	Unit
	0	-1300	1300	0.1 % (Ex. 405 = 40.5 %)
	Description			
	This register allows you to set the desired target torque in the *Torque* mode. In the *Velocity* or *Position* mode, it sets the maximum torque value of the motor, if drive operates in the closed-loop mode. In the open-loop operation mode, *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 the 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	Name			Mnemonic
6212h	**TargetVelocity**			
	Data Type	Access Type	PD Mapping	Note
	i32	Read/Write	RPD	💾
	Default Value	Minimum	Maximum	Unit
	0			0.1 rpm (Ex. 1000 = 100.0 rpm)
	Description			
	The register allows you 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 0.1 rpm.			

Address	Name			Mnemonic
6213h	**TargetPosition**			
	Data Type	Access Type	PD Mapping	Note
	i32	Read/Write	RPD	💾
	Default Value	Minimum	Maximum	Unit
	0			0.0001 rev (Ex. 50000 = 5.0000 rev)
	Description			
	It allows you to set the motor target position in the *Position* mode. The drives 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	
6215h	Acceleration				
	Data Type	Access Type	PD Mapping		Note
	u32	Read/Write	RPD	💾	
	Default Value	Minimum	Maximum		Unit
	100	5	300000	rpm (Ex. 100 = 100 rpm)	
	Description				
	It allows you to set the acceleration value used during the execution of a motion profile. The value is given in rpm/s.				

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

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

Address	Name			Mnemonic	
6218h	TorqueSlope				
	Data Type	Access Type	PD Mapping		Note
	u32	Read/Write	RPD	💾	
	Default Value	Minimum	Maximum		Unit
				0.1 %/s (Ex. 558 = 55.8 %/s)	
	Description				
	This register indicates the configured rate of change of torque. The value is given in thousandths of the nominal motor torque per second.				

7.28 Limits

Address	Name				Mnemonic
6221h	*MaxTorque*				
	Data Type	Access Type	PD Mapping		Note
	u16	Read/Write	RPD	💾	
	Default Value	Minimum	Maximum		Unit
	0	-1300	1300		0.1 % (Ex. 405 = 40.5 %)
	Description				
	The register allows you to set the maximum torque value that can be supplied by the motor when the drive operates in the closed-loop mode.				
	In the open-loop operation mode, *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
6225h	*MaxSpeed*				
	Data Type	Access Type	PD Mapping		Note
	u32	Read/Write	RPD	💾	
	Default Value	Minimum	Maximum		Unit
	30000	1	30000		0.1 rpm (Ex. 5000 = 500.0 rpm)
	Description				
	This register allows you to set the maximum allowed speed in either direction during a profiled motion.				
	The value is given in 0.1 rpm.				

Address	Name				Mnemonic
6229h	*MinPosition*				
	Data Type	Access Type	PD Mapping		Note
	i32	Read/Write	RPD	💾	
	Default Value	Minimum	Maximum		Unit
	0				0.0001 rev (Ex. -1000000 = -100 rev)
	Description				
	This register allows you 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 be set to 0.				

Address	Name			Mnemonic	
622A$_h$	*MaxPosition*				
	Data Type	Access Type	PD Mapping	Note	
	i32	Read/Write	RPD	💾	
	Default Value	Minimum	Maximum	Unit	
	0			0.0001 rev (Ex. 1000000 = 100 rev)	
	Description				

This register allows you to set the maximum position software limit.

The value is given in 1/10000 of revolution.

> To disable the position software limits, the register *MinPosition and MaxPosition* must be set to 0.

7.29 Validity and Error Windows

Address	Name			Mnemonic	
6401$_h$	*WindowTorque*				
	Data Type	Access Type	PD Mapping	Note	
	u16	Read/Write	RPD	💾	
	Default Value	Minimum	Maximum	Unit	
	100			0.1 rpm (Ex. 1000 = 100.0 rpm)	
	Description				

This register allows you 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 bit 13 *TorqueReached* of StatusDWord is set to 1.

The value is given in 0.1 rpm.

Address	Name			Mnemonic	
6402$_h$	*WindowTorqueTime*				
	Data Type	Access Type	PD Mapping	Note	
	u16	Read/Write	RPD	💾	
	Default Value	Minimum	Maximum	Unit	
	0		30000	1 ms (Ex. 200 = 200 ms)	
	Description				

It allows you 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
6411ₕ	*WindowVelocity*			

	Data Type	Access Type	PD Mapping	Note
	u16	Read/Write	RPD	💾

	Default Value	Minimum	Maximum	Unit
	100			0.1 rpm (Ex. 1000 = 100.0 rpm)

Description

This register allows you 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 bit 14 *VelocityReached* in the *StatusDWord* is set to 1.

The value is given in 0.1 rpm.

Address	Name			Mnemonic
6412ₕ	*WindowVelocityTime*			

	Data Type	Access Type	PD Mapping	Note
	u16	Read/Write	RPD	💾

	Default Value	Minimum	Maximum	Unit
	0		30000	1 ms (Ex. 200 = 200 ms)

Description

It allows you 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
6413ₕ	*WindowVelocityError*			

	Data Type	Access Type	PD Mapping	Note
	i32	Read only	RPD	💾

	Default Value	Minimum	Maximum	Unit
	1000	0	30000	0.1 rpm (Ex. 1000 = 100.0 rpm)

Description

It allows you 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 bit 18 *VelocityError* in the *StatusDWord* is set to 1.
If the value of the register *WindowVelocityError* is set to FFFFFFFFₕ the following error control is disabled.

> The drive is able to know the actual motor speed only if the motor has an encoder and the drive is configured in the closed-loop operation mode.

The value is given in 0.1 rpm.

Address	Name				Mnemonic
6414ₕ	*WindowVelocityErrorTimeOut*				
	Data Type	Access Type	PD Mapping		Note
	u16	Read/Write	RPD	💾	
	Default Value	Minimum	Maximum		Unit
	10		30000		1 ms (Ex. 280 = 280 ms)

Description

It allows you to set the time taken to validate the speed error condition (speed error greater than *WindowVelocityError* in absolute value).

> The drive is able to know the actual motor speed only if the motor has an encoder and the drive is configured in the closed-loop operation mode.

Address	Name				Mnemonic
6415ₕ	*WindowZeroSpeed*				
	Data Type	Access Type	PD Mapping		Note
	u16	Read/Write	RPD	💾	
	Default Value	Minimum	Maximum		Unit
	60				0.1 rpm (Ex. 500 = 50.0 rpm)

Description

This register allows you 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 bit 12 *Standstill* is set to 1.

The value is given in 0.1 rpm.

Address	Name				Mnemonic
6416ₕ	WindowZeroSpeedTime				
	Data Type	Access Type	PD Mapping		Note
	u16	Read/Write	RPD	💾	
	Default Value	Minimum	Maximum		Unit
	0		30000		1 ms (Ex. 200 = 200 ms)

Description

It allows you 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	Read/Write	RPD	💾
	Default Value	Minimum	Maximum	Unit
	10			0.0001 rev (Ex. 20000 = 2.0000 rev)
	Description			

The register allows you 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 bit 14 *VelocityReached of the StatusDWord* is set to 1.

The value is given in 1/10000 of revolution.

Address	Name			Mnemonic
6422ₕ	*WindowPositionTime*			
	Data Type	Access Type	PD Mapping	Note
	u16	Read/Write	RPD	💾
	Default Value	Minimum	Maximum	Unit
	0		30000	1 ms (Ex. 600 = 0.6 s)
	Description			

It allows you 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	
6423ₕ	*WindowPositionError*				
	Data Type	Access Type	PD Mapping	Note	
	u32	Read/Write	RPD	🖫	
	Default Value	Minimum	Maximum	Unit	
	10000			0.0001 rev (Ex. 20000 = 2.0000 rev)	
	Description				

This register allows you 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 bit 19 *PositionError* of the *StatusDWord* is set to 1.

If the value of the register *WindowPositionError* is set to FFFFFFFFₕ the control on the following error is disabled.

The value is given in 1/10000 of revolution.

Address	Name			Mnemonic	
6424ₕ	*WindowPositionErrorTimeOut*				
	Data Type	Access Type	PD Mapping	Note	
	u16	Read/Write	RPD	🖫	
	Default Value	Minimum	Maximum	Unit	
	10		30000	1 ms (Ex. 250 = 250 ms)	
	Description				

It allows you 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.

7.30 Homing

Address	Name		Mnemonic
6480ₕ	*HomingMethod*		

Data Type	Access Type	PD Mapping		Note
i8	Read/Write	RPD	💾	

Default Value	Minimum	Maximum		Unit

				Description

This register allows you 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 the negative limit switch is inactive, the motor will start rotating in the counterclockwise direction up to the limit switch, then reverse and home at the first index (Z or I pulse) outside the negative limit switch. At the start, if the negative limit switch is active, the motor will start rotating in the clockwise direction until it leaves the limit switch (becomes inactive), then home at the first index (Z or I pulse) outside the negative limit switch.	●			●
2	At the start, if the positive limit switch is inactive, the motor will rotate in the clockwise direction up to the limit switch, then reverse and home at the first index (Z or I pulse) outside the positive limit switch. At the start, if the positive limit switch is active, the motor will rotate in the counterclockwise direction until it leaves the limit switch, then home at the first index (Z or I pulse) outside the positive limit switch.		●		●
3	At the start, if the home switch is inactive, the motor will rotate in the clockwise direction up to home switch, then reverse and home at the first index (Z or I pulse) outside the home switch. At the start, if the home switch is active, the motor will rotate in the counterclockwise direction until it leaves the switch, then home at the first index (Z or I pulse) outside the home switch.			●	●
4	At the start, if the home switch inactive, the motor will rotate in the clockwise direction up to home switch, then home at the first index (Z or I pulse) inside the home switch.			●	●

		At the start, if the home switch is active, the motor will rotate in the counterclockwise direction until it leaves the switch, then reverse and home at the first index (Z or I pulse) inside the home switch.			
	5	At the start, if the home switch is active, the motor will rotate in the clockwise direction until it leaves the switch, then home at the first index (Z or I pulse) outside the home switch. At the start, if the home switch is inactive, the motor will rotate in the counterclockwise direction up to find the switch, then reverse and home at the first index (Z or I pulse) outside the home switch.		●	●
	6	At the start, if the home switch is active, the motor will rotate in the clockwise direction until it leaves the switch, then reverse and home at the first index (Z or I pulse) inside the home switch. At the start, if the home switch is inactive, the motor will rotate in the counterclockwise direction up to find the switch, then reverse and home at the first index (Z or I pulse) inside the home switch.		●	●
	7	At the start, if the home switch is inactive, the motor will rotate in the clockwise direction up to find the home switch or the positive limit switch. In case of home switch, reverse and home at the first index (Z or I pulse) outside the switch. In case of positive limit switch, reverse up to find the home switch, continue up to leave the home switch, then home at the first index (Z or I pulse) outside the switch. At the start, if the home switch is active, the motor will rotate in the counterclockwise direction up to leave the home switch, then home at the first index (Z or I pulse) outside the switch.	●	●	●
	8	At the start, if the home switch is inactive, the motor will rotate in the clockwise direction up to find the home switch or the positive limit switch. In case of home switch, it homes at the first index (Z or I pulse) 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 home at the first index (Z or I pulse) inside the switch. At the start, if the home switch active, the motor will rotate in the counterclockwise direction up to leave the home switch, then reverse and home at the first index (Z or I pulse) inside the switch.	●	●	●
	9	At the start, if the home switch is inactive, the motor will rotate in the clockwise direction up to find the home switch or the positive limit switch. In case of	●	●	●

	home switch, continue up to leave it then reverse and home at the first index (Z or I pulse) inside the switch. In case of positive limit switch, reverse up to find the home switch, then home at the first index (Z or I pulse) inside the switch. At the start, if the home switch is active the motor will rotate in the clockwise direction up to leave the home switch, then reverse and home at the first index (Z or I pulse) inside the home switch.				
10	At the start, if the home switch is inactive the motor will rotate in the clockwise direction up to find the home switch or the positive limit switch. In case of home switch, continue up to leave it then home at the first index (Z or I pulse) 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 home at the first index (Z or I pulse) outside the switch. At the start, if the home switch is active the motor will rotate in the clockwise direction up to leave the home switch, then home at the first index (Z or I pulse) outside the home switch.	●		●	●
11	At the start, if the home switch is inactive the motor will rotate in the counterclockwise direction up to find the home switch or the negative limit switch. In case of home switch, reverse and home at the first index (Z or I pulse) 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 home at the first index (Z or I pulse) outside the switch. At the start, if the home switch is active the motor will rotate in the clockwise direction up to leave the home switch, then home at the first index (Z or I pulse) outside the home switch.		●	●	●
12	At the start, if the home switch is inactive the motor will rotate in the counterclockwise direction up to find the home switch or the negative limit switch. In case of home switch, home at the first index (Z or I pulse) 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 home at the first index (Z or I pulse) inside the switch. At the start, if the home switch is active the motor will rotate in the clockwise direction up to leave the home switch, then reverse and home at the first index (Z or I pulse) inside the home switch.		●	●	●

	13	At the start, if the home switch is inactive the motor will rotate in the counterclockwise direction up to find the home switch or the negative limit switch. In case of home switch, continue up to leave it then reverse and home at the first index (Z or I pulse) inside the switch. In case of negative limit switch, reverse up to find the home switch, then home at the first index (Z or I pulse) inside the switch. At the start, if the home switch is active the motor will rotate in the clockwise direction up to leave the home switch, then reverse and home at the first index (Z or I pulse) inside the home switch.	●	●	●
	14	At the start, if the home switch is inactive the motor will rotate in the counterclockwise direction up to find the home switch or the negative limit switch. In case of home switch, continue up to leave it then home at the first index (Z or I pulse) 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 home at the first index (Z or I pulse) outside the switch. At the start, if the home switch is active the motor will rotate in the counterclockwise direction up to leave the home switch, then home at the first index (Z or I pulse) outside the switch.	●	●	●
	17	At the start, if the negative limit switch is inactive the motor will rotate in the counterclockwise direction up to find the limit switch, then reverse and home at the active/inactive switch transition. At the start, if the negative limit switch is active the motor will rotate in the clockwise direction with homing at the active/inactive switch transition.	●		
	18	At the start, if the positive limit switch is inactive the motor will rotate in the clockwise direction up to the limit switch, then reverse and home at the active/inactive switch transition. At the start, if the negative limit switch is active the motor will rotate in the counterclockwise direction with homing at the active/inactive switch transition.	●		
	19	At the start, if the home switch is inactive the motor will rotate in the clockwise direction up to the home switch, then reverse and home at the active/inactive switch transition. At the start, if the home switch is active the motor will rotate in the counterclockwise direction with homing at the active/inactive switch transition.		●	
	20	At the start, if the home switch is inactive the motor will rotate in the clockwise direction up to the home		●	

		switch, then home at the inactive/active switch transition. At the start, if the home switch is active the motor will rotate in the counterclockwise direction up to leave the switch, then reverse and home at the inactive/active switch transition.				
	21	At the start, if the home switch is active the motor will rotate in the clockwise direction up to leave the switch, then home at the active/inactive switch transition. At the start, if the home switch is inactive the motor will rotate in the counterclockwise direction up to find the switch, then reverse and home at the active/inactive switch transition.			●	
	22	At the start, if the home switch is active the motor will rotate in the clockwise direction up to leave the switch, then reverse and home at the inactive/active switch transition. At the start, if the home switch is inactive the motor will rotate in the counterclockwise direction up to find the switch, then home at the inactive/active switch transition.			●	
	23	At the start, if the home switch is inactive the motor will rotate in the clockwise direction up to find the home switch or the positive limit switch. In case of home switch, reverse and home 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 home at the active/inactive switch transition. At the start, if the home switch is active the motor will rotate in the counterclockwise direction up to leave the switch, then home at the active/inactive switch transition.	●		●	
	24	At the start, if the home switch is inactive the motor will rotate in the clockwise direction up to find the home switch or the positive limit switch. In case of home switch, home 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 home at the inactive/active switch transition. At the start, if the home switch is active the motor will rotate in the counterclockwise direction up to leave the home switch, then reverse and home at the inactive/active switch transition.	●		●	

	25	At the start, if the home switch is inactive the motor will rotate in the clockwise direction up to find the home switch or the positive limit switch. In case of home switch, continue up to leave it, then reverse and home 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 the home switch is active the motor will rotate in the clockwise direction up to leave the home switch, then reverse and home at the inactive/active switch transition.	●		●	
	26	At the start, if the home switch is inactive the motor will rotate in the clockwise direction up to find the home switch or the positive limit switch. In case of home switch, continue up to leave it, then home 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 home at the active/inactive switch transition. At the start, if the home switch is active the motor will rotate in the clockwise direction up to leave the home switch, then home at the active/inactive switch transition.	●		●	
	27	At the start, if the home switch is inactive the motor will rotate in the counterclockwise direction up to find the home switch or the negative limit switch. In case of home switch, reverse and home 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 home at the active/inactive switch transition. At the start, if the home switch is active the motor will rotate in the clockwise direction up to leave the home switch, then home at the active/inactive switch transition.		●	●	
	28	At the start, if the home switch is inactive the motor will rotate in the counterclockwise direction 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 home at the inactive/active switch transition. At the start, if the home switch is active the motor will rotate in the clockwise direction up to leave the home		●	●	

		switch, then reverse and home at the inactive/active switch transition.			
	29	At the start, if the home switch is inactive the motor will rotate in the counterclockwise direction up to find the home switch or the negative limit switch. In case of home switch, continue up to leave it then reverse and home at the inactive/active switch transition. In case of negative limit switch, reverse up to find the home switch, then home at the inactive/active switch transition. At the start, if the home switch is active the motor will rotate in the counterclockwise direction up to leave the home switch, then reverse and home at the inactive/active switch transition.	●	●	
	30	At the start, if the home switch is inactive the motor will rotate in the counterclockwise direction up to find the home switch or the negative limit switch. In case of home switch, continue up to leave it then home 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 home at the active/inactive switch transition. At the start, if the home switch is active the motor will rotate in the counterclockwise direction up to leave the home switch, then home at the active/inactive switch transition.	●	●	
	33	The motor will rotate in the counterclockwise direction and home at the first index (Z or I pulse) found.			●
	34	The motor will rotate in the clockwise direction and home at the first index (Z or I pulse) found.			●
	37	Homing at the actual position.			

Address	Name			Mnemonic
6481ₕ	*HomingSpeedForSwitch*			
	Data Type	Access Type	PD Mapping	Note
	u32	Read/Write	RPD	💾
	Default Value	Minimum	Maximum	Unit
	600		30000	0.1 rpm (Ex. 600 = 60.0 rpm)
				Description

It allows you to set the speed during the search for the limit switch in the homing procedure.

The value is given in 0.1 rpm.

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

Address	Name			Mnemonic
6483ₕ	*HomingOffset*			
	Data Type	**Access Type**	**PD Mapping**	**Note**
	i32	Read/Write	RPD	💾
	Default Value	**Minimum**	**Maximum**	**Unit**
				0.0001 rev (Ex. 20000 = 2.0000 rev)

Description

This register allows you 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.

7.31 Touch probe functionality

Address	Name			Mnemonic
6491ₕ	*Touch probe function*			
	Data Type	**Access Type**	**PDO Mapping**	**Note**
	u16	Read/Write	RPDO	💾
	Default Value	**Minimum**	**Maximum**	**Unit**
	0x0000			

Description

It allows you to configure the touch probe functions.
The bits of the low byte define the *Touch probe 1* functions while the high byte contains the bits related to *Touch probe 2*.

The following table shows the use of the bits and the allowed values:

	Touch probe 2				*Touch probe 1*	
Bit	**Description**			**Bit**	**Description**	
15, 14	Not used. Set to 0.			7, 6	Not used. Set to 0.	
13	0	Switch off sampling at negative edge.		5	0	Switch off sampling at negative edge.
	1	Enable sampling at negative edge.			1	Enable sampling at negative edge.
12	0	Switch off sampling at positive edge.		4	0	Switch off sampling at positive edge.
	1	Enable sampling at positive edge.			1	Enable sampling at positive edge.
11, 10				3, 2		

	00	Trigger with touch probe 2 input.		00	Trigger with touch probe 1 input.
	01	Trigger with Encoder Index.		01	Trigger with Encoder Index.
9	0	Trigger first event.	1	0	Trigger first event.
8	0	Switch off touch probe 2.	0	0	Switch off touch probe 1.
	1	Enable touch probe 2.		1	Enable touch probe 1.

> When the *Trigger with Encoder Index* mode is selected, the sampling on the positive edge must also be selected (*Enable sampling at positive edge*).

Address	Name				Mnemonic
6492h	*Touch probe status*				
	Data Type	Access Type	PDO Mapping		Note
	u16	Read only	TPDO		
	Default Value	Minimum	Maximum		Unit

Description

It indicates the touch probe status.
The bits of the low byte are related to the *Touch probe 1* while the bits in the high byte are related to the *Touch probe 2*.

The following table shows the meaning of each bit:

Touch probe 2		*Touch probe 1*	
Bit	Description	Bit	Description
15, 14	Reserved bit. Ignore the value.	7, 6	Reserved bit. Ignore the value.
13..11	0 — Reserved	5..3	0 — Reserved
10	0 — Touch probe 2 no negative edge value stored.	2	0 — Touch probe 1 no negative edge value stored.
	1 — Touch probe 2 negative edge position stored.		1 — Touch probe 1 negative edge position stored.

9	0	Touch probe 2 no positive edge value stored.	1	0	Touch probe 1 no positive edge value stored.
	1	Touch probe 2 positive edge position stored.		1	Touch probe 1 positive edge position stored.
8	0	Touch probe 2 is switched off.	0	0	Touch probe 1 is switched off.
	1	Touch probe 2 is enabled.		1	Touch probe 1 is enabled.

Address	Name	Mnemonic
6493ₕ	*Touch probe 1 positive edge*	

Data Type	Access Type	PDO Mapping	Note
i32	Read only	TPDO	

Default Value	Minimum	Maximum	Unit
			0.0001 rev (Ex. 25000 = 2.5000 rev)

Description

It contains the position read on the positive edge of the touch probe 1.

The value is given in 1/10000 per revolution.

Address	Name	Mnemonic
6494ₕ	*Touch probe 1 negative edge*	

Data Type	Access Type	PDO Mapping	Note
i32	Read only	TPDO	

Default Value	Minimum	Maximum	Unit
			0.0001 rev (Ex. 25000 = 2.5000 rev)

Description

It contains the position read on the negative edge of the touch probe 1.

The value is given in 1/10000 per revolution.

Address	Name				Mnemonic
6495h	*Touch probe 2 positive edge*				
	Data Type	Access Type	PDO Mapping		Note
	i32	Read only	TPDO		
	Default Value	Minimum	Maximum		Unit
					0.0001 rev (Ex. 25000 = 2.5000 rev)
	Description				
	It contains the position read on the positive edge of the touch probe 2. The value is given in 1/10000 of revolution.				

Address	Name				Mnemonic
6496h	*Touch probe 2 negative edge*				
	Data Type	Access Type	PDO Mapping		Note
	i32	Read only	TPDO		
	Default Value	Minimum	Maximum		Unit
					00001 rev (Ex. 25000 = 2.5000 rev)
	Description				
	It contains the position read on the negative edge of the touch probe 2. The value is given in 1/10000 of revolution.				

UNIMOTION

We cover all major markets. If you wish to
contact us, send us an enquiry and we will
be happy to assist you.

GERMANY

Unimotion GmbH
Waldstrasse 20
D - 78736 Epfendorf

T +49 (0) 7404 930 85 60
F +49 (0) 7404 930 85 61

www.unimotion.de
vertrieb@unimotion.de

NORTH AMERICA

Unimotion North America, Inc.
3952 Ben Hur Ave, Unit 5
Willoughby, OH 44094

T: +1 440-525-9106

www.unimotionusa.com
info@unimotionusa.com