CiA member

LAM Technologies electronic equipment

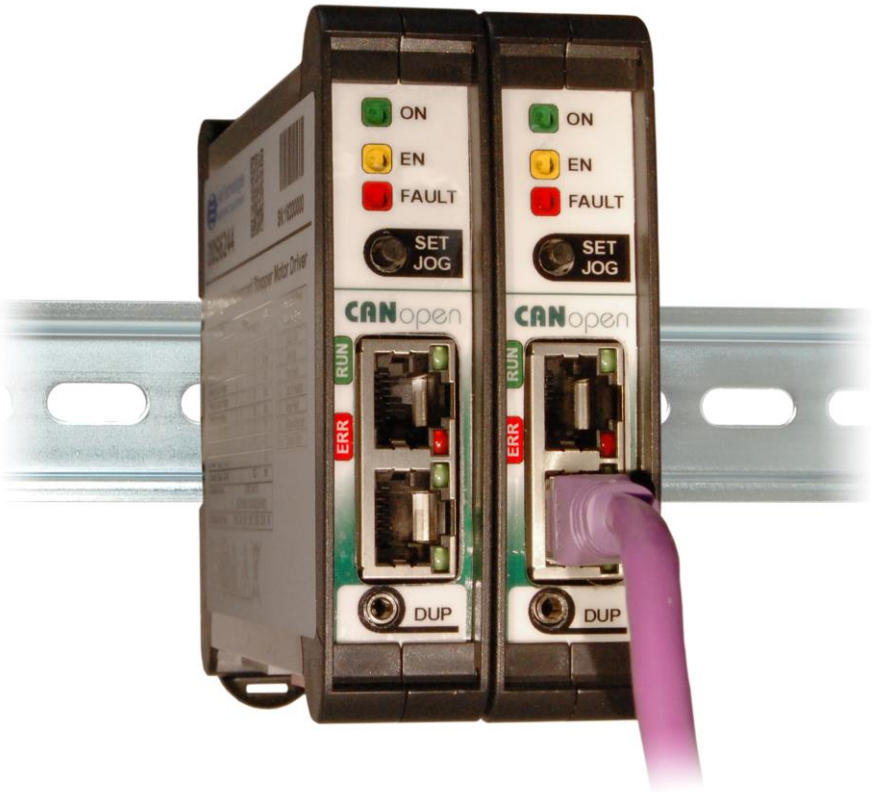
DDS6

CANopen

Vector Controlled

Stepper Motors Drives

DS301, DS402



User Manual

(Hardware rev. 1.00 Firmware rev. 0.22)

1 Introduction

The DDS6 drives series is realized in digital technology and drives the stepper motors with vector technique.

They are equipped with fieldbus in standard CANopen and implement the profiles /CiA301/ and /CiA402/. The fieldbus is isolated and does not require any auxiliary power supply.

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

Some models have the encoder input which allows the closed-loop motor control thus eliminating stall problem 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 setting of the node and bit rate address is through an easily accessible dip-switch. Alternatively, it is possible to use the free software *Omni Automation IDE*, running under Windows platform (Windows 7, Windows 8.1 and Windows 10 32bit or 64bit), which allows an assisted configuration and an accurate diagnostics. For the connections to the PC the UDP30 interface is needed.

1.1 Series

The series develops in 20 models different in functionalities and power.

Power Supply / Motor Current 24Vdc Auxiliary Power Supply	Digital I/O	Digital and Analog I/O A B Z Encoder	
20..50Vdc / 0.2..1.4Arms	**DDS6041**	**DDS6241**	DC Power Supply
20..50Vdc / 1.0..4.5Arms	**DDS6044**	**DDS6244**	
20..50Vdc / 2.0..10.0Arms	**DDS6048**	**DDS6248**	
24..90Vdc / 1.0..4.5Arms	**DDS6074**	**DDS6274**	
24..90Vdc / 2.0..10.0Arms	**DDS6078**	**DDS6278**	
16..36Vac / 0.2..1.4Arms	**DDS6041A**	**DDS6241A**	AC Power Supply
16..36Vac / 1.0..4.5Arms	**DDS6044A**	**DDS6244A**	
16..36Vac / 2.0..10.0Arms	**DDS6048A**	**DDS6248A**	
20..65Vac / 1.0..4.5Arms	**DDS6074A**	**DDS6274A**	
20..65Vac / 2.0..10.0Arms	**DDS6078A**	**DDS6278A**	

1.2 Terms, symbols and abbreviations

To indicate features common to a whole group of products the character "x" is used in place of any other character. For example, the term DDS6x44 implies the models DDS6044 and DDS6244.

The terms manual and document have the same meaning, moreover the words drive, device and product always refer to the DDS6 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 conditions that must be accurately evaluated and avoided. Failing to follow instructions marked with this symbol can be cause of serious damages to people, animals and things.

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

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

The characteristic names of registers, parameters, objects, modes, etc. defined in the CANopen Protocol Documents are 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 the 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	32,767
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		

Other abbreviations used::

Abbreviations	Description
AC, ac	Alternate current
AI	Analog input
AO	Analog output
COB	Communication Object
COD-ID	COB identifier
csp	Cyclic synchronous profile mode
cst	Cyclic synchronous torque mode
cstca	Cyclic synchronous torque mode with commutation angle
csv	Cyclic synchronous velocity mode
DC, dc	Direct current
FSA	Finite state automation
hm	Homing mode
DI	Digital input
DO	Digital output
ip	Interpolated position mode
NMT	Network management
OD	Object dictionary
PDO	Process data object
pp	Profile position mode
pv	Profile velocity mode
RMS, rms	Root mean square
RO, ro	Read-only
WO, wo	Write-only
RW, rw	Read-write
RPDO	Receive-PDO
RTR	Remote transmission request
SDO	Service data object
TPDO	Transmit-PDO
tq	Torque mode
vl	Velocity mode

1.3 Documents

The present manual applies to the standard series of DDS6 drives with Hardware and Firmware revisions as shown on the cover. Customized products or with a different Hardware or Firmware revision may have features and behaviors different to what herein described. It is technician and user's responsibility to use the documents appropriate to the products used.

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

The information contained herein replace any previously issued document.

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

LAM Technologies is a registered trade mark.

1.4 Contents of the pack

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

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

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 support@lamtechnologies.com. It is prohibited to use the products herein described if you are not sure to have understood their features and how to use and install them.

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

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

The user must make the 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 is considerably longer.

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

The high performance drive is able to 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, at installation 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 application 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 LAM Technologies.

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

Even if the products have been designed and realized with extreme care, there is always the possibility that under unpredictable circumstances and modes the products show malfunctions. Therefore, for any reason, the products described in this manual must be used in life support application and in all those cases in which the unexpected failure of the product could be cause of death or damage to people, things, animals or cause economic loss.

LAM Technologies reserves the right to make changes without prior notice to the products including design, technical specification, manufacturing process and functionality. LAM Technologies 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

LAM Technologies warrants the products described in this manual against defects in materials or workmanship for a period of 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 LAM Technologies. This warranty does not apply also in case the product's label has been removed or modified.
Any request for assistance must be sent to the purchase source of the product. In case of direct purchase from LAM Technologies, a returned material authorization number (RMA) must be obtained, before shipping the device, from support@lamtechnologies.com clearly specifying the product's code, the serial number, the problem found and the assistance required. The RMA number must be clearly written on each shipping document otherwise the parcel could be rejected. The customer shall be responsible for the packaging and shipping of the defective product to LAM Technologies and shipment must be made charges prepaid. The product inspected, repaired or replaced will be available to be collected at LAM Technologies'. In case of repair under warranty LAM Technologies can, at its own discretion, repair or replace the product. No cost for material or service will be charged in case of repair under warranty.

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

2 Installation

The DDS6 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 needed 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 expert staff and after having read and understood the instructions included herein.

2.1 Connectors

The DDS6 drives series has 4 connectors common to the whole family and other connectors specific for the models equipped with Encoder input. The common connectors are reserved to the power supply, the motor connection, the digital I/O and CANopen bus.

Connector	Function
CN1	Power supply
CN2	Motor
CN3	Digital I/Os
CN4	Motor Encoder (only for DDS62 series)
CN5	Analog I/Os (only for DDS62 series)
CN6	CANopen

2.1.1 CN1 – Power Supply, AC models

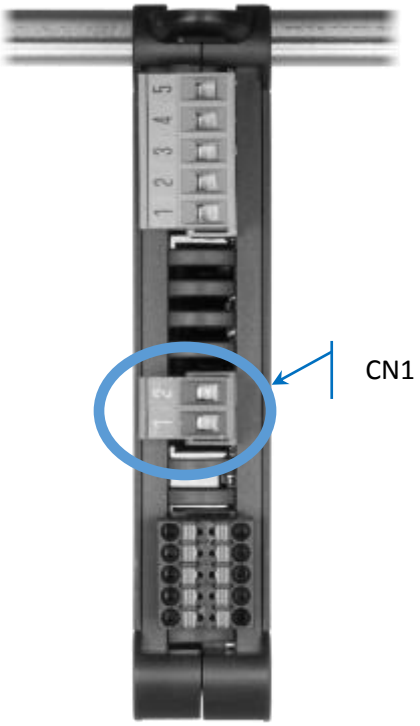
The AC supply drives are identified by the letter **A** placed at the end of the code (ex. DDS6274**A**). They integrate a rectifier bridge and the filter condensers necessary to rectify and filter the AC power supply voltage.

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

The overcoming of the *Vacbrk* voltage limit damages permanently the drive.

Do not supply the drive before the wiring is complete.

Do not connect the drive with the power supply on.



CN1 – Power supply	
Pin	Description
1	Vac, AC power supply voltage input
2	Vac, AC power supply voltage input

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

Model	Symbol	Description	Unit	Value		
				Min	Typ.	Max
DDS6x4xA	Vac	Nominal AC supply voltage	Vac	16	32	36
	Vacbrk	AC supply voltage causing the permanent damage	Vac			42
DDS6x7xA	Vac	Nominal AC supply voltage	Vac	20	55	65
	Vacbrk	AC supply voltage causing the permanent damage	Vac			75

The drive has protections that intervene when the supply voltage has a value such as to no longer guarantee the correct operation.

Model	Symbol	Description	Unit	Value		
				Min	Typ.	Max
DDS6x4xA	Vacl	Under voltage protection intervention threshold	Vac		15	
	Vach	Over voltage protection intervention threshold	Vac		38	
DDS6x7xA	Vacl	Under voltage protection intervention threshold	Vac		18	
	Vach	Over voltage protection intervention threshold	Vac		68	

For the connection with the transformer it is necessary to use a conductor with section adequate to the drive's calibration (for safety's it is better to use the max current supplied by the drive). The following table resumes the cable sections suggested for each drive:

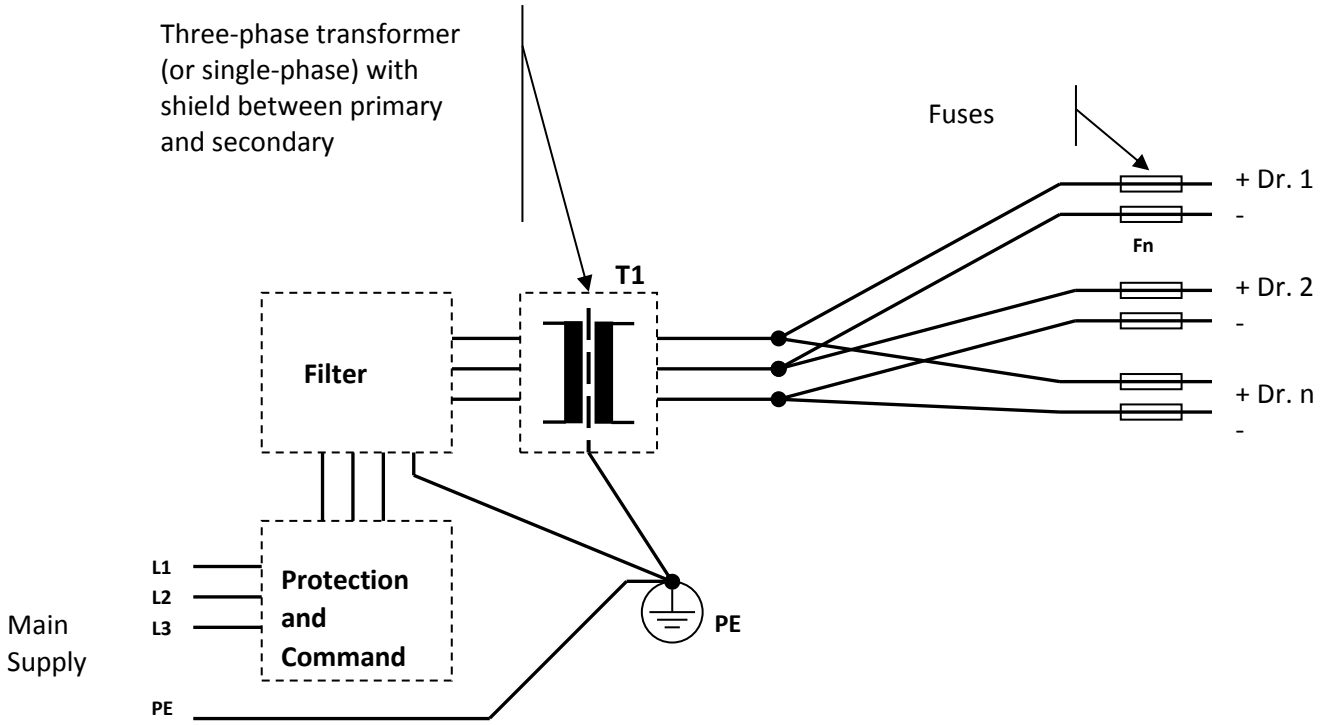
Model	Cable section (mm^2)
DDS6x41A	1
DDS6x44A	1
DDS6x48A	1.5
DDS6x76A	1
DDS6x78A	1.5

The power cable can be installed together with the ones connecting the drive to the motor. It is recommended not to place the power supply cable nearby the signal ones.

While choosing the transformer secondary voltage it is important to take into account the maximum net fluctuation expected in the worst operative conditions, the maximum vacuum voltage and the minimum full load voltage and to ensure that the maximum and minimum values, result of the combination of these components, are within the maximum and minimum voltage values specified for the chosen drive model.

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

The following is an example of base connection.



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

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

Do not connect the transformer secondary to earth otherwise there is a risk of permanent damage to the drive.

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

The reduction level the filter must guarantee can vary a lot according to the rules applied to the field to which the application and/or installation belongs.

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

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

The following table relates the suggested value for some components according to the number of drives present in the application. The calculation considers also an oscillation of the main supply voltage included within +10/-20%.

Model	Fuses Fn (A T)	Number of drives	Secondary T1 (Vac)	Power T1 (VA)	Current D1 (Arms)
DDS6x41A	2	1	32	50	25A
		2		100	25A
		3		150	25A
		4..5		250	25A
		6..8		350	25A
DDS6x44A	6.3	1	32	125	25A
		2		250	25A
		3		375	25A
		4..5		600	25A
		6..8		900	35A
DDS6x48A	12.5	1	32	250	25A
		2		500	25A
		3		750	25A
		4..5		1100	35A
		6..8		1800	50A
DDS6x74A	8	1	55	300	25A
		2		600	25A
		3		900	25A
		4..5		1400	35A
		6..8		2100	50A
DDS6x78A	16	1	55	400	25A
		2		800	25A
		3		1200	25A
		4..5		1800	35A
		6..8		2800	50A

The working voltage of the T1 transformer primary winding must be chosen according to the main 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 earth with a short and not inductive connection. The secondary winding voltage is meant without the load, with the primary winding supplied at the nominal voltage.

In the applications with more than a drive, if the drives are not all calibrated 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 made when the motors' speed is limited.

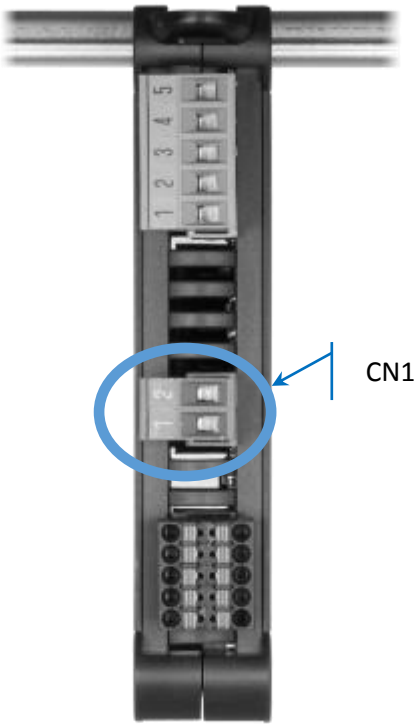
The set composed by the filter and the transformer must be used only to supply voltage to the drives. It is advised against deriving other supplies from any of these parts. On the contrary, it is suggested to get auxiliary supplies using directly the main supply upstream of the filter.

2.1.2 CN1 – Power Supply, DC models

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

Do not supply the drive before the wiring is complete.

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



CN1 – Power Supply	
Pin	Description
1	+Vp, positive DC supply voltage
2	-Vp (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
DDS6x4x	Vp	DC supply voltage	V	20		50
	Vpbrk	Voltage causing permanent damage	V	-0.5		60
DDS6x7x	Vp	DC supply voltage	V	24		90
	Vpbrk	Voltage causing permanent damage	V	-0.5		105

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
DDS6x4x	Vpl	Under voltage protection intervention threshold	V		18	
	Vph	Over voltage protection intervention threshold	V		52	
DDS6x7x	Vpl	Under voltage protection intervention threshold	V		22	
	Vph	Over voltage protection intervention threshold	V		96	

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

Model	Voltage (V)	Capacity (µF)
DDS6x41	63	470
DDS6x44	63	470
DDS6x48	63	1000
DDS6x76	100	470
DDS6x78	100	1000

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

Model	Section (mm^2)
DDS6x41	1
DDS6x44	1
DDS6x48	1.5
DDS6x76	1
DDS6x78	1.5

The power supply cable can be installed together with the ones which connect the drive to the motor. We recommend not to place 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 network's 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's tolerance during its sizing so that, in case of mains supply and load fluctuations, 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, to ensure the correct working and to fulfill each safety requirements. To determine the power supply output voltage it must be considered the maximum mains voltage fluctuation expected in the worst operative conditions, the maximum vacuum 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 from the torque required to the motor as well as from the rotation speed) and by the motor and drive efficiency.

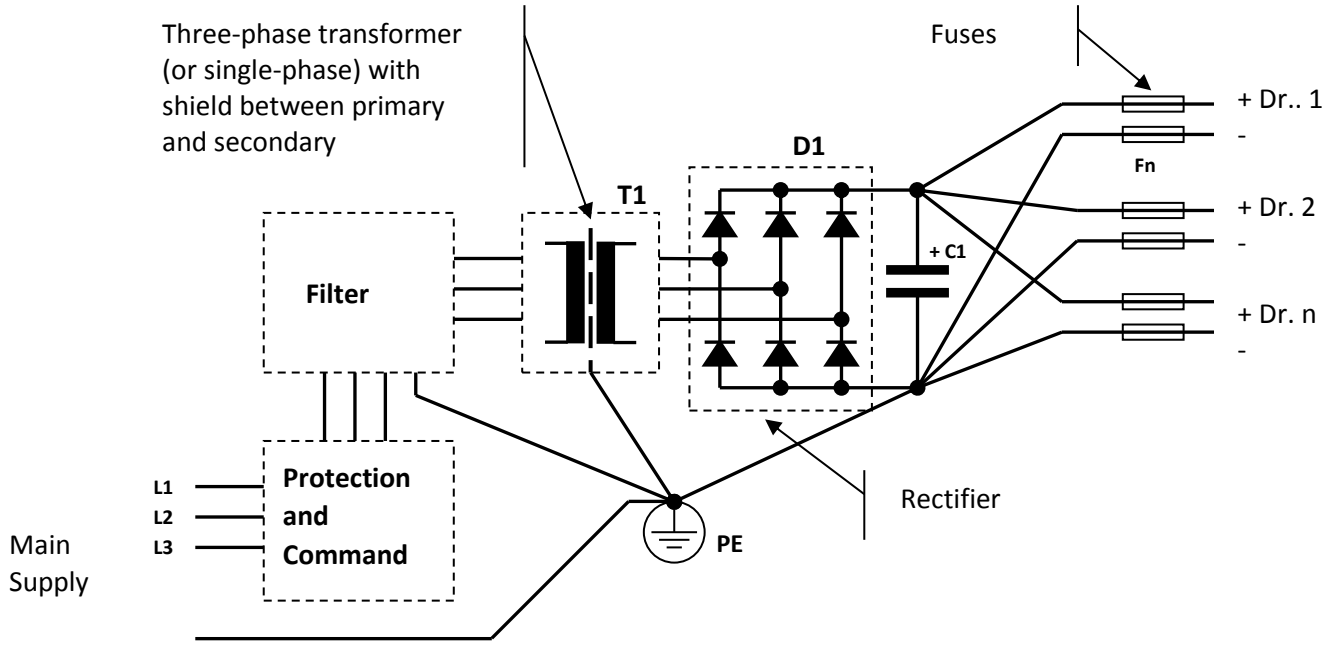
The following formula provides a rough indication:

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 1.5Ω and is power supplied with a current of 3Arms, and works at a speed of 500rpm with a load of 2Nm, 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 the power supply has output capacitors suitable to the size of the chosen drive (see further on).

To limit the peak of current at the enabling of 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 2410_h:02_h *Current_Enable_Ramp_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 metal housing of the electric system and the earth of the plant.

Also the wiring to the drives must be star-like fixing the star canter on the poles of the filter capacitor C1.

It is mandatory to provide a fuse on each phase of the transformer primary winding, able to intervene in case of short circuit or malfunctioning. It is also compulsory 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 considers also an oscillation of the main supply voltage +10/-20%.

Model	Fuses Fn (A T)	Number of drives	Secondary T1 (Vac)	Power T1 (VA)	Current D1 (Arms)	Voltage C1 (Vdc)	Capacity C1 (µF)
DDS6x41	2	1	32	50	25A	63	1000
		2		100	25A		2200
		3		150	25A		3300
		4..5		250	25A		4700
		6..8		350	25A		5600
DDS6x44	6,3	1	32	125	25A	63	3300
		2		250	25A		4700
		3		375	25A		5600
		4..5		600	25A		8200
		6..8		900	35A		10000
DDS6x48	12,5	1	32	250	25A	63	4700
		2		500	25A		6800
		3		750	25A		8200
		4..5		1100	35A		10000
		6..8		1800	50A		15000
DDS6x74	8	1	55	300	25A	100	2200
		2		600	25A		3300
		3		900	25A		3900
		4..5		1400	35A		4700
		6..8		2100	50A		6800
DDS6x78	16	1	55	400	25A	100	3300
		2		800	25A		4700
		3		1200	25A		5600
		4..5		1800	35A		8200
		6..8		2800	50A		10000

The capacity values suggested for C1 can also be obtained 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 at power supply turned off.

The working voltage of the T1 transformer primary winding must be chosen according to the main 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 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 current supplied 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 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 configurations with more than one drive, if the drives are not all calibrated 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 in a different way the components, it is also possible to realize a single-phase power supply, which is not recommended when the required power is greater than 800W.

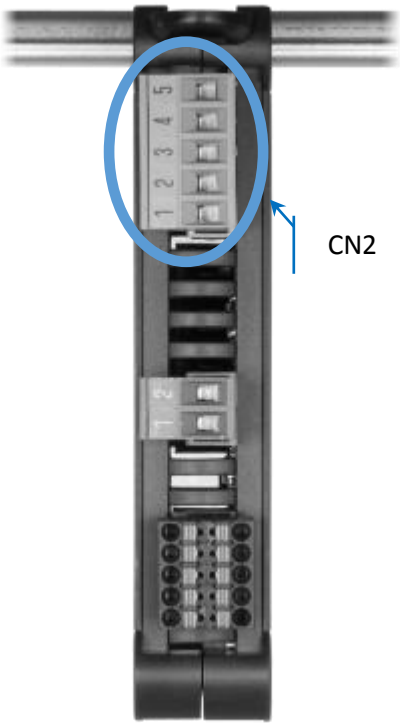
The set filter, transformer and power supply must be used only to supply voltage to the drives. It is not recommended to derive other supplies from any of these parts. Rather, it is suggested to get auxiliary supplies using directly the main supply upstream of the filter.

2.1.3 CN2 – Motor

The drive regulates the phase current of the motor through the supply voltage modulation in PWN technique. The use of a good quality shielded cable and a correct wiring are essential to better reduce the electromagnetic emission.

The cable shield must be connected to the SHIELD terminal (pin 5) of the drive but not to the body of the motor if electrically connected to the structure on which it is fixed. Differently, unwanted ground loop may occur which could damage the drive. Only in the event that the motor is insulated from the structure it is possible to connect the cable shield also to the motor side.

When the motor is electrically connected to the structure it is possible to connect the body of the motor to the machine ground node.



CN2 – Motor	
Pin	Description
1	A-, negate output phase A
2	A+, positive output phase A
3	B+, positive output phase B
4	B-, negative output phase B
5	SHIELD, (internally connected with GND)
Note: Inverting the FA+ phase with the FA-, or the FB+ with FB-, the motor rotation direction is inverted.	

The cable section can be dimensioned according to the drive current calibration, 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 10m. For cables with a greater length, the cable size must be increased to counterbalance the voltage drop.

The following table reports the cable section suggested for each drive according to the cable length:

Model	Section (mm^2)	
	Cable length < = 10m	Cable length > 10m
DDS6x41	0.5	1
DDS6x44	1	1.5
DDS6x48	1.5	2.5
DDS6x76	1	1.5
DDS6x78	1.5	2.5

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

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

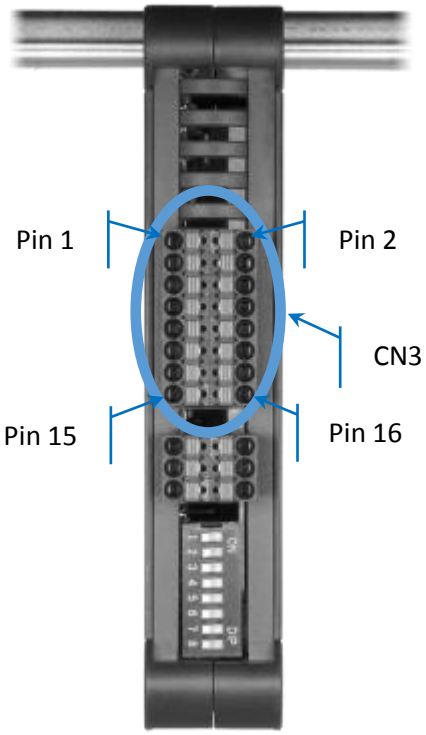
2.1.4 CN3, CN5 – I/O Control Signals

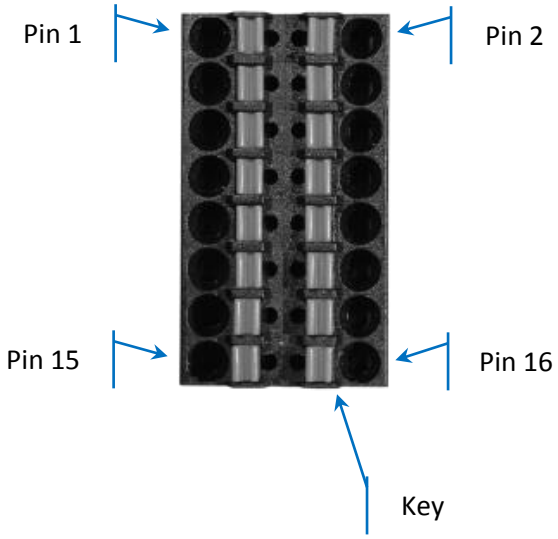
The connection with the digital control signals is through a 16ct removable spring terminal block. The terminal block can be easily oriented through the key, as shown in the picture below.

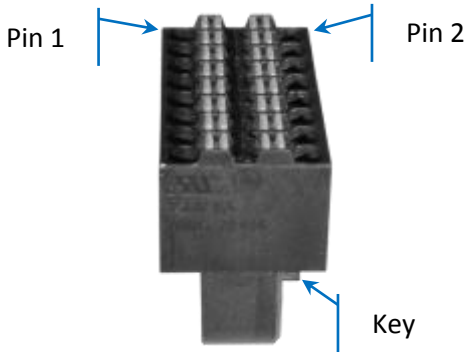
To insert the cable into the connector, press with a small screwdriver the orange presser and simultaneously insert the wire into the near hole, then release the presser. It is suggested to remove the wire covering to about 8mm.

On the CN3 connector are a total of 6 digital inputs and 3 digital outputs.

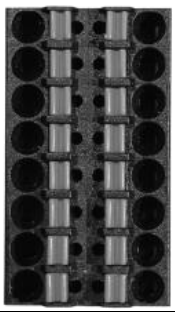
The DDS62 series has an additional connector CN5 for the connection of the analog inputs and outputs.







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

CN3 – Digital I/O Control Signals				
Description	Pin		Pin	Description
+24V Auxiliary Power Supply	1		2	0V (GND) Auxiliary Power Supply
DI567COM (common DI5..DI7)	3		4	DI234COM (common DI2..DI4)
DI5 (Digital Input 5)	5		6	DI2 (Digital Input 2)
DI6 (Digital Input 6)	7		8	DI3 (Digital Input 3)
DI7 (Digital Input 7)	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)
DO2+ (Digital Output 2)	15		16	DO2- (Digital Output 2)

Note: The numbering of the inputs starts from 2 instead of 0 in coherence with other types of DDS6 Series drives (for example DDS1), where the digital inputs DI0 and DI1 have special features.

2.1.4.1 Auxiliary Power Supply

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

Keeping supplied the logic section of the drive, the signals, the fieldbus and the encoder reading are maintained powered. The encoder reading allows to keep track of the motor position 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

2.1.4.2 Digital Inputs

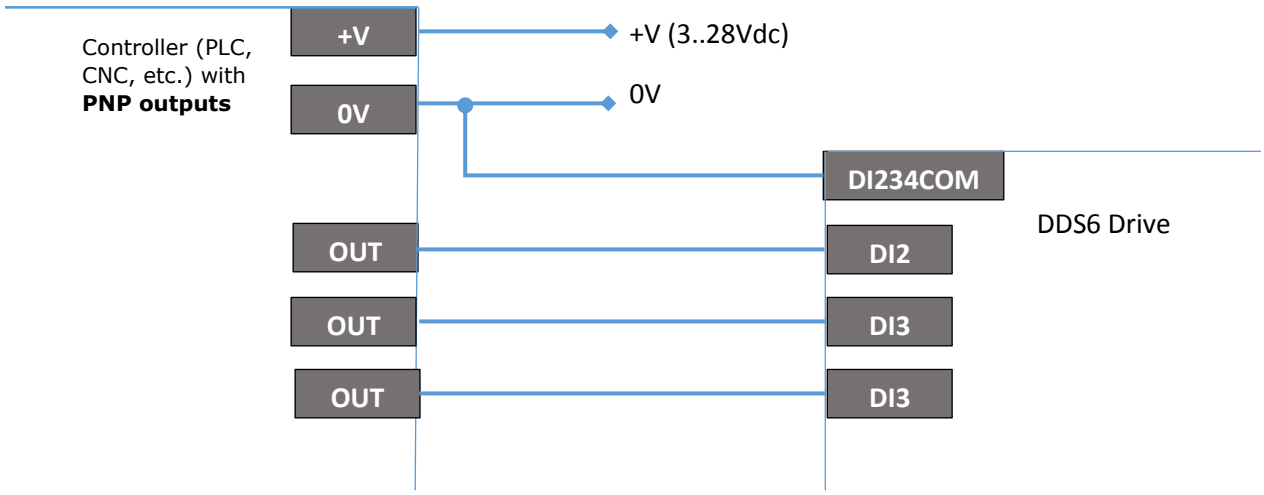
All the digital inputs are optocoupled and have a current limiting circuit which grants a constant absorption independently from the voltage applied to the input. This allows a correct functioning with a wide input voltage range without the need to introduce any external limit resistor. This simplifies installation and wiring.

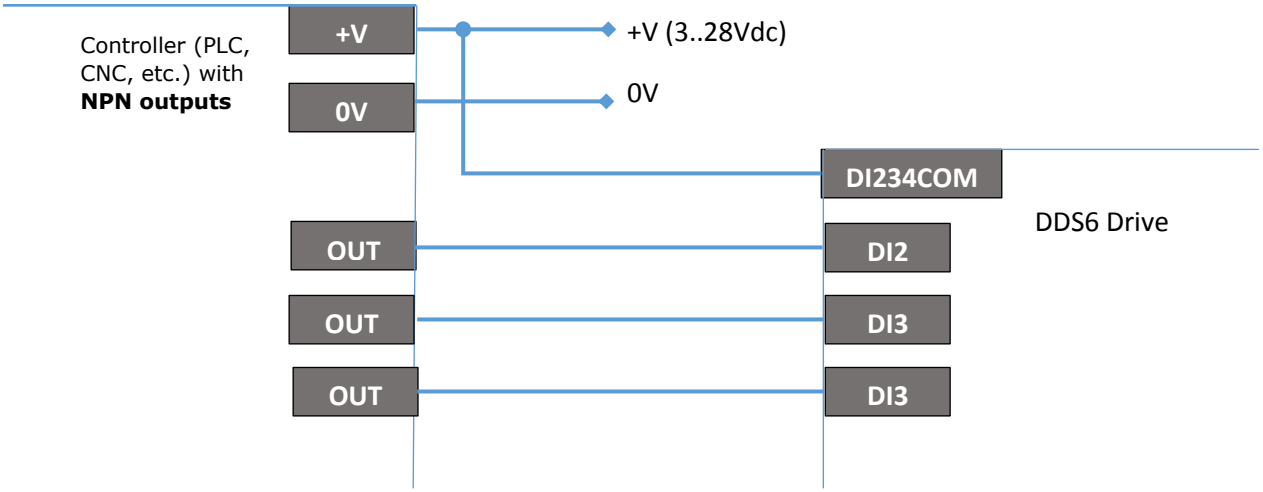
The following table shows the voltage values which correspond to the *Active* and *Inactive* input status, together with other parameters:

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

The 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 examples bellow:





2.1.4.3 Digital Outputs

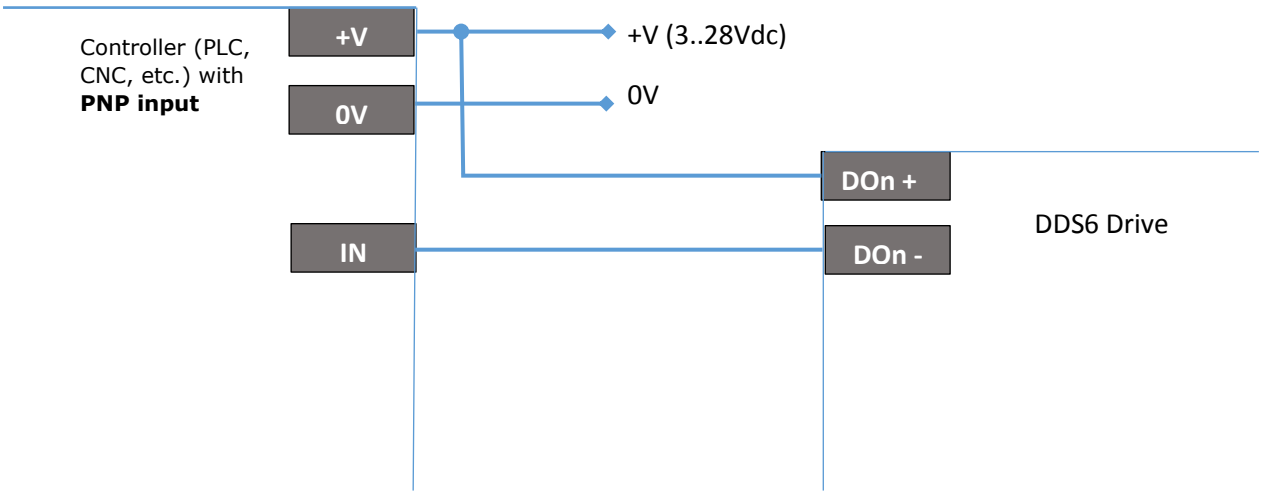
All the digital outputs are optocoupled and have both + and - connections, therefore they can be freely used in NPN or PNP configuration. On each output is placed a zener diode which allows the connection of medium entity inductive loads (for example signal relays) without the need to add an external recirculation diode.

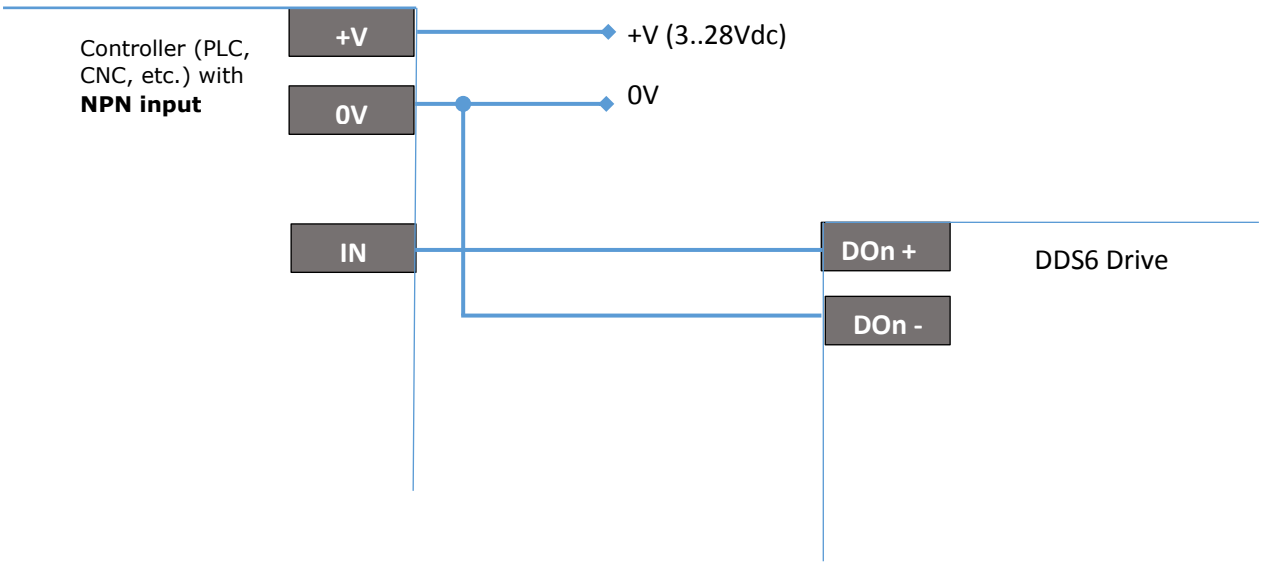
The following table shows the electrical characteristics of the digital outputs:

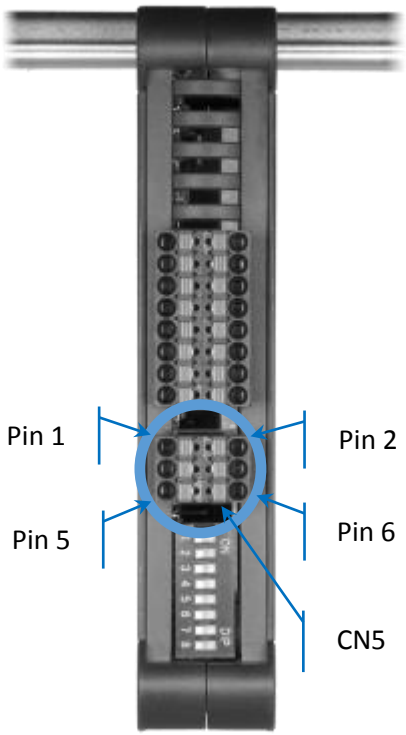
Symbol	Description	Unit	Value		
			Min	Typ	Max
Vdo	Digital output operating voltage	Vdc	1		30
Vdobrk	Digital output breakdown voltage	Vdc	-0.5		36
Vdoz	Zener diode voltage placed in parallel to each output	Vdc	36	39	42
Ido	Digital output available current	mA			80
Idobrk	Digital output breakdown current	mA	120		
Pwdo	Digital output dissipable power	mW			400

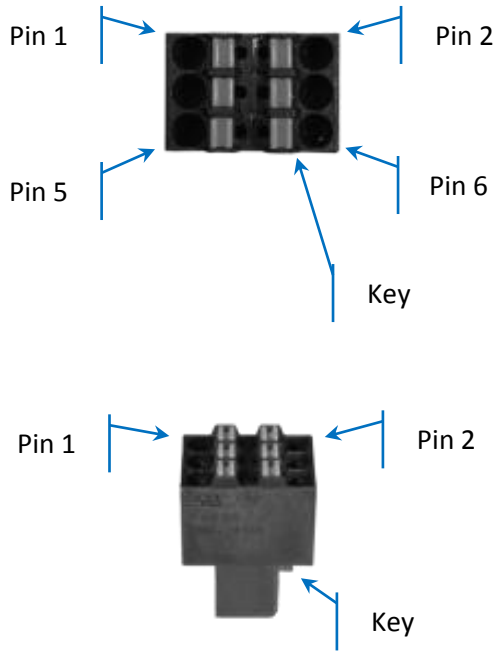
To use an output in NPN configuration simply connect the - to GND and the + to the input of the master controller (PLC, CNC, etc.), while in case of PNP connection connect the + to the positive reference and the – to the input of the control system.

The described NPN and PNP connections are shown in the example below:



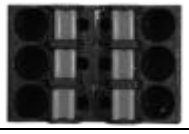






The CN5 connector is present only on the DDS62 Series and makes available 2 analog inputs and 2 analog outputs.

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

CN5 –Analog I/O control signals				
Description	Pin		Pin	Description
AO0 (Analog Output 0)	1		2	AI0 (Analog Input 0)
AO1 (Analog Output 1)	3		4	AI1 (Analog Input 1)
GND (Analog Ground)	5		6	GND (Analog Ground)

The analog inputs and outputs are not insulated and the ground reference of the analog signals is connected to the drive internally with the terminal 2 of CN1 (-Vp).

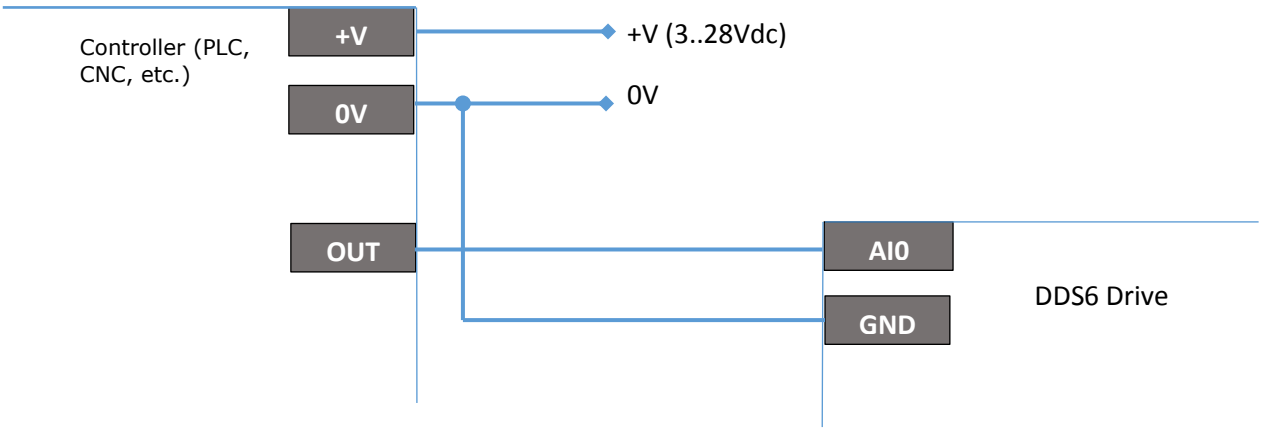
2.1.4.4 Analog Inputs

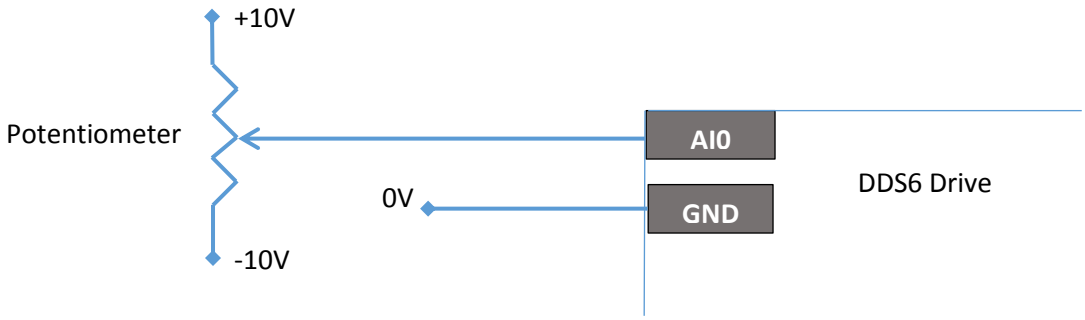
The analog input is able to measure voltages between -10V and +10V.

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

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

Example of connection of the analog input:





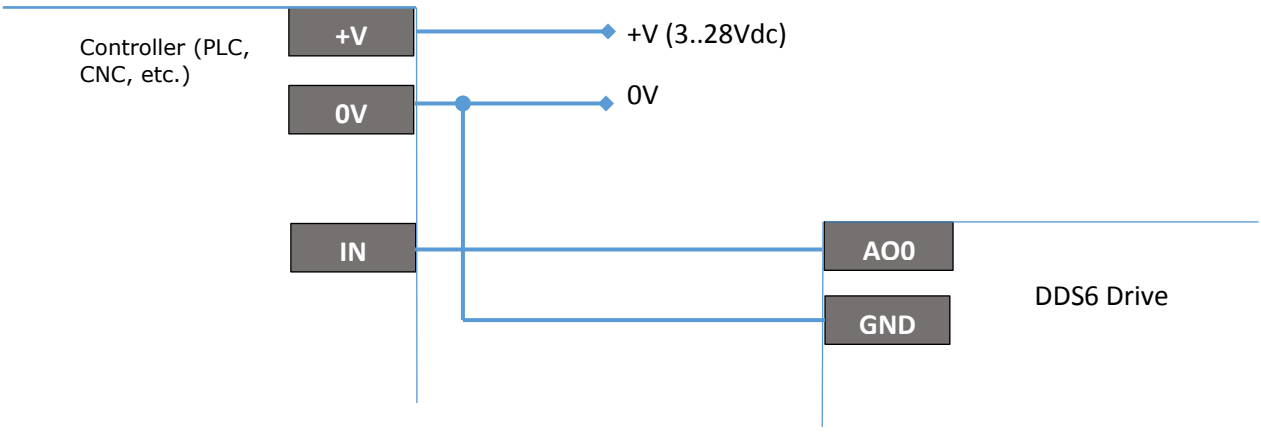
2.1.4.5 Analog outputs

The analog outputs can supply voltages between 0 and 10V.

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

Symbol	Description	Unit	Value		
			Min	Typ	Max
Vao	Analog outputs operating voltage	Vdc	0		+10.2
Iao	Analog outputs operating current	mA		10	
Rai	Analog outputs impedance	Ω			47
DAst	D/A converter conversion time	ms		1	
DAsoff	D/A converter start offset	%fs		1	
DAdoff	D/A converter offset drift	%fs		0.2	
DAline	D/A converter linearity error	%fs		1	

Example of connection of the analog output:

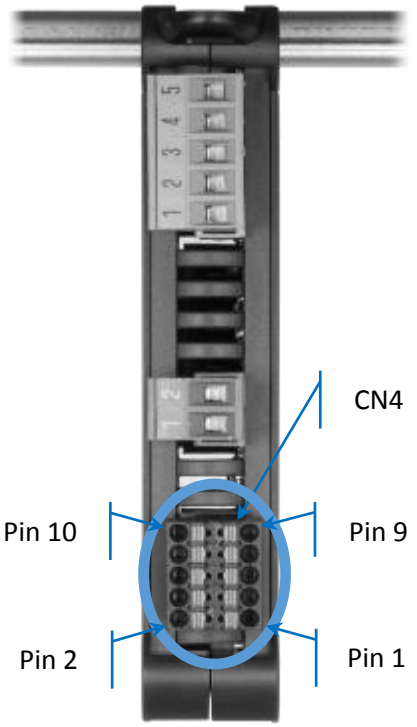


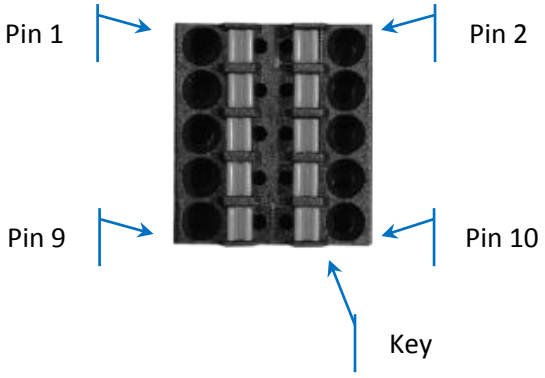
2.1.5 CN4 – Motor Encoder

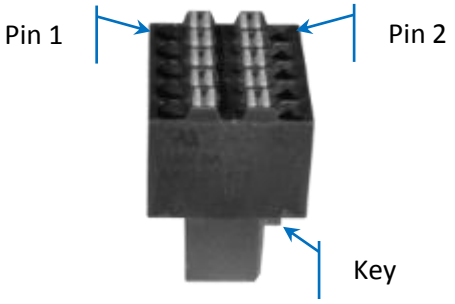
The connection with the encoder is through a 10ct removable spring terminal block. The terminal block can be easily oriented through the key, as shown in the picture below.

The CN4 connector is present only on the DDS62 Series drives.

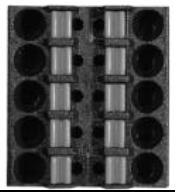
To insert the cable into the connector, press with a small screwdriver the orange presser and simultaneously insert the wire into the near hole, then release the presser. It is suggested to remove the wire covering to about 8mm.







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

CN4 – Motor Encoder				
Description	Pin		Pin	Description
+V Encoder Power Supply	1		2	0V (GND) Encoder Power Supply
A+ (Encoder Phase A)	3		4	A- (Negate Encoder Phase A)
B+ (Encoder Phase B)	5		6	B- (Negate Encoder Phase B)
I+ (Encoder Index)	7		8	I- (Negate Encoder Index)
Not used (leave disconnected)	9		10	Not used (leave disconnected)

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 0V reference.

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 values (object 2330_h:02_h *CPR_ENCMTR*).

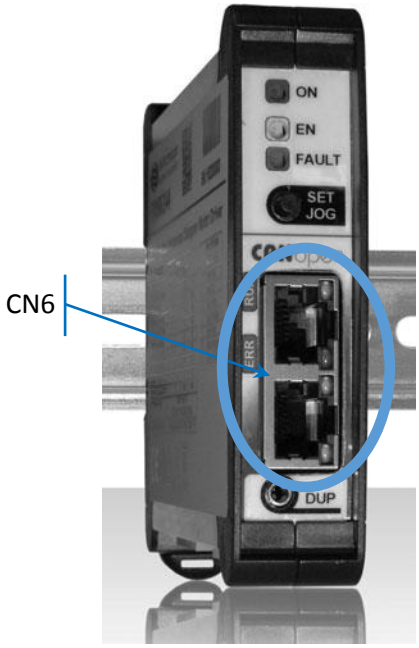
To power the Encoder, the drive supplies a voltage of +5V with a current of 100mA suitable for the most encoders, however it is also possible to connect encoders with a different supply voltage, provided that they are externally supplied.

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

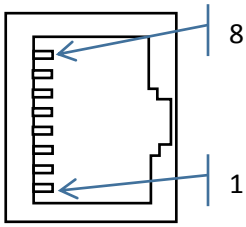
Encoder output signals types	Encoder Signal	Drive Signal	Notes
Line Driver	A+	A+	
	A-	A-	
	B+	B+	
	B-	B-	
	I+	I+	
	I-	I-	
TTL/CMOS	A	A+	The inputs A-, B- and I- remains disconnected.
	B	B+	
	I	I+	
Open Collector	A	A+	The inputs A-, B- and I- remains disconnected.
	B	B+	
	I	I+	
Push-Pull	A	A+	**ATTENTION, when using an encoder supplied with voltage higher than 5V 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+	

2.1.6 CN6 – CANopen Bus

The CN6 connector has two sockets for standard 8pins RJ45 connector. All the signals of each socket are connected together.



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

CN5 – CANopen Bus		
Description	Pin	
Non used	8	
Non used	7	
Non used	6	
Non used	5	
Non used	4	
CAN_GND	3	
CAN_L	2	
CAN_H	1	Socket front view

For the connection it is possible to use a common and cheap Ethernet cable CAT5or superior class.

Please note that, according to the CAN specifications, the bus must be ended with two resistors; each one of them placed at the two ends of the network. The resistors must have a value of 120ohm and must be connected between the CAN_L and CAN_H signals.

3 Configuration

The DDS6 Series drives are configured through the CANopen fieldbus. During the initialization phase, the master controller writes in the dictionary objects the configuration values required by the application.

A dip-switch allows to select the communication bit-rate and the node address, in order to make the drive accessible by the master controller.

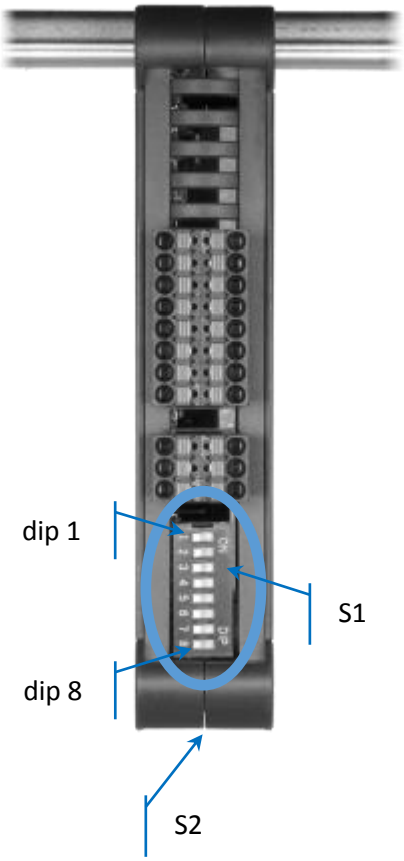
ATTENTION, incorrect settings or unsuitable to the application can cause unexpected movements of the motor, unwanted activation of signals, monitoring functions disabling, etc. Some settings become active at the following restart of the device.

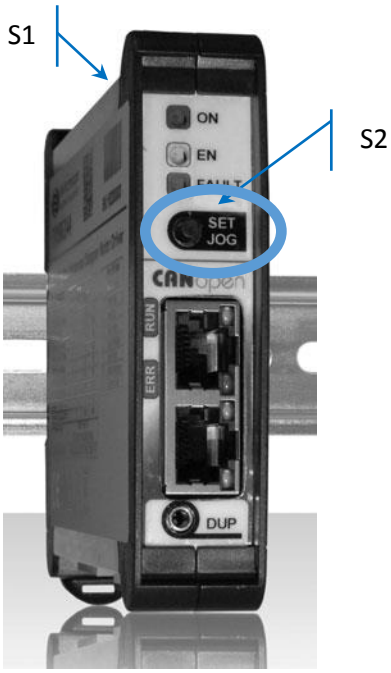
ATTENTION, do not use the device if you do not know or have not understood the settings. After each setting change, test accurately the application in any possible condition of use or error so not to cause damages to people, animals or things or economic loss.

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

3.1 Configuration Dip-switch

The dip-switch configuration takes place in two phases; first it is necessary to select the parameter to configure and set its value using the dip-switch, then press the S2 button (on the front of the drive) for 1s to store the parameter in the non-volatile memory of the device. The drive confirms the storage by flashing three times the green LED ON.





The following table shows the dip-switches involved in the selection of the parameter and value setting:

Parameter	Dip-switch							
	1	2	3	4	5	6	7	8
None	Off	Off	Off	Off	Off	Off	Off	Off
Node address setting	Off							

Dip-switch							Node-ID
2	3	4	5	6	7	8	
Off	Off	Off	Off	Off	Off	On	1
Off	Off	Off	Off	Off	On	Off	2
Off	Off	Off	Off	Off	On	On	3
…	…	…	…	…	…	…	4…124
On	On	On	On	On	Off	On	125
On	On	On	On	On	On	Off	126
On	On	On	On	On	On	On	127

<table>
<tr><td>Bit-rate setting</td><td>On</td><td>Off</td><td>Off</td><td>Off</td><td>
<table>
<tr><th colspan="4">Dip-switch</th><th rowspan="2">Bit-rate</th></tr>
<tr><th>5</th><th>6</th><th>7</th><th>8</th></tr>
<tr><td>Off</td><td>Off</td><td>Off</td><td>Off</td><td>10 Kbit/s</td></tr>
<tr><td>Off</td><td>Off</td><td>Off</td><td>On</td><td>20 Kbit/s</td></tr>
<tr><td>Off</td><td>Off</td><td>On</td><td>Off</td><td>50 Kbit/s</td></tr>
<tr><td>Off</td><td>Off</td><td>On</td><td>On</td><td>100 Kbit/s</td></tr>
<tr><td>Off</td><td>On</td><td>Off</td><td>Off</td><td>125 Kbit/s</td></tr>
<tr><td>Off</td><td>On</td><td>Off</td><td>On</td><td>250 Kbit/s</td></tr>
<tr><td>Off</td><td>On</td><td>On</td><td>Off</td><td>500 Kbit/s</td></tr>
<tr><td>Off</td><td>On</td><td>On</td><td>On</td><td>1 Mbit/s</td></tr>
</table>
</td></tr>
</table>

To store the new configuration the motor must be disabled. On the contrary, if the drive is in the *Operation enabled* or *Quick stop active* status, the S2 button will have no effect as in these status the motor is enabled.

The new configuration becomes active at the following restart of the device or after a NMT Reset Node.

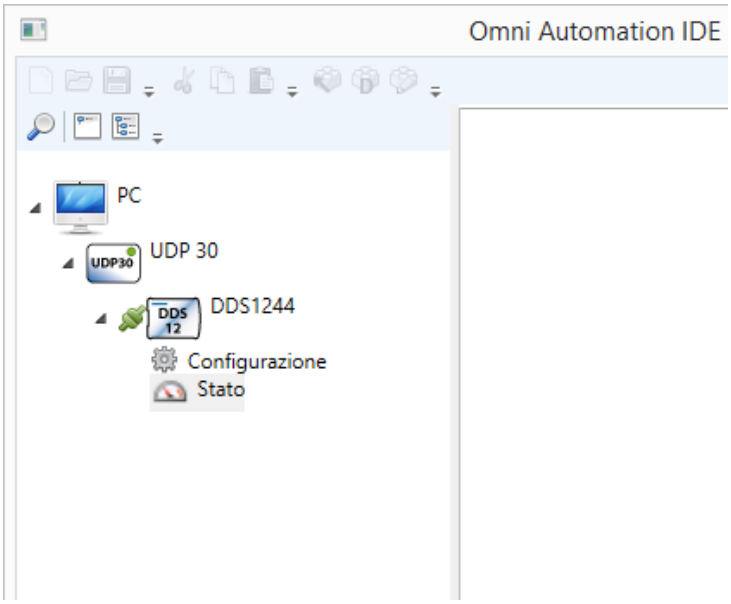
3.2 Configuration software

The free software *Omni Automation IDE* (hereinafter OAI) running under Windows platform (Windows 7, Windows 8.1 and Windows 10 32bit or 64bit) allows to configure the drive through a useful interface assisted by help tooltip and also support the device diagnostics.

The connection between PC and drive is through the UDP30 interface which also galvanically isolate the PC from the device. The UDP30 interface is also able to supply the logic section allowing its configuration even without power supply.

After having connected the UDP30 interface to the DUP port, on the front of the drive, it is possible to press the button *Search* so that OAI starts scanning UDP30 interfaces and connected devices.

At the end of the research it appears the tree of the devices connected to the PC, similar to the image below:



Double-clicking on the drive name (DDS1244 on the image) it appears the tab with the characteristic data of the device such as the serial number, the firmware revision, etc. Through the link *Update* it is also possible to update the device firmware

Double-clicking on *Configuration* (visible under the drive name) it opens the tab which allows to modify the device configuration.

If a value in a field of the *Configuration* differs from the one present in the drive, it appears a yellow frame around the field to highlight the difference. Resting the mouse on the field appears a tooltip that shows the value present in the device.

A red frame around a field indicates a compiling error, such as a value out of range, the use of illegal characters, etc. Resting the mouse on the field a tooltip appears showing the error details.

On top you can see the *Name* field which allows to assign a name or a brief description to the configuration.

Then there is the *Device* section that also contains the *Name* field, this time referred to the drive. The string entered here is stored in the drive and it is useful to easily identify the device. For example, the drive adopted for raising the spindle could be called *Z Axis*.

Note that when you change the configuration you must press the *Write* button to store the same configuration in the device.

Follows a description of the remaining sections of the Configuration:

3.2.1 Node address (Node ID)

It sets the address of the CANopen node.

As provided by the standard, you can select an address from 1 to 127.

3.2.2 Communication bit rate

It configures the communication speed (bit rate) of the node.

The selected bit rate must be identical to the one on the network to which the device is connected.

The supported speed are listed in the table below:

10 Kbit/s
20 Kbit/s
50 Kbit/s
100 Kbit/s
125 Kbit/s
250 Kbit/s
500 Kbit/s
1 Mbit/s

The new configuration becomes active at the next restart of the device or after a NMT Reset Node.

4 Operating

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

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

The drive implements the profile /CiA402/ dedicated to the drives and to the motion control devices. The operating modes supported by the firmware revision described in the this manual are 5, as shown in the table below:

Abbreviation	Description
pp	Profile position mode
pv	Profile velocity mode
tq	Torque profile mode
hm	Homing mode
ip	Interpolated Position

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

Before operating, the drive requires some preliminary settings by writing appropriate value in the objects that compose the dictionary of the device itself. Many objects are provided by the profile /CiA402/ and contained in the *Standardized profile area*, while others are a DDS6 Series' peculiarity and are contained in the *Manufacturer-specific profile area*.

4.1 Minimum settings

The minimum settings to be made before enabling the motor are the parameter setting of the motor connected to the drive (object 2310_h *Motor Data*) and the setting of the running and idle current (object 3310_h *Motor Configuration*).

4.1.1 Motor parameters setting

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

All characteristic parameters are contained inside the object 2310_h *Motor Data* Record type. It follows a detailed description of the object contained within the record.

4.1.1.1 CMC_MTRDT (2310h:01h)

If you are using a LAM Technologies motor just compile the sub-index 01_h *CMC_MTRDT* with the CMC motor code and automatically the drive will use the optimal configuration for the chosen motor.

The following table shows the correspondence between the CMC code and the motors. For the motors which allow *Bipolar Parallel* or *Bipolar Series phases* connections are given different CMC codes, as they vary in the electrical features and dynamics.

LAM Technologies motors CMC code

CMC	Motor	Type of connection
	NEMA 17	
130200	M1173020	Unchangeable
130210	M1173021	Unchangeable
130300	M1173030	Unchangeable
130310	M1173031	Unchangeable
130400	M1173040	Unchangeable
130410	M1173041	Unchangeable
	NEMA 23	
230110	M1233011	Unchangeable
230120	M1233012	Unchangeable
230210	M1233021	Unchangeable
230220	M1233022	Unchangeable
230310	M1233031	Unchangeable
230320	M1233032	Unchangeable
230410	M1233041	Unchangeable
230510	M1233051	Unchangeable
230610	M1233061	Unchangeable
230620	M1233062	Unchangeable
230640	M1233064	Unchangeable
230700	M1233070	Unchangeable
230710	M1233071	Unchangeable
	NEMA 24	
530410	M1243041	Unchangeable
530420	M1243042	Unchangeable
530440	M1243044	Unchangeable
	NEMA 34	
330110	M1343011	Bipolar Parallel
330111	M1343011	Bipolar Series
330200	M1343020	Bipolar Parallel
330201	M1343020	Bipolar Series
330210	M1343021	Bipolar Parallel
330211	M1343021	Bipolar Series
330310	M1343031	Bipolar Parallel
330311	M1343031	Bipolar Series
330410	M1343041	Bipolar Parallel
330411	M1343041	Bipolar Series
330500	M1343050	Bipolar Parallel
330501	M1343050	Bipolar Series
330510	M1343051	Bipolar Parallel
330511	M1343051	Bipolar Series
330600	M1343060	Bipolar Parallel
330601	M1343060	Bipolar Series

	NEMA 42	
430100	M1433010	Bipolar Parallel
430101	M1433010	Bipolar Series
430200	M1433020	Bipolar Parallel
430201	M1433020	Bipolar Series
430400	M1433040	Bipolar Parallel
430401	M1433040	Bipolar Series
	NEMA23 with Encoder	
230410101	M1241E106	Unchangeable
230620101	M1262E106	Unchangeable
230640101	M1264E106	Unchangeable
230700101	M1270E106	Unchangeable
230710101	M1271E106	Unchangeable
	NEMA34 with Encoder	
330200101	M1320E106	Unchangeable
330201101	M1325E106	Unchangeable
330310101	M1331E106	Unchangeable
330311101	M1336E106	Unchangeable
330500101	M1350E106	Unchangeable
330501101	M1355E106	Unchangeable
330600101	M1360E106	Unchangeable
330601101	M1365E106	Unchangeable

If the motor is not in the table it may have been introduced recently. Usually the CMC code is shown in the datasheet of the motor and on the dedicated page on the website. If you do not find it you can ask for it writing to support@lamtechnologies.com.

4.1.1.2 Pole_Pairs_MTRDT (2310h:03h)

The object *Pole_Pairs_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 motor.

In case of a two-phases stepper motor each pole gives rise to 4 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).

4.1.1.3 Resistance_MTRDT (2310h:05h)

The object *Resistance_MTRDT* must be compiled with the correct value of the motor phase resistance. Each unit is worth 10mOhm (i.e. 0.01Ohm) then, for example, to set a value of 3.5Ohm 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 technical datasheet.

Some motors allow more types of phases connection and in this case it is necessary to verify for which connection is specified the resistance value 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 2.2Ohm in unipolar and is connected to the drive with the phases set in bipolar parallel, the object *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 object *Resistance_MTRDT* will be 440 (2.2 * 2 / 0.01).

If the two-phase motor has 4 wires, it means that the type of phase connection has been already decided during production and the resistance value specified by the manufacturer is therefore the one to be written in the object *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.

4.1.1.4 Inductance_MTRDT (2310h:06h)

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

Some motors allow more types of phase connection and in this case it is necessary to verify for which connection is specified the inductance value 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 1.6mH in unipolar and is connected to the drive with the phases in bipolar parallel, you will have to compile the object *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 4 wires it means that the type of phase connection has been already decided during production and the inductance value specified by the manufacturer is therefore the one to be used for the object *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.

4.1.1.5 Back_EMF_MTRDT (2310h:07h)

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

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

Some motors allow more types of phase connection and in this case it is necessary to verify for which connection is specified the value of counter-electromotive force 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 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 28V at 1000rpm in unipolar and is connected to the drive with the phases in bipolar parallel, you will have to compile the object *Back_EMF_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 4 wires it means that the type of phase connection has been already decided during production and the value of the counter-electromotive force generated by the motor is therefore the one to be used for the object *Back_EMF_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 50Hz. Successively you need to compare the voltage value measured at a speed of 1000rpm 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 in the end 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 60rpm (to obtain 50Hz) and the measured voltage value was of 4.85Vac, the Vbemf value will be equal to 80.83 (4.85 * 1000 / 60) and the value to be inserted in the object *Back_EMF_MTRDT* will be (80.83 / 0.01).

Note that the frequency of the BEMF is connected 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 make rotate a motor of 50 poles (corresponding to a step angle of 1.8°) at 100rpm we obtain a BEMF frequency equal to about 83.3Hz.

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

4.1.1.6 Rated_Current_MTRDT (2310h:08h)

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

The value written in the object *Rated_Current_MTRDT* must take into account the phases connection chosen for the motors that provide more possibilities. According to the connection chosen for the phases and to the configuration chosen by the manufacturer to characterize the 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 2A in unipolar connection and you choose to connect the motor to the drive in bipolar parallel, it is necessary to set the object *Rated_Current_MTRDT* to the value 280 (2 * 1.41 / 0.01).

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

If a two-phase motor has 4 wires it means that the type of phase connection has been already decided 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 object *Rated_Current_MTRDT* corresponds exactly to the rated current of the motor and it is never used this parameter to set the working current.</u> The working current of the motor is set through the object 3310_h:03_h *Current_Max_MTRCNF* described later.

4.1.1.7 Max_Current_MTRDT (2310h:09h)

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

In the event that the data is not available, it is suggested to use the same value of the object 2310_h:08_h *Rated_Current_MTRDT*.

The value written in the object *Max_Current_MTRDT* must consider the connection chosen for the phases for the motors which provide more possibilities. See previous chapter (*Rated_Current_MTRDT*).

4.1.1.8 Rated_Torque_MTRDT (2310h:0Ah)

The object *Rated_Torque_MTRDT* must be compiled with the static torque value of the motor when supplied at the rated current. Each unit is worth 10mNm (i.e. 0.01Nm) then to set, for example, a value of 6.8Nm 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 is specified the rated torque 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 3.1Nm in unipolar and is connected to the drive with the phases in bipolar parallel or in bipolar series, you will have to compile the object *Rated_Torque_MTRDT* with the value 437 (3.1 * 1.41 / 0.01).

If a two-phase motor has 4 wires it means that the type of phase connection has been already decided 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 <u>that the value in the object *Rated_Torque_MTRDT* corresponds exactly to the rated torque of the motor.</u>

4.1.1.9 Max_Speed_MTRDT (2310h:0Bh)

The object *Max_Speed_MTRDT* must be compiled with the maximum speed reached by the motor in the application. Each unit is worth 0.1rpm therefore to set, for example, a value of 600rpm it is necessary to write the value 6000 (600 / 0.1) in the object.

4.1.2 Running and idle current configuration

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

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

When the motor is provided with Encoder and the drive configured for closed-loop control, you can set the current regulation so that it adapts to the load applied to the motor (object $23A0_h$:01_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).

The parameters related to the operating current of the motor are within the object 3310_h *Motor Configuration*. The object is Record type and is provided with sub-indices. Current_Min_MTRCNF (3310h:02h)

4.1.2.1 Current_Min_MTRCNF (3310h:02h)

The object *Current_Min_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 object *Current_Idle_Delay_MTRCNF* described later.

The object is expressed as a percentage of the motor rated current (object 2310_h:08_h *Rated_Current_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 object *Current_Min_MTRCNF* with the value 3000 (30 / 0.01). If the configured motor rated current is for example of 4Arms, the idle current will be equal to 1.2Arms (30% of 4A).

4.1.2.2 Current_Max_MTRCNF (3310h:03h)

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

The object is expressed as a percentage of the motor rated current (object 2310_h:08_h *Rated_Current_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 object *Current_Max_MTRCNF* with the value 8000 (80 / 0,01). If the configured motor rated current is for example of 4Arms, the idle current will be equal to 3.2Arms (80% of 4A).

4.1.2.3 Current_Idle_Delay_MTRCNF (3310h:04h)

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

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

4.2 Saving and restoring of default values

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

Name	Mnemonic
PDO Mapping	Note
Maximum	Unit
	Descrizione

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

By saving a value different from the default one it is possible to adapt the device to the application without the need to configure it each time. Apparently this seems to be a simplification but it forces to prepare the device (saving the wished data in the dictionary objects) 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 objects 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. Un this case, in fact, specific application will initialize the objects with the wished values and without the possibility of error. Furthermore, if the device should need to be replaced, the technical support can simply send a new device without worrying about the application and the version in which it will be installed.

When possible it is therefore recommended not to use the *Save* function to modify the default value of the objects. On the contrary, it is recommended to always initialize every object used in the application with the wished value, independently from the saving or default. The initialization must be repeated in case of *NMT Service Reset Node*.

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

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

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

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

The defaults values can be saved or restored only with the motor disabled or in the NMT status *Stopped* or *Pre-operational*. Trying the operation with the motor enabled or in the NMT *Operational* status an error code answer is received.

At most it is possible to save or restore the default values for 10,000 times.

4.3 Motor Holding Brakes

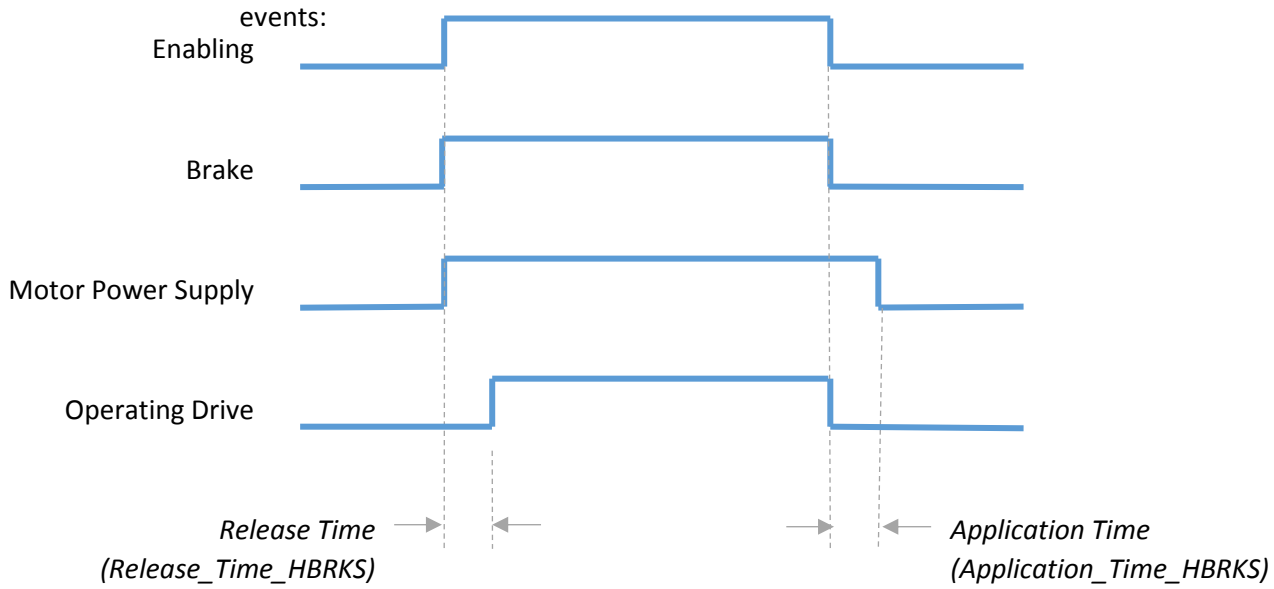
The drive is able to control the holding brake of the motor through one of the digital outputs.

Through the object *Option_HBRKS* it is possible to set the drive to control the handle of 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 immediately the digital output 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 the brake requires to completely disengage. This time can be freely set through the object *Release_Time_HBRKS*.

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

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



The object *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, and shared.

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

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

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

4.4 Operating Modes

The drive implements many of the operating modes provided by the profile /CiA402/ to meet the most different applications.

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

Abbreviation	Description
pp	Profile position mode
pv	Profile velocity mode
tq	Torque profile mode
hm	Homing mode
ip	Interpolated Position

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

The following paragraphs describe the different operating modes with complete examples of communication. All the examples assume that the drive is started with the default values and successively configured with the minimum setting described in the previous chapters (motor parameters and running and idle current setting).

4.4.1 Profile position (pp)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

The described operations can be carried out through the communication objects SDO, PDO or a combination of the two. In the following example you use the only SDO protocol to perform a cycle of 5 positionings. At the end of each of the first 2 positionings you must wait for a second, then execute the other 3 positionings in sequence using the buffer and the handshake mechanism between *Controlword* and *Statusword*.

The drive used in the following example has the address $0D_h$ and assumes to have been started with the default values and successively configured with the minimum settings described in the previous chapters (Motor parameters and Running and idle current configuration). Furthermore, NMT is considered in the *Pre-Operational* status (default status after the power on).

The values in the Time column refer to a bit rate of 250Kbit/s and can vary according to the traffic on the bus and to the reaction time of the master controller used, as well as to the firmware revision installed in the drive. The symbol → indicates a data flow from the bus to the drive while the symbol ← indicates a data flow from the drive to the CANopen bus. The communications highlighted in pale blue are those required to complete the first positioning, the subsequent lines, instead, show the evolution of the entire cycle.

The motion cycle shown in the example provides the rotation of the motor in two clockwise revolutions followed by a 1 second's break, a rotation of 6 counterclockwise revolutions followed by a 1 second's break and at the end a sequence, with no wait, of one forward revolution followed by two forward revolutions following again by one forward revolution. At the end of the cycle the motor will be in the same starting position.

→ ←	Time (ms)	COB-ID, Data	Description
→	0.0	60D, 23 81 60 00 C4 09 00 00	*Profile_velocity* object set with 2500 (250rpm)
←	1.6	58D, 60 81 60 00 00 00 00 00	
→	6.4	60D, 23 83 60 00 E8 03 00 00	*Profile_acceleration* object set with 1000 (1000rpm/s)
←	8.1	58D, 60 83 60 00 00 00 00 00	
→	12.9	60D, 23 84 60 00 D0 07 00 00	*Profile_deceleration* object set with 2000 (2000rpm/s)
←	14.6	58D, 60 84 60 00 00 00 00 00	
→	19.1	60D, 2F 60 60 00 01 00 00 00	*Modes_of_operation* object set with 1 (1 = *Profile position*)
←	20.6	58D, 60 60 60 00 00 00 00 00	
→	25.5	60D, 40 61 60 00 00 00 00 00	Reading *Modes_of_operation_display* object to check operating mode 1 active
←	27.1	58D, 4F 61 60 00 01 00 00 00	
→	31.9	60D, 2B 40 60 00 06 00 00 00	*Controlword* object set with 0006_h (PDS Shutdown)
←	33.6	58D, 60 40 60 00 00 00 00 00	
→	38.3	60D, 2B 40 60 00 0F 00 00 00	*Controlword* object set with $000F_h$ (*Switch on + Enable Operation*)
←	40.1	58D, 60 40 60 00 00 00 00 00	
→	44.7	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *Operation enabled* status (0637_h)
←	46.6	58D, 4B 41 60 00 33 26 00 00	
→	110.1	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *Operation enabled* status(0637_h)
←	111.6	58D, 4B 41 60 00 37 06 00 00	
→	115.1	60D, 23 7A 60 00 20 4E 00 00	*Target_position* object set with 20000 (2 complete clockwise revolutions)
←	116.6	58D, 60 7A 60 00 00 00 00 00	
→	119.7	60D, 2B 40 60 00 5F 00 00 00	Relative positioning and *new set-point* bit set in the *Controlword* object (value $005F_h$)
←	121.6	58D, 60 40 60 00 00 00 00 00	
→	124.5	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *set-point acknowledge* bit (value 1237_h).
←	126.1	58D, 4B 41 60 00 37 12 00 00	
→	128.5	60D, 2B 40 60 00 4F 00 00 00	Reset *new set point* bit in the *Controlword* object (value $004F_h$)
←	130.1	58D, 60 40 60 00 00 00 00 00	
→	139.0	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *set-point acknowledge* bit reset (value 0237_h)
←	140.6	58D, 4B 41 60 00 37 02 00 00	
→	…	…	Reading *Statusword* object waiting for *Target-reached* bit (value 0637_h)
←			
→	735.9	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *Target-reached* bit (value 0637_h)
←	737.6	58D, 4B 41 60 00 37 02 00 00	
→	796.3	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *Target-reached* bit (value 0637_h)
←	798.1	58D, 4B 41 60 00 37 06 00 00	
→	1810.7	60D, 23 7A 60 00 A0 15 FF FF	*Target_position* object set at -60000 (6 complete counterclockwise revolutions)
←	1812.6	58D, 60 7A 60 00 00 00 00 00	
→	1819.2	60D, 2B 40 60 00 5F 00 00 00	Relative positioning and *new set point* bit set in the *Controlword* object (value $005F_h$)
←	1821.1	58D, 60 40 60 00 00 00 00 00	
→	1826.1	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *set-point acknowledge* bit (value 1237_h)
←	1827.6	58D, 4B 41 60 00 37 12 00 00	
→	1832.5	60D, 2B 40 60 00 4F 00 00 00	Reset *new set point* bit in the *Controlword* object

←	1834.1	58D, 60 40 60 00 00 00 00 00	(value 004F$_h$)
→ ←	1839.0 1840.6	60D, 40 41 60 00 00 00 00 00 58D, 4B 41 60 00 37 02 00 00	Reading *Statusword* object waiting for *set-point acknowledge* bit reset (value 0237$_h$)
→ ←	…	…	Reading *Statusword* object waiting for *Target-reached* bit (value 0637$_h$)
→ ←	3407.0 3408.5	60D, 40 41 60 00 00 00 00 00 58D, 4B 41 60 00 37 02 00 00	Reading *Statusword* object waiting for *Target-reached* bit (value 0637$_h$)
→ ←	3468.1 3469.6	60D, 40 41 60 00 00 00 00 00 58D, 4B 41 60 00 37 06 00 00	Reading *Statusword* object waiting for *Target-reached* bit (value 0637$_h$)
→ ←	4478.9 4480.6	60D, 23 7A 60 00 10 27 00 00 58D, 60 7A 60 00 00 00 00 00	*Target_position* object set at 10000 (1 complete clockwise revolution)
→ ←	4486.3 4488.1	60D, 2B 40 60 00 5F 00 00 00 58D, 60 40 60 00 00 00 00 00	Relative positioning and *new set point* bit set in the *Controlword* object (value 005F$_h$)
→ ←	4492.6 4494.1	60D, 40 41 60 00 00 00 00 00 58D, 4B 41 60 00 37 12 00 00	Reading *Statusword* object waiting for *set-point acknowledge* bit (value 1237$_h$)
→ ←	4499.0 4500.6	60D, 2B 40 60 00 4F 00 00 00 58D, 60 40 60 00 00 00 00 00	Reset *new set point* bit in the *Controlword* object (value 004F$_h$)
→ ←	4505.6 4507.1	60D, 40 41 60 00 00 00 00 00 58D, 4B 41 60 00 37 02 00 00	Reading *Statusword* object waiting for *set-point acknowledge* bit reset (value 0237$_h$)
→ ←	4512.1 4513.6	60D, 23 7A 60 00 20 4E 00 00 58D, 60 7A 60 00 00 00 00 00	*Target_position* object set at 20000 (2 complete clockwise revolutions)
→ ←	4518.5 4520.1	60D, 2B 40 60 00 5F 00 00 00 58D, 60 40 60 00 00 00 00 00	Relative positioning and *new set point* bit set in the *Controlword* object (value 005F$_h$)
→ ←	4525.0 4526.6	60D, 40 41 60 00 00 00 00 00 58D, 4B 41 60 00 37 12 00 00	Reading *Statusword* object waiting for *set-point acknowledge* bit (value 1237$_h$)
→ ←	4531.5 4533.1	60D, 2B 40 60 00 4F 00 00 00 58D, 60 40 60 00 00 00 00 00	Reset *new set point* bit in the *Controlword* object (value 004F$_h$)
→ ←	4537.9 4539.6	60D, 40 41 60 00 00 00 00 00 58D, 4B 41 60 00 37 12 00 00	Reading *Statusword* object waiting for *set-point acknowledge* bit reset (value 0237$_h$)
→ ←	…	…	Reading *Statusword* object waiting for *set-point acknowledge* bit reset (value 0237$_h$)
→ ←	4859.3 4861.1	60D, 40 41 60 00 00 00 00 00 58D, 4B 41 60 00 37 12 00 00	Reading *Statusword* object waiting for *set-point acknowledge* bit reset (value 0237$_h$)
→ ←	4924.4 4926.1	60D, 40 41 60 00 00 00 00 00 58D, 4B 41 60 00 37 02 00 00	Reading *Statusword* object waiting for *set-point acknowledge* bit reset (value 0237$_h$)
→ ←	4931.5 4933.1	60D, 23 7A 60 00 10 27 00 00 58D, 60 7A 60 00 00 00 00 00	*Target_position* object set at 10000 (1 complete clockwise revolution)
→ ←	4937.4 4939.1	60D, 2B 40 60 00 5F 00 00 00 58D, 60 40 60 00 00 00 00 00	Relative positioning and *new set point* bit set in the *Controlword* object (value 005F$_h$)
→ ←	4943.4 4945.1	60D, 40 41 60 00 00 00 00 00 58D, 4B 41 60 00 37 12 00 00	Reading *Statusword* object waiting for *set-point acknowledge* bit (value 1237$_h$)
→ ←	4949.7 4951.6	60D, 2B 40 60 00 4F 00 00 00 58D, 60 40 60 00 00 00 00 00	Reset *new set point* bit in the *Controlword* object (value 004F$_h$)
→ ←	4957.4 4959.1	60D, 40 41 60 00 00 00 00 00 58D, 4B 41 60 00 37 12 00 00	Reading *Statusword* object waiting for *set-point acknowledge* bit reset (value 0237$_h$)
→ ←	…	…	Reading *Statusword* object waiting for *set-point acknowledge* bit reset (value 0237$_h$)
→ ←	5546.4 5548.1	60D, 40 41 60 00 00 00 00 00 58D, 4B 41 60 00 37 12 00 00	Reading *Statusword* object waiting for *set-point acknowledge* bit reset (value 0237$_h$)

→	5609.2	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *set-point acknowledge* bit reset (value 0237_h)
←	5611.1	58D, 4B 41 60 00 37 02 00 00	
→	…	…	Reading *Statusword* object waiting for *Target-reached* bit (value 0637_h)
←			
→	5923.1	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *Target-reached* bit (value 0637_h)
←	5924.6	58D, 4B 41 60 00 37 02 00 00	
→	5986.5	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *Target-reached* bit (value 0637_h)
←	5988.1	58D, 4B 41 60 00 37 06 00 00	

In the following example the same motion cycle previously described is realized using the PDO for the process data exchange and the SDO protocol for the configuration only.

The TPDO2 is used in the default configuration to transmit to the master controller the object *Statusword* and the actual position of the motor (object *Position_actual_value*). The RPDO2, also in the default configuration, is used instead to set the *Controlword* and the position of the motor (object *Target_position*).

The drive used in the following example has the address $0D_h$ and remain valid the indications on the initial status described in the previous example.

→ ←	Time (ms)	COB-ID, Data	Description
→	0.0	60D, 23 81 60 00 C4 09 00 00	*Profile_velocity* object set with 2500 (250rpm)
←	1.5	58D, 60 81 60 00 00 00 00 00	
→	4.8	60D, 23 83 60 00 E8 03 00 00	*Profile_acceleration* object set with 1000 (1000rpm/s)
←	6.5	58D, 60 83 60 00 00 00 00 00	
→	8.8	60D, 23 84 60 00 D0 07 00 00	*Profile_deceleration* object set with 1000 (1000rpm/s)
←	10.5	58D, 60 84 60 00 00 00 00 00	
→	12.8	60D, 2F 60 60 00 01 00 00 00	*Modes_of_operation* object set with 1 (1 = *Profile position*)
←	14.5	58D, 60 60 60 00 00 00 00 00	
→	16.8	60D, 40 61 60 00 00 00 00 00	Reading *Modes_of_operation_display* object to check operating mode 1 active
←	18.5	58D, 4F 61 60 00 01 00 00 00	
→	20.8	60D, 23 00 18 01 8D 01 00 80	TPDO1 disabled
←	22.6	58D, 60 00 18 01 00 00 00 00	
→	24.7	60D, 23 01 14 01 0D 03 00 00	RPDO2 enabled
←	26.5	58D, 60 01 14 01 00 00 00 00	
→	28.8	60D, 2B 01 18 05 00 00 00 00	Event timer TPDO2 set to 0 (default 100ms)
←	30.5	58D, 60 01 18 05 00 00 00 00	
→	32.9	60D, 23 01 18 01 8D 02 00 00	TPDO2 enabled
←	34.5	58D, 60 01 18 01 00 00 00 00	
→	36.9	000, 01 0D	Set NMT in *Start* state
←	38.6	28D, 50 26 00 00 00 00	TPDO2, *Statusword* = 2650_h, *Position_A._V.* = 0 Set PDS in *Switch on disabled* state
→	41.0	30D, 06 00 00 00 00 00	RPDO2, *Controlword* = 0006_h, *Target_position* = 0 Set PDS in *Shutdown* state
←	43.1	28D, 31 26 00 00 00 00	TPDO2, *Statusword* = 2631_h, *Position_A._V.* = 0 Set PDS in *Ready to switch on* state
→	45.9	30D, 0F 00 00 00 00 00	RPDO2, *Controlword* = $000F_h$, *Target_position* = 0

			Set PDS in *Operation enabled* state
←	48.2	28D, 33 26 00 00 00 00	TPDO2, *Statusword* = 2633_h, *Position_A._V.* = 0 Set PDS in *Switched on* state
←	64.1	28D, 37 06 00 00 00 00	TPDO2, *Statusword* = 0637_h, *Position_A._V.* = 0 Set PDS in *Operation enabled* state
→	77.0	30D, 5F 00 20 4E 00 00	RPDO2, *Controlword* = $005F_h$, *Target_position* = 20000 Quote and relative positioning set. N*ew set point* bit = 1
←	78.7	28D, 37 12 00 00 00 00	TPDO2, *Statusword* = 1237_h, *Position_A._V.* = 0 *Set-point acknowledge* bit = 1
→	81.7	30D, 4F 00 20 4E 00 00	RPDO2, *Controlword* = $004F_h$, *Target_position* = 20000 Set *New set point* bit = 0
←	83.6	28D, 37 02 02 00 00 00	TPDO2, *Statusword* = 0237_h, *Position_A._V.* = 0 *Set-point acknowledge* bit = 0
←	739.6	28D, 37 06 17 4E 00 00	TPDO2, *Statusword* = 0637_h, *Position_A._V.* = 19991 *Target-reached* bit = 1
→	1767.7	30D, 5F 00 A0 15 FF FF	RPDO2, *Controlword* = $005F_h$, *Target_Position* = -60000 Quote and relative positioning set. *New set point* bit = 1
←	1769.6	28D, 37 12 20 4E 00 00	TPDO2, *Statusword* = 1237_h, *Position_A._V.* = 20000 *Set-point acknowledge* bit = 1
→	1775.5	30D, 4F 00 A0 15 FF FF	RPDO2, *Controlword* = $004F_h$, *Target_position* = -60000 Set *New set point* bit = 0
←	1777.1	28D, 37 02 1C 4E 00 00	TPDO2, *Statusword* = 0237_h, *Position_A._V.* = 19996 *Set-point acknowledge* bit = 0
←	3390.1	28D, 37 06 CA 63 FF FF	TPDO2, *Statusword* = 0637_h, *Position_A._V.* = -39990 *Target-reached* bit = 1
→	4397.9	30D, 5F 00 10 27 00 00	RPDO2, *Controlword* = $005F_h$, *Target_position* = 10000 Quote and relative positioning set. *New set point* bit = 1
←	4399.7	28D, 37 12 C0 63 FF FF	TPDO2, *Statusword* = 0637_h, *Position_A._V.* = -40000 *Set-point acknowledge* bit = 1
→	4406.3	30D, 4F 00 10 27 00 00	RPDO2, *Controlword* = $004F_h$, *Target_position* = 10000 Set *New set point* bit = 0
←	4408.1	28D, 37 02 C6 63 FF FF	TPDO2, *Statusword* = 0237_h, *Position_A._V.* = -39994 *Set-point acknowledge* bit = 0
→	4413.0	30D, 5F 00 20 4E 00 00	RPDO2, *Controlword* = $005F_h$, *Target_position* = 20000 Quote and relative position set. *New set point* bit = 1
←	4414.6	28D, 37 12 D2 63 FF FF	TPDO2, *Statusword* = 0237_h, *Position_A._V.* = -39982 *Set-point acknowledge* bit = 1
→	4419.5	30D, 4F 00 20 4E 00 00	RPDO2, *Controlword* = $004F_h$, *Target_position* = 20000 *New set point* bit = 0
←	4821.1	28D, 37 02 C7 8A FF FF	TPDO2, *Statusword* = 0237_h, *Position_A._V.* = -30009 *Set-point acknowledge* bit = 0
→	4822.3	30D, 5F 00 10 27 00 00	RPDO2, *Controlword* = $005F_h$, *Target_position* = 10000 Quote and relative positioning set. *New set point* bit = 1
←	4824.1	28D, 37 12 D0 8A FF FF	TPDO2, *Statusword* = 1237_h, *Position_A._V.* = -30000

			Set-point acknowledge bit = 1 *Set New set point* bit = 0
→	4830.9	30D, 4F 00 10 27 00 00	RPDO2, *Controlword* = $004F_h$, *Target_Position* = 10000 Set *new set point* bit = 0
←	5467.6	28D, 37 02 E8 D8 FF FF	TPDO2, *Statusword* = 0237_h, *Position_A._V.* = -10008 *Set-point acknowledge* bit = 0
←	5873.6	28D, 37 06 F7 FF FF FF	TPDO2, *Statusword* = 0237_h, *Position_A._V.* = -9 *Target-reached* bit = 1

Note that the bit *Target-reached* becomes active before reaching the commanded position because of the object *Position_window* that by default is set to value 10. In this way the motor is considered “in position” (bit *Target-reached* = 1) each time the difference between the actual position and the commanded one is less than 10 in absolute value. If you want to have the bit *Target-reached* active at the reaching of the exact target position it is sufficient to set the object *Position_window* equal to 0.

4.4.2 Profile velocity mode (pv)

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

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

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

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

		UINT32 RPDO	
6085 h	*Quick_stop_deceleration*	rpm/s UINT32 RPDO	Indicate the configured deceleration used to stop the motor when the quick stop function is activated
60F8 h	*Max_slippage*	0,1rpm INT32 TPDO	Indicate the configured maximal slippage
60FF h	*Target_velocity*	0.1rpm INT32 TPDO	Indicate the configured target velocity

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

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

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

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

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

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

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

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

The minimum steps required to rotate the motor in the *Profile velocity* mode are the configuration of the motion profile (acceleration and deceleration) and the setting of the target velocity. This can be carried out through the communication objects SDO, PDO or a combination of the two. In the following example you use the only SDO protocol to set 3 different speeds. Each speed is maintained active for 3 seconds.

The drive used in the following example has the address $0D_h$ and assumes to have been started with the default values and successively configured with the minimum settings described in the previous chapters (Motor parameters and Running and idle current configuration). Furthermore, NMT is considered in the *Pre-Operational* status (default status after the power on).

The values in the Time columns refer to a bit rate of 250Kbit/s and can vary according to the traffic on the bus and to the reaction time of the master controller used, as well as to the firmware revision installed in the drive. The symbol → indicates a data flow from the bus to the drive while the symbol ← indicates a data flow from the drive to the CANopen bus. The communications highlighted in pale blue are those required to reach the first speed, the subsequent lines, instead, show the evolution of the entire cycle. The motion cycle shown in the example provides the rotation of the motor at 3 different speeds lasting 3 seconds each.

→ ←	Time (ms)	COB-ID, Data	Description
→	0.0	60D, 23 83 60 00 E8 03 00 00	*Profile_acceleration* object set with 1000 (1000rpm/s)
←	1.9	58D, 60 83 60 00 00 00 00 00	
→	5.2	60D, 23 84 60 00 D0 07 00 00	*Profile_deceleration* object set with 2000 (2000rpm/s)
←	6.9	58D, 60 84 60 00 00 00 00 00	
→	10.4	60D, 2F 60 60 00 03 00 00 00	*Modes_of_operation* object set with 3 (3 = *Profile_velocity*)
←	11.9	58D, 60 60 60 00 00 00 00 00	
→	15.6	60D, 40 61 60 00 00 00 00 00	Reading *Modes_of_operation_display* object to check operating mode 3 active
←	17.4	58D, 4F 61 60 00 03 00 00 00	
→	23.0	60D, 2B 40 60 00 06 00 00 00	*Controlword* object set with 0006_h (Shutdown)
←	24.4	58D, 60 40 60 00 00 00 00 00	
→	28.9	60D, 2B 40 60 00 0F 00 00 00	*Controlword* object set with $000F_h$ (*Switch on + enable operation*)
←	30.4	58D, 60 40 60 00 00 00 00 00	
→	34.2	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *Operation enabled* state (1637_h)
←	35.9	58D, 4B 41 60 00 33 32 00 00	
→	99.9	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *Operation enabled* state (1637_h)
←	101.4	58D, 4B 41 60 00 37 16 00 00	
→	105.5	60D, 23 FF 60 00 2C 01 00 00	*Target_velocity* object set with 300 (30rpm counterclockwise)
←	107.4	58D, 60 FF 60 00 00 00 00 00	
→	110.6	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *Target-reached* bit (value 0637_h)
←	112.4	58D, 4B 41 60 00 37 12 00 00	
→	161.5	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *Target-reached* bit (value 0637_h)
←	163.4	58D, 4B 41 60 00 37 06 00 00	
→	3172.5	60D, 23 FF 60 00 48 F4 FF FF	*Target_velocity* object set with -3000 (300rpm counterclockwise)
←	3174.4	58D, 60 FF 60 00 00 00 00 00	
→	3178.1	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *Target-reached* bit (value 0637_h)
←	3179.9	58D, 4B 41 60 00 37 02 00 00	
→	…	…	Reading *Statusword* object waiting for *Target-reached* bit (value 0637_h)
←			
→	3429.2	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *Target-reached* bit (value 0637_h)
←	3430.9	58D, 4B 41 60 00 37 02 00 00	
→	3493.0	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *Target-reached* bit (value 0637_h)
←	3494.9	58D, 4B 41 60 00 37 06 00 00	
→	6503.7	60D, 23 FF 60 00 D0 07 00 00	*Target_velocity* object set with 2000 (200rpm clockwise)
←	6505.4	58D, 60 FF 60 00 00 00 00 00	
→	6507.8	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *Target-reached* bit (value 0637_h)
←	6509.4	58D, 4B 41 60 00 37 02 00 00	
→	…	…	Reading *Statusword* object waiting for *Target-reached* bit (value 0637_h)
←			
→	6817.9	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *Target-reached* bit (value 0637_h)
←	6819.9	58D, 4B 41 60 00 37 02 00 00	
→	6883.1	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *Target-reached* bit (value 0637_h)
←	6884.8	58D, 4B 41 60 00 37 06 00 00	
→	9892.8	60D, 23 FF 60 00 00 00 00 00	*Target_velocity* object set with 0 (0rpm)
←	9894.4	58D, 60 FF 60 00 00 00 00 00	
→	9900.4	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *Target-reached* and *Speed* bit (value 1637_h)
←	9901.9	58D, 4B 41 60 00 37 02 00 00	
→	9962.7	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *Target-reached*

←	9964.4	58D, 4B 41 60 00 37 02 00 00	and *Speed* bit (value 1637_h)
→ ←	9986.6 9988.3	60D, 40 41 60 00 00 00 00 00 58D, 4B 41 60 00 37 02 00 00	Reading *Statusword* object waiting for *Target-reached* and *Speed* bit (value 1637_h)
→ ←	10022.8 10024.4	60D, 40 41 60 00 00 00 00 00 58D, 4B 41 60 00 37 16 00 00	Reading *Statusword* object waiting for *Target-reached* and e *Speed* bit (value 1637_h)

In the following example the same motion cycle previously described is realized using the PDO for the process data exchange and the SDO protocol for the configuration only.

The TPDO3 is used in the default configuration to transmit to the master controller the object *Statusword* and the actual speed of the motor (object *Velocity_actual_value*). The RPDO3, also in the default configuration, is used instead to set the *Controlword* and the speed of the motor (object *Target_velocity*).

The drive used in the following example has the address $0D_h$ and remain valid the indications on the initial status described in the previous example.

→ ←	Time (ms)	COB-ID, Data	Description
→ ←	0.0 1.9	60D, 23 83 60 00 E8 03 00 00 58D, 60 83 60 00 00 00 00 00	*Profile_acceleration* object set with 1000 (1000rpm/s)
→ ←	3.6 5.4	60D, 23 84 60 00 D0 07 00 00 58D, 60 84 60 00 00 00 00 00	*Profile_deceleration* object set with 2000 (2000rpm/s)
→ ←	6.6 8.4	60D, 2F 60 60 00 03 00 00 00 58D, 60 60 60 00 00 00 00 00	*Modes_of_operation* object set with 3 (3 = *Profile_velocity*)
→ ←	9.8 11.4	60D, 40 61 60 00 00 00 00 00 58D, 4F 61 60 00 03 00 00 00	Reading *Modes_of_operation_display* object to check operating mode 3 active
→ ←	12.5 14.4	60D, 23 00 18 01 8D 01 00 80 58D, 60 00 18 01 00 00 00 00	TPDO1 disabled
→ ←	15.8 17.4	60D, 23 02 14 01 0D 04 00 00 58D, 60 02 14 01 00 00 00 00	RPDO3 enabled
→ ←	18.5 20.4	60D, 2B 02 18 05 00 00 00 00 58D, 60 02 18 05 00 00 00 00	Event timer TPDO3 set to 0 (default 100ms)
→ ←	21.6 23.4	60D, 23 02 18 01 8D 03 00 00 58D, 60 02 18 01 00 00 00 00	TPDO3 enabled
→	47.0	000, 01 0D	Set NMT in the *Start* state
←	48.5	38D, 50 32 00 00 00 00	TPDO3, *Statusword* = 3250_h, *Velocity_A._V.* = 0 Set PDS in *Switch on disabled* state
→	49.8	40D, 06 00 00 00 00 00	RPDO3, *Controlword* = 0006_h, *Target_Velocity* = 0 Set PDS in *Shutdown* state
←	52.0	38D, 31 32 00 00 00 00	TPDO3, *Statusword* = 3231_h, *Velocity_A._V.* = 0 Set PDS in *Ready to switch on* state
→	53.7	40D, 0F 00 00 00 00 00	RPDO3, *Controlword* = $000F_h$, *Target_Velocity* = 0 Set PDS in *Operation enabled* state
←	56.0	38D, 33 32 00 00 00 00	TPDO3, *Statusword* = 3233_h, *Velocity_A._V.* = 0 Set PDS in *Switched on* state
←	71.5	38D, 33 36 00 00 00 00	TPDO3, *Statusword* = 3633_h, *Velocity_A._V.* = 0 Set PDS in *Switched on* state
←	72.0	38D, 37 16 00 00 00 00	TPDO3, *Statusword* = 1637_h, *Velocity_A._V.* = 0 Set PDS in *Operation enabled* state
→	73.9	40D, 0F 00 2C 01 00 00	RPDO3, *Controlword* = $000F_h$, *Target_velocity* = 300

			(30rpm)
←	75.5	38D, 37 12 00 00 00 00	TPDO3, *Statusword* = 1237_h, *Velocity_A._V.* = 0 *Target-reached* bit = 0, *Speed* bit = 1
←	82.5	38D, 37 02 44 00 00 00	TPDO3, *Statusword* = 0237_h, *Velocity_A._V.* = 68 (6.8rpm) *Target-reached* bit = 0, *Speed bit* = 0
←	96.5	38D, 37 06 D0 00 00 00	TPDO3, *Statusword* = 0637_h, *Velocity_A._V.* = 208 (20.8rpm) *Target-reached* bit = 1, *Speed* bit = 0
→	3100.4	40D, 0F 00 B8 0B 00 00	RPDO3, *Controlword* = $000F_h$, *Target_velocity* = 3000 (300rpm)
←	3101.9	38D, 37 02 2C 01 00 00	TPDO3, *Statusword* = 0237_h, *Velocity_A._V.* = 300 (30rpm) *Target-reached* bit = 0, *Speed* bit = 0
←	3362.4	38D, 37 06 56 0B 00 00	TPDO3, *Statusword* = 0637_h, *Velocity_A._V.* = 2902 (290.2rpm) *Target-reached* bit = 1, *Speed* bit = 0
→	6369.6	40D, 0F 00 D0 07 00 00	RPDO3, *Controlword* = $000F_h$, *Target_velocity* = 2000 (200rpm)
←	6371.5	38D, 37 02 B8 0B 00 00	TPDO3, *Statusword* = 0237_h, *Velocity_A._V.* = 3000 (300rpm) *Target-reached* bit = 0, *Speed* bit = 0
←	6417.5	38D, 37 06 24 08 00 00	TPDO3, *Statusword* = 0637_h, *Velocity_A._V.* = 2084 (208.4rpm) *Target-reached* bit = 1, *Speed* bit = 0
→	9421.8	40D, 0F 00 00 00 00 00	RPDO3, *Controlword* = $000F_h$, *Target_velocity* = 0 (0rpm)
←	9423.5	38D, 37 02 D0 07 00 00	TPDO3, *Statusword* = 0237_h, *Velocity_A._V.* = 2000 (30rpm) *Target-reached* bit = 0, *Speed* bit = 0
←	9519.5	38D, 37 06 54 00 00 00	TPDO3, *Statusword* = 0637_h, *Velocity_A._V.* = 84 (8.4rpm) *Target-reached* bit = 1, *Speed* bit= 0
←	9521.5	38D, 37 16 2C 00 00 00	TPDO3, *Statusword* = 1637_h, *Velocity_A._V.* = 44 (4.4rpm) *Target-reached* bit = 1, *Speed* bit = 1

Note that the bit *Target-reached* becomes active before reaching the commanded speed because of the object *Velocity_window* that by default is set to value 100. In this way the motor is considered “in velocity” (bit *Target-reached* = 1) each time the difference between the actual speed and the commanded one is less than 100 in absolute value. If you want to have the bit *Target-reached* active at the reaching of the exact target velocity, it is sufficient to set the object *Velocity_window* equal to 0.

Also the commutation threshold of the bit *Speed* can be modified operating on the object *Velocity_threshold* that in the example is set to 60. In this way the motor is considered stopped (bit *Speed* = 1) each time the actual speed of the motor is lower than 60 in absolute value. If you want to have the bit *Speed* active only when the speed is exactly equal to 0 it is sufficient to set the object *Velocity_threshold* equal to 0.

4.4.3 Profile torque mode (tq)

The *Profile 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 object *Target_torque*. The torque variation on the motor shaft always occurs according to the ramp set through the object *Torque_slope*.

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

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

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

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

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

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

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

Controlword bits organization

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

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

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

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

The minimum steps required to regulate the torque to the motor shaft in the *Profile torque* mode are the configuration of the ramp and the setting of the target torque. This can be done through the communication objects SDO, PDO or a combination of the two. In the following example you use the only SDO protocol to set 2 different torque values. Each value is maintained for 3 seconds.

The drive used in the following example has the address $0D_h$ and assumes to have been started with the default values and successively configured with the minimum settings described in the previous chapters (Motor parameters and Running and idle current configuration). Furthermore, it is necessary that the motor encoder is correctly configured through the objects 2330_h:01_h *Configuration_ENCMTR* and 2330_h:02_h *CPR_ENCMTR*, and activated the encoder feedback through the object 2410_h:01_h *Feedback_MTNSTP*. In the end, NMT is considered in the *Pre-Operational* status (default status after the power on).

The values in the Time column refer to a bit rate of 250Kbit/s and can vary according to the traffic on the bus and to the reaction time of the master controller used, as well as to to the firmware revision installed in the drive. The symbol → indicates a data flow from the bus to the drive while the symbol ← indicates a data flow from the drive to the CANopen bus. The communications highlighted in pale blue are those required to reach the first torque value, the subsequent lines, instead, show the evolution of the entire cycle. The cycle shown in the example provides the setting of the torque to motor shaft at 2 different values, lasting 3 seconds each.

→ ←	Time (ms)	COB-ID, Data	Description
→	0.0	60D, 23 87 60 00 F4 01 00 00	*Torque_sloper* object set with 500 (50% variation in 1 second)
←	1.5	58D, 60 87 60 00 00 00 00 00	
→	5.2	60D, 2B 10 23 0B B8 0B 00 00	*Max_Speed_Motor_Data* object set with 3000 (300rpm)
←	7.0	58D, 60 10 23 0B 00 00 00 00	
→	9.6	60D, 2F 60 60 00 04 00 00 00	*Mode_of_operation* object set with 4 (4 = *Profile_torque*)
←	11.5	58D, 60 60 60 00 00 00 00 00	
→	21.4	60D, 40 61 60 00 00 00 00 00	Reading *Mode_of_operation_display* object to check operating mode 4 active
←	23.0	58D, 4F 61 60 00 04 00 00 00	
→	25.7	60D, 2B 40 60 00 06 00 00 00	*Controlword* object set with 0006_h (Shutdown)
←	27.5	58D, 60 40 60 00 00 00 00 00	
→	30.0	60D, 2B 40 60 00 0F 00 00 00	*Controlword* object set with $000F_h$ (*Switch on + enable operation*)
←	32.0	58D, 60 40 60 00 00 00 00 00	
→	34.5	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *Operation enabled* state (0637_h)
←	36.0	58D, 4B 41 60 00 33 02 00 00	
→	…	…	Reading *Statusword* object waiting for *Operation enabled* state (0637_h)
←			
→	510.5	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *Operation enabled* state (0637_h)
←	512.0	58D, 4B 41 60 00 33 02 00 00	
→	578.5	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *Operation enabled* state (0637_h)
←	580.0	58D, 4B 41 60 00 37 06 00 00	
→	586.6	60D, 2B 71 60 00 C8 00 00 00	*Target_torque* object set to 200 (20% of rated torque)
←	588.5	58D, 60 71 60 00 00 00 00 00	
→	592.9	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *Target-reached* bit (value 0637_h)
←	594.4	58D, 4B 41 60 00 37 02 00 00	
→	…	…	Reading *Statusword* object waiting for *Target-reached* bit (value 0637_h)
←			
→	948.3	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *Target-reached* bit (value 0637_h)
←	950.0	58D, 4B 41 60 00 37 02 00 00	

→	1016.0	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *Target-reached* bit (value 0637_h)
←	1017.5	58D, 4B 41 60 00 37 06 00 00	
→	4060.9	60D, 2B 71 60 00 F4 01 00 00	*Target_torque* object set to 200 (20% of rated torque)
←	4062.5	58D, 60 71 60 00 00 00 00 00	
→	4068.4	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *Target-reached* bit (value 0637_h)
←	4070.0	58D, 4B 41 60 00 37 02 00 00	
→	…	…	Reading *Statusword* object waiting for *Target-reached* bit (value 0637_h)
←			
→	4651.3	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *Target-reached* bit (value 0637_h)
←	4653.0	58D, 4B 41 60 00 37 02 00 00	
→	4713.5	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *Target-reached* bit (value 0637_h)
←	4715.5	58D, 4B 41 60 00 37 06 00 00	
→	7727.1	60D, 2B 71 60 00 00 00 00 00	*Target_torque* object set with 0 (0% of nominal torque)
←	7728.9	58D, 60 71 60 00 00 00 00 00	
→	7734.5	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *Target-reached* bit (value 0637_h)
←	7735.9	58D, 4B 41 60 00 37 02 00 00	
→	…	…	*Target_position* object set with 0 (0rpm)
←			
→	8697.9	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *Target-reached* bit (value 0637_h)
←	8699.4	58D, 4B 41 60 00 37 02 00 00	
→	8763.6	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *Target-reached* bit (value 0637_h)
←	8765.4	58D, 4B 41 60 00 37 06 00 00	

In the following example the same cycle previously described is realized using the PDO for the process data exchange and the SDO protocol for the configuration only.

The TPDO4 is used to transmit to the master controller the object *Statusword* and the actual torque available on the motor shaft (object *Torque_actual_value*). The RPDO4 is used instead to set the *Controlword* and the target torque (object *Target_torque*).

The drive used in the example has address $0D_h$ and remain valid the indications on the initial status described in previous example.

→ ←	Time (ms)	COB-ID, Data	Description
→	0.0	60D, 23 87 60 00 F4 01 00 00	*Torque_slope* object set with 500 (50% variation in 1 second)
←	1.8	58D, 60 87 60 00 00 00 00 00	
→	22.8	60D, 2B 10 23 0B B8 0B 00 00	*Max_Speed_MTRDT* object set with 3000 (300rpm)
←	24.3	58D, 60 10 23 0B 00 00 00 00	
→	27.8	60D, 23 00 18 01 8D 01 00 80	TPDO1 disabled
←	29.3	58D, 60 00 18 01 00 00 00 00	
→	32.7	60D, 23 03 14 01 0D 05 00 80	RPDO4 disabled
←	34.3	58D, 60 03 14 01 00 00 00 00	
→	37.7	60D, 2F 03 14 02 FF 00 00 00	*Transmission type* 255 set (0xFF)
←	39.3	58D, 60 03 14 02 00 00 00 00	
→	43.2	60D, 2F 03 16 00 00 00 00 00	Set number of elements mapped to 0
←	44.8	58D, 60 03 16 00 00 00 00 00	

→	48.4	60D, 23 03 16 01 10 00 40 60	*Controlword* mapping in RPDO4
←	49.8	58D, 60 03 16 01 00 00 00 00	
→	53.6	60D, 23 03 16 02 10 00 71 60	*Target_torque* mapping in RPDO4
←	55.3	58D, 60 03 16 02 00 00 00 00	
→	58.8	60D, 2F 03 16 00 02 00 00 00	Set number of elements mapped to 2
←	60.3	58D, 60 03 16 00 00 00 00 00	
→	64.1	60D, 23 03 14 01 0D 05 00 00	RPDO4 enabled
←	65.8	58D, 60 03 14 01 00 00 00 00	
→	69.5	60D, 23 03 18 01 8D 04 00 80	TPDO4 disabled
←	71.3	58D, 60 03 18 01 00 00 00 00	
→	74.9	60D, 2F 03 18 02 FF 00 00 00	*Transmission type* 255 set (0xFF)
←	76.8	58D, 60 03 18 02 00 00 00 00	
→	80.2	60D, 2F 03 1A 00 00 00 00 00	Set number of elements mapped to 0
←	81.8	58D, 60 03 1A 00 00 00 00 00	
→	85.0	60D, 23 03 1A 01 10 00 41 60	*Statusword* mapping in TPDO4
←	86.8	58D, 60 03 1A 01 00 00 00 00	
→	90.7	60D, 23 03 1A 02 10 00 77 60	*Torque_actual_value* mapping in TPDO4
←	92.3	58D, 60 03 1A 02 00 00 00 00	
→	95.1	60D, 2F 03 1A 00 02 00 00 00	Set number of elements mapped to 2
←	96.8	58D, 60 03 1A 00 00 00 00 00	
→	99.2	60D, 23 03 18 01 8D 04 00 00	TPDO4 enabled
←	100.8	58D, 60 03 18 01 00 00 00 00	
→	103.5	60D, 2F 60 60 00 04 00 00 00	*Modes_of_operation* object set with 4 (4 = *Profile_torque*)
←	105.3	58D, 60 60 60 00 00 00 00 00	
→	108.3	60D, 40 61 60 00 00 00 00 00	Reading *Modes_of_operation_display* object to check operating mode 4 active
←	109.8	58D, 4F 61 60 00 04 00 00 00	
→	111.9	000, 01 0D	Set NMT in *Start* state
←	113.3	48D, 50 02 00 00	TPDO4, *Statusword* = 0250_h, *Torque_A._V.* = 0 Set PDS in *Switch on disabled* state
→	115.2	50D, 06 00 00 00	RPDO4, *Controlword* = 0006_h, *Target_torque* = 0 Set PDS in *Shutdown* state
←	117.8	48D, 31 02 00 00	TPDO4, *Statusword* = 0231_h, *Torque_A._V.* = 0 Set PDS in *Ready to switch on* state
→	119.7	50D, 0F 00 00 00	RPDO4, *Controlword* = $000F_h$, *Target_ torque* = 0 Set PDS in *Operation enabled* state
←	121.8	48D, 33 02 00 00	TPDO4, *Statusword* = 0233_h, *Torque_A._V.* = 0 Set PDS in *Switched on* state
←	636.8	48D, 37 06 00 00	TPDO4, *Statusword* = 0637_h, *Velocity_A._V.* = 0 Set PDS in *Operation enabled* state
→	638.1	50D, 0F 00 C8 00	RPDO4, *Controlword* = $000F_h$, *Target_ torque* = 100
←	639.8	48D, 37 02 00 00	TPDO4, *Statusword* = 0237_h, *Velocity_A._V.* = 0 *Target-reached* bit = 0
←	1039.8	48D, 37 06 C8 00	TPDO4, *Statusword* = 0637_h, *Velocity_A._V.* = 100 *Target-reached* bit = 1
→	4046.7	50D, 0F 00 F4 01	RPDO4, *Controlword* = $000F_h$, *Target_ torque* = 500
←	4048.3	48D, 37 02 C8 00	TPDO4, *Statusword* = 0237_h, *Velocity_A._V.* = 100 *Target-reached* bit = 0

← 4648.3 48D, 37 06 F4 01	TPDO4, *Statusword* = 0637_h, *Velocity_A._V.* = 500 *Target-reached* bit = 1
→ 7655.0 50D, 0F 00 00 00	RPDO4, *Controlword* = $000F_h$, *Target_ torque* = 0
← 7656.8 48D, 37 02 F3 01	TPDO4, *Statusword* = 0237_h, *Velocity_A._V.* = 499 *Target-reached* bit = 0
← 8656.8 48D, 37 06 00 00	TPDO4, *Statusword* = 0637_h, *Velocity_A._V.* = 100 *Target-reached* bit = 1

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

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

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

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

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

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

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

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

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

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

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

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

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

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

The sensors connected with the 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 object 2810_h *Digital_Input_Action*.

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

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

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

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

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

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

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

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

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

The drive used in the following example has the address $0D_h$ and assumes to have been started with the default value and successively configured with the minimum setting described in the previous chapters (Motor Parameters and Running and idle current configuration). Furthermore, NMT is considered in the *Pre-Operational* status (default status after the power on).

The value in the Time column refer to a bit rate of 250Kbit/s and can vary according to the traffic on the bus and to the reaction time of the master controlled used, as well as to the firmware revision installed in the drive. The symbol → indicates a data flow from the bus to the drive while the symbol ← indicates a data flow from the device to the CANopen bus.

→ ←	Time (ms)	COB-ID, Data	Description
→	0.0	60D, 2B 10 28 05 05 02 00 00	Assign DI5 input to the home switch and set active (active when powered)
←	1.9	58D, 60 10 28 05 00 00 00 00	
→	6,7	60D, 2B 10 28 04 06 02 00 00	Assign DI6 input to the counterclockwise limit switch and set active (active when powered)
←	8.4	58D, 60 10 28 04 00 00 00 00	
→	12.0	60D, 2B 10 28 03 07 02 00 00	Assign DI7 input to the clockwise home switch and set active (active when powered)
←	13.9	58D, 60 10 28 03 00 00 00 00	
→	17.3	60D, 23 83 60 00 C8 00 00 00	*Profile_acceleration* object set with 200 (200rpm/s)
←	18.9	58D, 60 83 60 00 00 00 00 00	
→	22.4	60D, 23 84 60 00 F4 01 00 00	*Profile_deceleration* object set with 500 (500rpm/s)
←	23.9	58D, 60 84 60 00 00 00 00 00	
→	27.7	60D, 23 99 60 01 2C 01 00 00	Search for switch speed set to 300 (30rpm)
←	29.4	58D, 60 99 60 01 00 00 00 00	
→	33.0	60D, 23 99 60 02 32 00 00 00	Search for homing speed set to 50 (5rpm)
←	34.4	58D, 60 99 60 02 00 00 00 00	
→	38.0	60D, 2F 60 60 00 06 00 00 00	*Modes_of_operation* object set with 6 (6 = *Profile_velocity*)
←	39.9	58D, 60 60 60 00 00 00 00 00	
→	43.0	60D, 40 61 60 00 00 00 00 00	Reading *Modes_of_operation_display* object to check operating mode 6 active
←	44.9	58D, 4F 61 60 00 06 00 00 00	
→	51.2	60D, 2F 98 60 00 01 00 00 00	Homing mode selection 1
←	52.9	58D, 60 98 60 00 00 00 00 00	
→	56.3	60D, 2B 40 60 00 06 00 00 00	*Controlword* object set with 0006_h (Shutdown)
←	57.9	58D, 60 40 60 00 00 00 00 00	
→	61.7	60D, 2B 40 60 00 0F 00 00 00	*Controlword* object set with $000F_h$ (*Switch on + enable operation*)
←	63.4	58D, 60 40 60 00 00 00 00 00	
→	66.9	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *Operation enabled* state (1637_h)
←	68.4	58D, 4B 41 60 00 33 16 00 00	
→	…	…	Reading *Statusword* object waiting for *Operation enabled* state (1637_h)
←			
→	547.3	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *Operation enabled* state (1637_h)
←	548.9	58D, 4B 41 60 00 33 16 00 00	
→	608.3	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *Operation enabled* state (1637_h)
←	609.9	58D, 4B 41 60 00 37 16 00 00	
→	613.0	60D, 2B 40 60 00 1F 00 00 00	*Homing_operation_start* bit set to 1 for the homing start
←	615.0	58D, 60 40 60 00 00 00 00 00	
→	616.5	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *Target-reached* and *Homing_attained* bit (value 1637_h)
←	618.4	58D, 4B 41 60 00 37 02 00 00	
→	…	…	Reading *Statusword* object waiting for the *Target-reached* and *Homing_attained* bit (value 1637_h)
←			
→	10106.8	60D, 40 41 60 00 00 00 00 00	Reading *Statusword* object waiting for *Target-reached* and *Homing_attained* bit (value 1637_h)
←	10108.3	58D, 4B 41 60 00 37 16 00 00	
→	10114.6	60D, 2B 40 60 00 0F 00 00 00	*Homing_operation_start* bit set to 0
←	10116.4	58D, 60 40 60 00 00 00 00 00	

In the following example the homing is performed using the PDO for the process data exchange and the SDO protocol for the configuration only.

The TPDO3 is used in the default configuration to transmit to the master controller the object *Statusword* and the actual position of the motor (object *Velocity_actual_value*). The RPDO3, also in the default configuration, is used instead to set the *Controlword* and the speed of the motor (object *Target_velocity*).

The driver has address $0D_h$ e remain valid the indications on the initial status described for the previous example.

→ ←	Time (ms)	COB-ID, Data	Description
→	0.0	60D, 2B 10 28 05 05 02 00 00	Assign DI5 input to the home switch and set active (active when powered)
←	1.5	58D, 60 10 28 05 00 00 00 00	
→	6.8	60D, 2B 10 28 04 06 02 00 00	Assign DI6 input to the counterclockwise limit switch and set active (active when powered)
←	8.5	58D, 60 10 28 04 00 00 00 00	
→	11.7	60D, 2B 10 28 03 07 02 00 00	Assign DI7 input to the clockwise home switch and set active (active when powered)
←	13.5	58D, 60 10 28 03 00 00 00 00	
→	17.4	60D, 23 83 60 00 C8 00 00 00	*Profile_acceleration* object set with 200 (200rpm/s)
←	19.0	58D, 60 83 60 00 00 00 00 00	
→	22.6	60D, 23 84 60 00 F4 01 00 00	*Profile_deceleration* object set with 500 (500rpm/s)
←	24.5	58D, 60 84 60 00 00 00 00 00	
→	27.6	60D, 23 99 60 01 2C 01 00 00	Search for switch speed set to 300 (30rpm)
←	29.5	58D, 60 99 60 01 00 00 00 00	
→	32.6	60D, 23 99 60 02 32 00 00 00	Search for homing speed set to 50 (5rpm)
←	34.5	58D, 60 99 60 02 00 00 00 00	
→	37.7	60D, 2F 60 60 00 06 00 00 00	*Modes_of_operation* object set with 6 (6 = *Profile_velocity*)
←	39.5	58D, 60 60 60 00 00 00 00 00	
→	43.3	60D, 40 61 60 00 00 00 00 00	Reading *Modes_of_operation_display* object to check operating mode 6 active
←	45.0	58D, 4F 61 60 00 06 00 00 00	
→	48.8	60D, 2F 98 60 00 01 00 00 00	Homing mode selection 1
←	50.5	58D, 60 98 60 00 00 00 00 00	
→	54.0	000, 01 0D	Set NMT in *Start* state
←	55.5	18D, 50 06	TPDO1, *Statusword* = 0650_h, Set PDS in *Switch on disabled* state
→	59.1	20D, 06 00	RPDO1, *Controlword* = 0006_h, Set PDS in *Shutdown* state
←	61.5	18D, 31 06	TPDO1, *Statusword* = 0631_h, Set PDS in *Ready to switch on* state
→	64.3	20D, 0F 00	RPDO1, *Controlword* = $000F_h$, Set PDS in *Operation enabled* state
←	66.5	18D, 33 06	TPDO1, *Statusword* = 0633_h, Set PDS in *Switched on* state
←	581.5	18D, 37 06	TPDO1, *Statusword* = 0637_h, Set PDS in *Switched on* state
→	584.6	20D, 1F 00	RPDO1, *Controlword* = $001F_h$, *Homing_operation_start* = 1
←	586.5	18D, 37 02	TPDO1, *Statusword* = 0237_h, bit *Homing_attained* = 0, *Target-reached* = 0
←	10165.9	18D, 37 12	TPDO1, *Statusword* = 1237_h, bit *Homing_attained* = 1, *Target-reached* = 0
←	10169.4	18D, 37 16	TPDO1, *Statusword* = 1637_h, bit *Homing_attained* = 1, *Target-reached* = 1

→ 10173.3 20D, 0F 00	RPDO1, *Controlword* = 000F$_h$, *Homing_operation_start* = 0

4.4.5 Interpolated position (ip)

The *Interpolated position* mode allows to command the drive the execution of a positioning path through the transmission of consecutive set-points. For a best operation the set-points must be transmitted with a known and regular frequency (period, cycle time), furthermore if the application requires multiple axis interpolated amongst them, it is necessary that the set-points are applied in a synchronous mode to all the axis involved in the interpolation. This result can be obtained through the use of PDO configured to be synchronously processed at the receiving of the SYNC object.

The drive is provided with an internal micro-interpolator able to interpolate the path between two set-points with micro intermediate positions in order to improve the movement of the motor and to make it continuous, even with slow update frequencies. Through the object *Dampening_A_Value_IPCNF* it is also possible to filter eventual discontinuities of the path, for example due to an irregular update of the set-points, and so improve the movement.

The motor speed is limited by the objects *Max_profile_velocity* and *Profile_velocity*. Optionally it is also possible to enable the acceleration and deceleration limitation (configurable through the objects *Profile_acceleration* and *Profile_deceleration* respectively).

The objects that modify the behavior of the *Interpolated position* mode are contained in the *Interpolated Position Configuration* record.

It has finally been provided a complete monitoring system to verify the correct receipt of the SYNC object, able to detect the receiving in advance or delayed with respect to the nominal period. By changing the alarm threshold it is possible to trigger an emergency at the first timing violation or tolerate sporadic violations maintaining the alarm condition only for repeated violations in the receiving of the SYNC object. All the objects for the configuration and monitoring of the SYNC are contained in the *Sync Guard* record.

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

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

		RPDO	to the position demand value
6066 h	*Following_error_time_out*	ms UINT16 RPDO	Indicate the configured time for a following error condition, after that the bit 13 of the *Statusword* is set to 1
6067 h	*Position_window*	0.0001rev UINT32 RPDO	Indicate the configured symmetrical range of accepted positions relative to target position
6068 h	*Position_window_time*	ms UINT16 RPDO	Indicate the configured time, during which the actual position within the position window is measured
607D h:01h	*Min_software_position_limit*	0.0001rev INT32 TPDO	Min position range limit
607D h:02h	*Max_software_position_limit*	0.0001rev INT32 TPDO	Max position range limit
607F h	*Max_profile_velocity*	0.1rpm UINT32 RPDO	Indicate the configured maximal allowed velocity in either direction
6081 h	*Profile_velocity*	0.1rpm UINT32 RPDO	Indicate the configured velocity attained at the end of the acceleration ramp. It is valid for both directions of motion
6083 h	*Profile_acceleration*	rpm/s UINT32 RPDO	Indicate the configured acceleration
6084 h	*Profile_deceleration*	rpm/s UINT32 RPDO	Indicate the configured deceleration
6085 h	*Quick_stop_deceleration*	rpm/s UINT32 RPDO	Indicate the configured deceleration used to stop the motor when the quick stop function is activated
60C1h :01h	*Interpolation_data_record*	0.0001rev INT32 TPDO	Set point (target position)
60C2h :01h	*Interpolation_time_period_value*	--- UINT8 RPDO	Cycle time, value.
60C2h :02h	*Interpolation_time_index*	--- INT8 RPDO	Cycle time, exponent

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

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

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

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

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

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

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

Statusword bits organization		
Bit		Description
15		Reserved, ignore the value
14		Reserved, ignore the value
13	fe	Following error, 1=following error
12	ipa	Ip mode active, 1=processed set-points 0=not processed set-points
11	ila	Internal limit active, 1=Restriction of one or more parameteres for internal limit
10	tr	Target reached, 1=Target position reached. In case of Halt or QuickStop, motor halted
9	rm	Remote, 1= *Controlword* executed
8	h	Halt, 1=Active request
7	w	Warning, 1=Presence of one or more warnings
6	sod	Switch on disabled
5	qs	Quick stop, 0=Quick Stop procedure in progress or concluded
4	ve	Voltage enabled, 1=Power supply applied to the device
3	f	Fault, 1=Error or Fault procedure in progress or concluded

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

The *Interpolated position* mode is selected writing the object 6060_h *Modes_of_operation* with the value 07_h.

The minimum steps required to control the motor in *Interpolated position* mode are the configuration of the maximum speed, the setting of the cycle time through the objects *Interpolation_time_period_value* and *Interpolation_time_index* and the finally enabling to process the set-points putting to 1 the bit *ei* contained in the object *Controlword*.

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

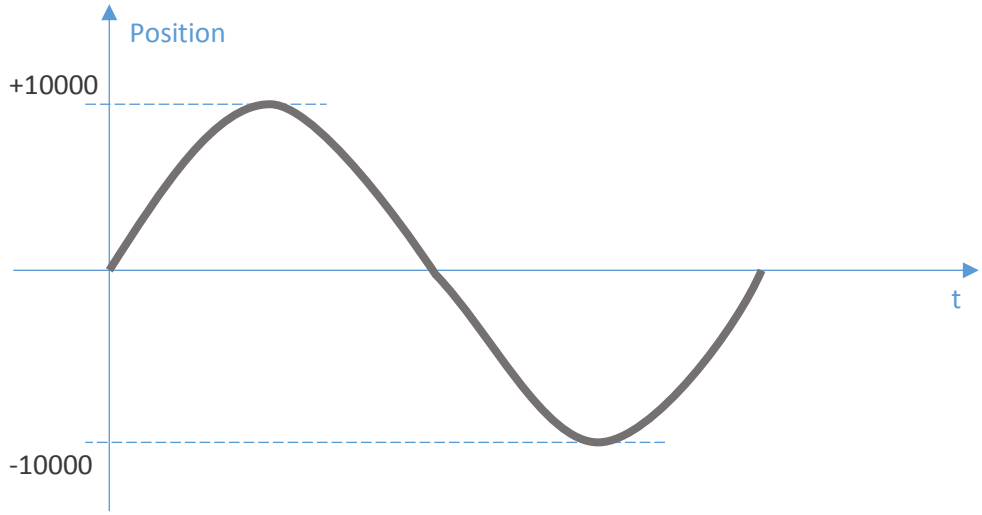
In the following example the SDO protocol is used to configure the drive and the PDO to enable and update the set-point. A second PDO is configured to transmit the *Statusword* and the actual position of the motor. Both the PDOs are configured in synchronous mode, with cycle time set to 1, that is for each SYNC object received.

Please take note that the cycle time can be set only when the interpolation is not active (bit *ipa* in the *Statusword* set to 0).

The drive used in the following example has the address $0D_h$ and assumes to have been started with the default values and successively configured with the minimum settings described in the previous chapters (Motor parameters and Running and idle current configuration). Furthermore, NMT is considered in the *Pre-Operational* state (default state after the power on) and the motor quote equal to 0.

The values in the time column refer to a bit rate of 250Kbit/s and can vary according to the traffic on the bus and to the reaction time of the master controller used, as well as to the firmware revision installed in the drive. The symbol → indicates a data flow from the bus to the drive while the symbol ← indicates a data flow from the drive to the CANopen bus.

The motion cycle of the example provides for the rotation of the motor of one clockwise revolution and one counterclockwise revolution with sinusoidal shape, as shown in the image below.



At the end of the cycle the motor will be in the same starting position.

The TPDO2 is used in the default configuration to transmit to the master controller the object *Statusword* and the actual position of the motor (object *Position_actual_value*). The RPDO2, instead, is re-configured to allow the setting of the *Controlword* and set-point (object *60C1h:01h Interpolation_data_record*).

→ ←	Time (ms)	COB-ID, Data	Description
→	0.0	60D, 23 00 14 01 0D 02 00 80	RPDO1 disabled
←	2.0	58D, 60 00 14 01 00 00 00 00	
→	6.9	60D, 23 00 18 01 8D 01 00 80	TPDO1 disabled
←	8.4	58D, 60 00 18 01 00 00 00 00	
→	12.2	60D, 23 01 14 01 0D 03 00 80	RPDO2 disabled
←	13.9	58D, 60 01 14 01 00 00 00 00	
→	18.2	60D, 2F 01 16 00 00 00 00 00	Set number of elements mapped to 0
←	20.0	58D, 60 01 16 00 00 00 00 00	
→	23.9	60D, 23 01 16 01 10 00 40 60	*Controlword* mapping in RPDO2
←	25.5	58D, 60 01 16 01 00 00 00 00	
→	29.8	60D, 23 01 16 02 20 01 C1 60	Object *Interpolation_data_record* mapping in RPDO2
←	31.5	58D, 60 01 16 02 00 00 00 00	
→	35.8	60D, 2F 01 16 00 02 00 00 00	Set number of elements mapped to 2
←	37.5	58D, 60 01 16 00 00 00 00 00	
→	41.6	60D, 23 01 14 01 0D 03 00 00	RPDO2 enabled
←	43.5	58D, 60 01 14 01 00 00 00 00	
→	47.3	60D, 2F 01 14 02 01 00 00 00	Set cyclic transmission each SYNC for RPDO2
←	48.9	58D, 60 01 14 02 00 00 00 00	
→	53.5	60D, 2F 01 18 02 01 00 00 00	Set cyclic transmission each SYNC for TPDO2
←	54.9	58D, 60 01 18 02 00 00 00 00	
→	59.6	60D, 23 01 18 01 8D 02 00 00	TPDO2 enabled
←	61.5	58D, 60 01 18 01 00 00 00 00	
→	65.5	60D, 2F 60 60 00 07 00 00 00	*Modes_of_operation* object set with 7 (7 = *Interpolated position*)
←	67.0	58D, 60 60 60 00 00 00 00 00	

→	71.8	60D, 40 61 60 00 00 00 00 00	Reading *Modes_of_operation_display* object to verify operative mode 7 active
←	73.4	58D, 4F 61 60 00 07 00 00 00	
→	78.0	60D, 2F C2 60 01 14 00 00 00	*Interpolation_time_period_value* object set with 20
←	79.4	58D, 60 C2 60 01 00 00 00 00	
→	84.4	60D, 2F C2 60 02 FD 00 00 00	*Interpolation_time_index* object set with -3
←	86.0	58D, 60 C2 60 02 00 00 00 00	
→	90.8	60D, 23 81 60 00 30 75 00 00	*Profile_velocity* object set with 30000 (3000rpm)
←	92.5	58D, 60 81 60 00 00 00 00 00	
→	97.8	000, 01 0D	Set NMT in *Start* state
	…	…	
→	3361.2	080,	SYNC object reception
←	3362.4	28D, 50 26 00 00 00 00	TPDO2, *Statusword* = 2650_h
→	3381.1	080,	SYNC object reception
←	3382.4	28D, 50 26 00 00 00 00	TPDO2, *Statusword* = 2650_h
→	3401.1	080,	SYNC object reception
←	3402.4	28D, 50 26 00 00 00 00	TPDO2, *Statusword* = 2650_h
→	3421.2	080,	SYNC object reception
→	3421.2	30D, 06 00 00 00 00 00	RPDO2, *Controlword* = 0006_h
←	3422.7	28D, 50 26 00 00 00 00	TPDO2, *Statusword* = 2650_h
→	3441.1	080,	SYNC object reception
→	3441.2	30D, 06 00 00 00 00 00	RPDO2, *Controlword* = 0006_h
←	3442.6	28D, 50 26 00 00 00 00	TPDO2, *Statusword* = 2650_h
→	3461.2	080,	SYNC object reception
→	3461.3	30D, 06 00 00 00 00 00	RPDO2, *Controlword* = 0006_h
←	3462.8	28D, 31 26 00 00 00 00	TPDO2, *Statusword* = 2631_h
→	3481.1	080,	SYNC object reception
→	3481.2	30D, 06 00 00 00 00 00	RPDO2, *Controlword* = 0006_h
←	3482.7	28D, 31 26 00 00 00 00	TPDO2, *Statusword* = 2650_h
	…	…	
→	4141.3	080,	SYNC object reception
→	4141.4	30D, 06 00 00 00 00 00	RPDO2, *Controlword* = 0006_h
←	4142.9	28D, 31 26 00 00 00 00	TPDO2, *Statusword* = 2631_h
→	4161.2	080,	SYNC object reception
→	4161.2	30D, 0F 00 00 00 00 00	RPDO2, *Controlword* = $000F_h$
←	4162.7	28D, 31 26 00 00 00 00	TPDO2, *Statusword* = 2631_h
→	4181.2	080,	SYNC object reception
→	4181.2	30D, 0F 00 00 00 00 00	RPDO2, *Controlword* = $000F_h$
←	4182.7	28D, 31 26 00 00 00 00	TPDO2, *Statusword* = 2631_h,
→	4201.2	080,	SYNC object reception
→	4201.3	30D, 0F 00 00 00 00 00	RPDO2, *Controlword* = $000F_h$
←	4202.7	28D, 37 06 00 00 00 00	TPDO2, *Statusword* = 0637_h
	…	…	
→	4965.3	080,	SYNC object reception
→	4965.3	30D, 0F 00 00 00 00 00	RPDO2, *Controlword* = $000F_h$
←	4966.8	28D, 37 06 00 00 00 00	TPDO2, *Statusword* = 0637_h
→	4985.2	080,	SYNC object reception
→	4985.2	30D, 1F 00 00 00 00 00	RPDO2, *Controlword* = $001F_h$
←	4986.7	28D, 37 06 00 00 00 00	TPDO2, *Statusword* = 0637_h
→	5005.3	080,	SYNC object reception

→	5005.3	30D, 1F 00 00 00 00 00	RPDO2, *Controlword* = $001F_h$
←	5006.8	28D, 37 06 00 00 00 00	TPDO2, *Statusword* = 0637_h
→	5025.4	080,	SYNC object reception
→	5025.4	30D, 1F 00 00 00 00 00	RPDO2, *Controlword* = $001F_h$
←	5026.9	28D, 37 16 00 00 00 00	TPDO2, *Statusword* = 1637_h
→	5045.2	080,	SYNC object reception
→	5045.3	30D, 1F 00 00 00 00 00	RPDO2, *Controlword* = $001F_h$
←	5046.7	28D, 37 16 00 00 00 00	TPDO2, *Statusword* = 1637_h
	…	…	
→	5865.2	080,	SYNC object reception
→	5865.3	30D, 1F 00 00 00 00 00	RPDO2, *Controlword* = $001F_h$, set-point 0
←	5866.7	28D, 37 16 00 00 00 00	TPDO2, *Statusword* = 1637_h , position 0
→	5885.3	080,	SYNC object reception
→	5885.3	30D, 1F 00 57 02 00 00	RPDO2, *Controlword* = $001F_h$, set-point 599
←	5886.4	28D, 37 16 00 00 00 00	TPDO2, *Statusword* = 1637_h , position 0
→	5905.3	080,	SYNC object reception
→	5905.4	30D, 1F 00 AD 04 00 00	RPDO2, *Controlword* = $001F_h$, set-point 1197
←	5906.9	28D, 37 16 00 00 00 00	TPDO2, *Statusword* = 1637_h , position 0
→	5925.3	080,	SYNC object reception
→	5925.3	30D, 1F 00 FE 06 00 00	RPDO2, *Controlword* = $001F_h$, set-point 1790
←	5926.8	28D, 37 12 3D 02 00 00	TPDO2, *Statusword* = 1237_h , position 573
→	5945.2	080,	SYNC object reception
→	5945.3	30D, 1F 00 49 09 00 00	RPDO2, *Controlword* = $001F_h$, set-point 2377
←	5946.7	28D, 37 12 80 04 00 00	TPDO2, *Statusword* = 1237_h , position 1152
→	5965.2	080,	SYNC object reception
→	5965.2	30D, 1F 00 8B 0B 00 00	RPDO2, *Controlword* = $001F_h$, set-point 2955
←	5966.7	28D, 37 12 D6 06 00 00	TPDO2, *Statusword* = 1237_h , position 1750
→	5985.2	080,	
→	5985.2	30D, 1F 00 C2 0D 00 00	It continues as described above
←	5986.7	28D, 37 12 23 09 00 00	
→	6005.3	080,	
→	6005.4	30D, 1F 00 ED 0F 00 00	
←	6006.8	28D, 37 12 77 0B 00 00	
→	6025.3	080,	
→	6025.4	30D, 1F 00 09 12 00 00	
←	6026.9	28D, 37 12 A9 0D 00 00	
→	6045.3	080,	
→	6045.4	30D, 1F 00 15 14 00 00	
←	6046.9	28D, 37 12 D5 0F 00 00	
→	6065.3	080,	
→	6065.3	30D, 1F 00 0E 16 00 00	
←	6066.4	28D, 37 12 E0 11 00 00	
→	6085.3	080,	
→	6085.3	30D, 1F 00 F3 17 00 00	
←	6086.3	28D, 37 12 F1 13 00 00	
→	6105.3	080,	
→	6105.3	30D, 1F 00 C1 19 00 00	
←	6106.9	28D, 37 12 FA 15 00 00	
→	6125.3	080,	
→	6125.4	30D, 1F 00 78 1B 00 00	

←	6126.9	28D, 37 12 DE 17 00 00	
→	6145.3	080,	
→	6145.4	30D, 1F 00 16 1D 00 00	
←	6146.8	28D, 37 12 AC 19 00 00	
→	6165.3	080,	
→	6165.4	30D, 1F 00 99 1E 00 00	
←	6166.8	28D, 37 12 64 1B 00 00	
→	6185.3	080,	
→	6185.4	30D, 1F 00 FF 1F 00 00	
←	6186.8	28D, 37 12 02 1D 00 00	
→	6205.4	080,	
→	6205.4	30D, 1F 00 49 21 00 00	
←	6206.9	28D, 37 12 87 1E 00 00	
→	6225,3	080,	
→	6225.4	30D, 1F 00 73 22 00 00	
←	6226.9	28D, 37 12 EE 1F 00 00	
→	6245.3	080,	
→	6245.3	30D, 1F 00 7E 23 00 00	
←	6246.4	28D, 37 12 2F 21 00 00	
→	6265.3	080,	
→	6265.3	30D, 1F 00 68 24 00 00	
←	6266.8	28D, 37 12 67 22 00 00	
→	6285.3	080,	
→	6285.4	30D, 1F 00 30 25 00 00	
←	6286.9	28D, 37 12 73 23 00 00	
→	6305.4	080,	
→	6305.6	30D, 1F 00 D7 25 00 00	
←	6307.0	28D, 37 12 5D 24 00 00	
→	6325.4	080,	
→	6325.5	30D, 1F 00 5A 26 00 00	
←	6326.9	28D, 37 12 24 25 00 00	
→	6345.3	080,	
→	6345.4	30D, 1F 00 BA 26 00 00	
←	6346.8	28D, 37 12 CF 25 00 00	
→	6365.4	080,	
→	6365.4	30D, 1F 00 F6 26 00 00	
←	6366.9	28D, 37 16 54 26 00 00	
→	6385.3	080,	
→	6385.4	30D, 1F 00 0F 27 00 00	
←	6386.9	28D, 37 16 B5 26 00 00	
→	6405.3	080,	
→	6405.3	30D, 1F 00 03 27 00 00	
←	6406.3	28D, 37 16 F1 26 00 00	
→	6425.4	080,	
→	6425.5	30D, 1F 00 D4 26 00 00	
←	6426.9	28D, 37 16 0E 27 00 00	
→	6445.3	080,	
→	6445.4	30D, 1F 00 81 26 00 00	
←	6446.8	28D, 37 16 04 27 00 00	
→	6465.3	080,	

→	6465.3	30D, 1F 00 0A 26 00 00	
←	6466.3	28D, 37 16 D7 26 00 00	
→	6485.3	080,	
→	6485.4	30D, 1F 00 70 25 00 00	
←	6486.8	28D, 37 16 84 26 00 00	
→	6505.4	080,	
→	6505.5	30D, 1F 00 B4 24 00 00	
←	6506.9	28D, 37 16 10 26 00 00	
→	6525.3	080,	
→	6525.4	30D, 1F 00 D6 23 00 00	
←	6526.8	28D, 37 12 78 25 00 00	
→	6545.3	080,	
→	6545.4	30D, 1F 00 D7 22 00 00	
←	6546.9	28D, 37 12 BD 24 00 00	
→	6565.3	080,	
→	6565.4	30D, 1F 00 B8 21 00 00	
←	6566.9	28D, 37 12 E0 23 00 00	
→	6585.3	080,	
→	6585.3	30D, 1F 00 79 20 00 00	
←	6586.8	28D, 37 12 E3 22 00 00	
→	6605.3	080,	
→	6605.4	30D, 1F 00 1D 1F 00 00	
←	6606.8	28D, 37 12 C5 21 00 00	
→	6625.3	080,	
→	6625.4	30D, 1F 00 A4 1D 00 00	
←	6626.9	28D, 37 12 87 20 00 00	
→	6645.4	080,	
→	6645.5	30D, 1F 00 10 1C 00 00	
←	6646.9	28D, 37 12 2D 1F 00 00	
→	6665.4	080,	
→	6665.4	30D, 1F 00 62 1A 00 00	
←	6666.9	28D, 37 12 B7 1D 00 00	
→	6685.3	080,	
→	6685.4	30D, 1F 00 9C 18 00 00	
←	6686.9	28D, 37 12 24 1C 00 00	
→	6705.3	080,	
→	6705.4	30D, 1F 00 BF 16 00 00	
←	6706.9	28D, 37 12 75 1A 00 00	
→	6725.3	080,	
→	6725.4	30D, 1F 00 CD 14 00 00	
←	6726.8	28D, 37 12 B1 18 00 00	
→	6745.4	080,	
→	6745.4	30D, 1F 00 C8 12 00 00	
←	6746.9	28D, 37 12 D5 16 00 00	
→	6765.3	080,	
→	6765.4	30D, 1F 00 B1 10 00 00	
←	6766.9	28D, 37 12 E5 14 00 00	
→	6785.3	080,	
→	6785.4	30D, 1F 00 8B 0E 00 00	
←	6786.9	28D, 37 12 E0 12 00 00	

→	6805.3	080,	
→	6805.4	30D, 1F 00 58 0C 00 00	
←	6806.8	28D, 37 12 CA 10 00 00	
→	6825.4	080,	
→	6825.4	30D, 1F 00 1A 0A 00 00	
←	6826.9	28D, 37 12 A5 0E 00 00	
→	6845.4	080,	
→	6845.4	30D, 1F 00 D2 07 00 00	
←	6846.9	28D, 37 12 74 0C 00 00	
→	6865.4	080,	
→	6865.4	30D, 1F 00 83 05 00 00	
←	6866.9	28D, 37 12 35 0A 00 00	
→	6885.4	080,	
→	6885.5	30D, 1F 00 2F 03 00 00	
←	6887.0	28D, 37 12 EE 07 00 00	
→	6905.4	080,	
→	6905.4	30D, 1F 00 D7 00 00 00	
←	6906.9	28D, 37 12 A2 05 00 00	
→	6925.3	080,	
→	6925.4	30D, 1F 00 81 FE FF FF	
←	6926.9	28D, 37 12 4C 03 00 00	
→	6945.3	080,	
→	6945.4	30D, 1F 00 2A FC FF FF	
←	6946.9	28D, 37 12 F3 00 00 00	
→	6965.4	080,	
→	6965.4	30D, 1F 00 D7 F9 FF FF	
←	6966.9	28D, 37 12 9E FE FF FF	
→	6985.3	080,	
→	6985.4	30D, 1F 00 8A F7 FF FF	
←	6986.8	28D, 37 12 46 FC FF FF	
→	7005.4	080,	
→	7005.4	30D, 1F 00 44 F5 FF FF	
←	7006.9	28D, 37 12 F2 F9 FF FF	
→	7025.4	080,	
→	7025.4	30D, 1F 00 09 F3 FF FF	
←	7026.9	28D, 37 12 A6 F7 FF FF	
→	7045.4	080,	
→	7045.4	30D, 1F 00 D9 F0 FF FF	
←	7046.9	28D, 37 12 60 F5 FF FF	
→	7065.3	080,	
→	7065.4	30D, 1F 00 B7 EE FF FF	
←	7066.8	28D, 37 12 25 F3 FF FF	
→	7085.4	080,	
→	7085.5	30D, 1F 00 A6 EC FF FF	
←	7086.9	28D, 37 12 F2 F0 FF FF	
→	7105.3	080,	
→	7105.4	30D, 1F 00 A6 EA FF FF	
←	7106.9	28D, 37 12 D3 EE FF FF	
→	7125.4	080,	
→	7125.5	30D, 1F 00 B9 E8 FF FF	

←	7127.0	28D, 37 12 BF EC FF FF	
→	7145.3	080,	
→	7145.4	30D, 1F 00 E3 E6 FF FF	
←	7146.9	28D, 37 12 BF EA FF FF	
→	7165.3	080,	
→	7165.4	30D, 1F 00 23 E5 FF FF	
←	7166.9	28D, 37 12 D0 E8 FF FF	
→	7185.4	080,	
→	7185.5	30D, 1F 00 7C E3 FF FF	
←	7186.9	28D, 37 12 F9 E6 FF FF	
→	7205.3	080,	
→	7205.4	30D, 1F 00 EF E1 FF FF	
←	7206.9	28D, 37 12 39 E5 FF FF	
→	7225.3	080,	
→	7225.4	30D, 1F 00 7E E0 FF FF	
←	7226.9	28D, 37 12 90 E3 FF FF	
→	7245.4	080,	
→	7245.5	30D, 1F 00 2A DF FF FF	
←	7246.9	28D, 37 16 EF E1 FF FF	
→	7265.3	080,	
→	7265.4	30D, 1F 00 F5 DD FF FF	
←	7266.9	28D, 37 12 93 E0 FF FF	
→	7285.3	080,	
→	7285.4	30D, 1F 00 DE DC FF FF	
←	7286.8	28D, 37 12 3B DF FF FF	
→	7305.3	080,	
→	7305.4	30D, 1F 00 E9 DB FF FF	
←	7306.8	28D, 37 12 03 DE FF FF	
→	7325.3	080,	
→	7325.4	30D, 1F 00 14 DB FF FF	
←	7326.9	28D, 37 12 EB DC FF FF	
→	7345.3	080,	
→	7345.4	30D, 1F 00 61 DA FF FF	
←	7346.8	28D, 37 12 F5 DB FF FF	
→	7365.3	080,	
→	7365.4	30D, 1F 00 D1 D9 FF FF	
←	7366.8	28D, 37 12 1E DB FF FF	
→	7385.4	080,	
→	7385.4	30D, 1F 00 64 D9 FF FF	
←	7386.9	28D, 37 12 69 DA FF FF	
→	7405.4	080,	
→	7405.4	30D, 1F 00 1B D9 FF FF	
←	7406.9	28D, 37 16 D9 D9 FF FF	
→	7425.4	080,	
→	7425.4	30D, 1F 00 F6 D8 FF FF	
←	7426.9	28D, 37 16 6A D9 FF FF	
→	7445.4	080,	
→	7445.6	30D, 1F 00 F4 D8 FF FF	
←	7446.9	28D, 37 16 1E D9 FF FF	
→	7465.4	080,	

→	7465.4	30D, 1F 00 17 D9 FF FF	
←	7466.9	28D, 37 16 F8 D8 FF FF	
→	7485.4	080,	
→	7485.4	30D, 1F 00 5D D9 FF FF	
←	7486.9	28D, 37 16 F4 D8 FF FF	
→	7505.3	080,	
→	7505.4	30D, 1F 00 C7 D9 FF FF	
←	7506.8	28D, 37 16 15 D9 FF FF	
→	7525.4	080,	
→	7525.5	30D, 1F 00 54 DA FF FF	
←	7527.0	28D, 37 16 5A D9 FF FF	
→	7545.6	080,	
→	7545.7	30D, 1F 00 04 DB FF FF	
←	7547.1	28D, 37 16 C1 D9 FF FF	
→	7565.4	080,	
→	7565.4	30D, 1F 00 D6 DB FF FF	
←	7566.8	28D, 37 12 4C DA FF FF	
→	7585.4	080,	
→	7585.5	30D, 1F 00 CA DC FF FF	
←	7586.9	28D, 37 12 FB DA FF FF	
→	7605.5	080,	
→	7605.6	30D, 1F 00 DD DD FF FF	
←	7607.0	28D, 37 12 CB DB FF FF	
→	7625.4	080,	
→	7625.5	30D, 1F 00 11 DF FF FF	
←	7627.0	28D, 37 12 BD DC FF FF	
→	7645.5	080,	
→	7645.6	30D, 1F 00 62 E0 FF FF	
←	7647.0	28D, 37 12 CF DD FF FF	
→	7665.4	080,	
→	7665.4	30D, 1F 00 D1 E1 FF FF	
←	7666.9	28D, 37 12 FF DE FF FF	
→	7685.4	080,	
→	7685.5	30D, 1F 00 5B E3 FF FF	
←	7686.9	28D, 37 12 51 E0 FF FF	
→	7705.4	080,	
→	7705.5	30D, 1F 00 00 E5 FF FF	
←	7707.0	28D, 37 12 BF E1 FF FF	
→	7725.4	080,	
→	7725.4	30D, 1F 00 BE E6 FF FF	
←	7726.9	28D, 37 12 47 E3 FF FF	
→	7745.4	080,	
→	7745.5	30D, 1F 00 93 E8 FF FF	
←	7747.1	28D, 37 12 EB E4 FF FF	
→	7765.4	080,	
→	7765.5	30D, 1F 00 7E EA FF FF	
←	7767.0	28D, 37 12 A6 E6 FF FF	
→	7785.4	080,	
→	7785.4	30D, 1F 00 7C EC FF FF	
←	7787.0	28D, 37 12 7B E8 FF FF	

→	7805.4	080,	
→	7805.4	30D, 1F 00 8C EE FF FF	
←	7807.0	28D, 37 12 65 EA FF FF	
→	7825.5	080,	
→	7825.6	30D, 1F 00 AD F0 FF FF	
←	7827.0	28D, 37 12 62 EC FF FF	
→	7845.5	080,	
→	7845.6	30D, 1F 00 DC F2 FF FF	
←	7847.0	28D, 37 12 6F EE FF FF	
→	7865.4	080,	
→	7865.5	30D, 1F 00 16 F5 FF FF	
←	7866.9	28D, 37 12 8F F0 FF FF	
→	7885.4	080,	
→	7885,4	30D, 1F 00 5B F7 FF FF	
←	7886.9	28D, 37 12 C0 F2 FF FF	
→	7905.4	080,	
→	7905.5	30D, 1F 00 A8 F9 FF FF	
←	7906.9	28D, 37 12 FB F4 FF FF	
→	7925.5	080,	
→	7925.6	30D, 1F 00 FA FB FF FF	
←	7927.1	28D, 37 12 3E F7 FF FF	
→	7945.4	080,	
→	7945.5	30D, 1F 00 51 FE FF FF	
←	7947.0	28D, 37 12 86 F9 FF FF	
→	7965.4	080,	
→	7965.5	30D, 1F 00 A8 00 00 00	
←	7966.9	28D, 37 12 DA FB FF FF	
→	7985.3	080,	
→	7985.4	30D, 1F 00 FF 02 00 00	
←	7986.9	28D, 37 12 31 FE FF FF	
→	8005.4	080,	
→	8005.4	30D, 1F 00 53 05 00 00	
←	8006.9	28D, 37 12 8B 00 00 00	
	…	…	

Please note that the bit *Target-reached* bit becomes active before reaching the commanded position because of the object *Position_window* that by default is set to 10. In this way the motor is considered "in position" (bit *Target-reached* = 1) each time the difference between the actual position and the commanded one is less than 10 in absolute value. If you want to have the bit *Target-reached* active at the reaching of the exact target position, it is sufficient to set the object *Position_window* equal to 0.

5 Errors and diagnostics

The drive is able to detect many 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. According to the profile /CiA301 / the error code appears in the field *Emergency error code* of the object *Emergency object (EMCY),* while the sub-code in the field *Manufacturer-specific error code* of the same object.

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:

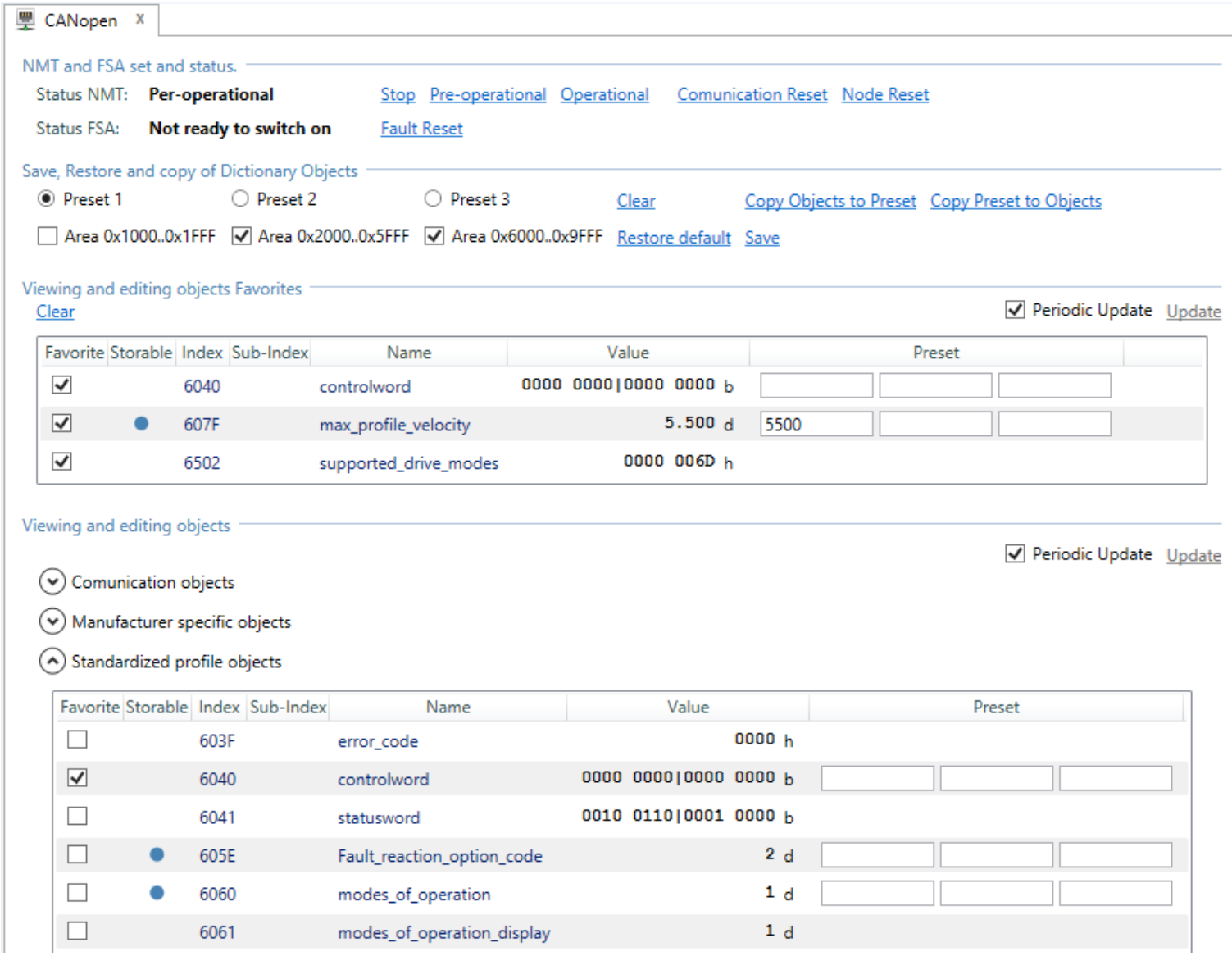
<table>
<tr><th>Error code</th><th>Class</th><th>LED Flash</th><th>bit</th><th>Description / Sub-code</th></tr>
<tr><td>2310_h</td><td>6</td><td>4</td><td>3</td><td>Detected over current motor outputs side
<table>
<tr><th>Sub-code</th><th>Description</th></tr>
<tr><td>0100_h</td><td>During PWM modulation</td></tr>
<tr><td>0101_h</td><td>Towards ground</td></tr>
<tr><td>0102_h</td><td>Towards positive power bus</td></tr>
<tr><td>0103_h</td><td>Short circuit between phase A+ and ground</td></tr>
<tr><td>0104_h</td><td>Short circuit between phase A- and ground</td></tr>
<tr><td>0105_h</td><td>Short circuit between phase B+ and ground</td></tr>
<tr><td>0106_h</td><td>Short circuit between Phase B- and ground</td></tr>
<tr><td>0109_h</td><td>Abnormal current between phase A+ and ground</td></tr>
<tr><td>$010A_h$</td><td>Abnormal current between phase A- and ground</td></tr>
<tr><td>$010B_h$</td><td>Abnormal current between phase B+ and ground</td></tr>
<tr><td>$010C_h$</td><td>Abnormal current between phase B- and ground</td></tr>
</table></td></tr>
<tr><td>$23A0_h$</td><td>6</td><td>5</td><td>5</td><td>Detected phase open (not connected or interrupted) motor outputs side
<table>
<tr><th>Sub-code</th><th>Description</th></tr>
<tr><td>0201_h</td><td>Phase A</td></tr>
<tr><td>0202_h</td><td>Phase B</td></tr>
</table></td></tr>
<tr><td>$23B2_h$</td><td>6</td><td>8</td><td>4</td><td>Detected over current on the encoder power supply +V</td></tr>
<tr><td>3210_h</td><td>6</td><td>2</td><td>1</td><td>Power supply voltage higher than the maximum allowed</td></tr>
<tr><td>3220_h</td><td>6</td><td>1</td><td>0</td><td>Power supply voltage lower than the minimum necessary for correct functioning</td></tr>
<tr><td>4210_h</td><td>4</td><td>3</td><td>2</td><td>Power stage temperature higher than the maximum allowed value</td></tr>
<tr><td>5592_h</td><td>8</td><td>13</td><td>24</td><td>Invalid device descriptor</td></tr>
<tr><td>5594_h</td><td>8</td><td>13</td><td>24</td><td>Invalid device configuration</td></tr>
<tr><td>$55A0_h$</td><td>8</td><td>13</td><td>31</td><td>Inappropriate device installed firmware</td></tr>
<tr><td>$55A2_h$</td><td>8</td><td>13</td><td>31</td><td>Incompatible device firmware revision</td></tr>
<tr><td>6200_h</td><td>8</td><td>15</td><td>31</td><td>Firmware error</td></tr>
<tr><td>$63A0_h$</td><td>8</td><td>10</td><td>16</td><td>Conflicts in the configuration
<table>
<tr><th>Sub-code</th><th>Description</th></tr>
<tr><td>0701_h</td><td>Selected Torque mode without encode feedback (present or configured).</td></tr>
<tr><td>0702_h</td><td>Selected current proportional to the load (object Mode_CRRG) without encoder feedback (present o configured).</td></tr>
<tr><td>0703_h</td><td>Set interpolation time (objects …and….) out of allowed limits.</td></tr>
<tr><td>0704_h</td><td>Activated encoder feedback without having activated or configured the encoder</td></tr>
</table></td></tr>
<tr><td>7282_h</td><td>8</td><td>14</td><td>11</td><td>Detected invalid offset values</td></tr>
<tr><td>8110_h</td><td>2</td><td>7</td><td>28</td><td>CAN overrun (lost messages)</td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td></td><td><table><tr><th>Sub-code</th><th>Description</th></tr><tr><td>0501_h</td><td>In receipt</td></tr><tr><td>0502_h</td><td>In transmission</td></tr></table></td></tr>
<tr><td>8120_h</td><td>2</td><td>7</td><td>28</td><td>CAN in passive mode</td></tr>
<tr><td>8130_h</td><td>2</td><td>7</td><td>22</td><td>Error Life Guard or Heartbeat</td></tr>
<tr><td>8140_h</td><td>2</td><td>7</td><td>28</td><td>Bus off condition recovery</td></tr>
<tr><td>8200_h</td><td>2</td><td>7</td><td>28</td><td>CANopen protocol error</td></tr>
<tr><td>8210_h</td><td>2</td><td>7</td><td>22</td><td>Unprocessed PDO because of incorrect length</td></tr>
<tr><td>8220_h</td><td>2</td><td>7</td><td>22</td><td>PDO length greater than expected</td></tr>
<tr><td>8240_h</td><td>2</td><td>7</td><td>22</td><td>SYNC unexpected length</td></tr>
<tr><td>8250_h</td><td>2</td><td>7</td><td>22</td><td>RPDO time out</td></tr>
<tr><td>$82A1_h$</td><td>2</td><td>7</td><td>22</td><td>RPDO1 processing error</td></tr>
<tr><td>$82A2_h$</td><td>2</td><td>7</td><td>22</td><td>RPDO2 processing error</td></tr>
<tr><td>$82A3_h$</td><td>2</td><td>7</td><td>22</td><td>RPDO3 processing error</td></tr>
<tr><td>$82A4_h$</td><td>2</td><td>7</td><td>22</td><td>RPDO4 processing error</td></tr>
<tr><td>$82B2_h$</td><td>2</td><td>7</td><td>22</td><td>SYNC period breach</td></tr>
<tr><td>$82C2_h$</td><td>4</td><td>7</td><td>22</td><td>Error in the execution of the requested NMT service
<table><tr><th>Sub-Index</th><th>Description</th></tr><tr><td>0901_h</td><td>NMT Stop with motor enabled</td></tr><tr><td>0902_h</td><td>NMT Reset Comunication with motor enabled</td></tr><tr><td>0903_h</td><td>NMT Node Reset with motor enabled</td></tr></table></td></tr>
<tr><td>8611_h</td><td>2</td><td>6</td><td>18</td><td>Following error. The indicated class expresses the default value. The class can be modified through the object Following_Error_ERRCS.</td></tr>
<tr><td>8612_h</td><td>2</td><td>6</td><td>19</td><td>Limit switch activation
<table><tr><th>Sub-code</th><th>Description</th></tr><tr><td>0801_h</td><td>Positive limit activation</td></tr><tr><td>0802_h</td><td>Negative limit activation</td></tr><tr><td>0803_h</td><td>Positive movement with active positive limit</td></tr><tr><td>0804_h</td><td>Negative movement with active negative limit</td></tr></table></td></tr>
<tr><td>8613_h</td><td>2</td><td>6</td><td>17</td><td>Homing error
<table><tr><th>Sub-code</th><th>Description</th></tr><tr><td>0601_h</td><td>Unfound index</td></tr><tr><td>0602_h</td><td>Unexpected clockwise limit</td></tr><tr><td>0603_h</td><td>Unexpected counterclockwise limit</td></tr></table></td></tr>
</table>

6 CANopen

The DDS6 series drives implement the CANopen protocol standardized according to the *Communication Profile DS301 Version: 4.2.0* and the *Drives and Motion Control Device Profile DSP402 Version: 4.0.0,* as well as officially documented by the CiA (CAN in Automation). For a detailed description of the protocol and profiles, please refer to the documents available on the official website www.can-cia.org.

6.1 Visualization and modification of registers



Scrolling the tab from up to bottom the first section met shows the status of the NMT (Network management) and FSA (Finite State Automaton). The links on the right allow you to force the NMT status and execute the Fault reset.

The manual manipulation of the NMT or FSA status occurs simultaneously with the bus and this can lead to unforeseen conditions as the master controller may not provide for a change of status operated manually from the outside. Furthermore, the NMT or FSA status change can cause unexpected motor behavior and therefore represents a potential source of danger..

Do not manipulate the NMT or FSA status in the even that any of motor behaviour can cause damage to property or persons.

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

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

To transfer the Preset value into the corresponding object, simply press the Enter button or double-click with the left mouse button.

The Preset can be compiled with an hexadecimal value, putting the prefix “0x”, or binary, putting the prefix “0b”. For clarity, it is also possible to intersperse groups of figures with spaces. For example. it is possible to write the value 181 in binary as “0b 1011 0101”.

Totally, there are 3 Preset for each register that allow you to quickly stich from a value to another by simply double-clicking on the wished Preset value.

The 3 following boxes allow to select the objects range on which to operate. It is possible to check more than one box to select a wider range of objects.

The links on the side allow you to save the dictionary objects in the non-volatile memory or to restore the default values. Both operations are possible only with the motor disabled and in the NMT status *Stopped* or *Pre-operational*. For more details see chapter 4.2 Saving and restoring of default values.

The dictionary objects visualization and modification area is logically divided in two zones; one that resumes only the favorited objects and a second zone that instead contains all the dictionary objects.

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

The objects contained in the favorited 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 object can be saved in the non-volatile memory using the *Store Parameters* functions. If there is a blue point it means that the object will be stored in the non-volatile memory as a result of the *Store Parameters* command.

The following columns show the Index and Sub-Indey of the object in hexadecimal format and the name.

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

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

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

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

Symbol	Base
d	Decimal
h	Hexadecimal
b	Binary

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

Please note that some objects 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.

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

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

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

By clicking with the right mouse button on the CANopen item (on the left in the devices tree) it is possible to save and load the board configuration with the obejcts included in the preferred area, the value visualization base, etc.

6.2 Objects Dictionary

The objects dictionary accessible through the CANopen protocol can be logically divided into three areas. The objects with index between 1000_h and $1FFF_h$ are standardized by the profile DS301 and are mainly related to the communication; the objects with index between 2000_h and $5FFF_h$ are specific of LAM Technologies products and described in this manual; finally the objects between 6000_h and $9FFF_h$ are standardized by the profile DSP402.

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

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

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

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

Their function, the access mode, etc. are described in the manuals of the /CiA301/ standards available on the official site www.can-cia.org, therefore in this manual there will be no detailed description of each object but a synthetic summary of the implemented ones.

6.3.1 1000_h Device type

Index	Name			Mnemonic
1000_h	*Device_type*			
	Data Type: u32	Access Type: ro	PDO Mapping	Note
	Default Value: 0x00440192	Minimum	Maximum	Unit

Description

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

bits 31..16	bits 15..0
Additional information	Device profile number

6.3.2 1001_h Error register

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

Description

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

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

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

6.3.3 1003_h Pre-defined error field

Index	Name	Type
1003_h	*Pre-defined error field*	RECORD

Sub-Index	Name			Mnemonic
00_h	*Number_of_Errors*			
	Data Type: u8	Access Type: rw	PDO Mapping	Note
	Default Value: 00_h	Minimum	Maximum	Unit
	Description: Number of record errors. Writing the object with the value 0h, the error history is deleted. No other value is allowed.			

Sub-Index	Name			Mnemonic
01_h...$0a_h$	*Pre-defined_error_field_1…10*			
	Data Type: u32	Access Type: rw	PDO Mapping	Note
	Default Value: 0	Minimum	Maximum	Unit

Description

It allows to read the chronological sequence of errors.

The object *Pre-defined_error_field_1* contains the most recent error while the object *Pre-defined_error_field _10* contains the last error in chronological order.

It is possible to read only the existing objects. The number of the existing objects is contained in the object *Number_of_Errors*.

The bits of the object are used as follows:

bits 31..16	bits 15..0
Additional information	Error Code

The error codes are listed in chapter 5 Errors and diagnostics.

6.3.4 1005_h COB-ID SYNC message

Index	Name			Mnemonic
1005_h	*COB-ID_SYNC_message*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw		🖫
	Default Value	Minimum	Maximum	Unit
	0x00000080			
	Description			
	This object indicates the configured COB-ID of the synchronization object (SYNC).			

6.3.5 1008_h Manufacturer device name

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

6.3.6 1009_h Manufacturer hardware version

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

6.3.7 $100A_h$ Manufacturer software version

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

6.3.8 $100C_h$ Guard time

Index	Name			Mnemonic
$100C_h$	*Guard_time*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		🖫
	Default Value	Minimum	Maximum	Unit
	0x0000			ms (Ex. 150 = 150ms)

	Description The objects at index $100C_h$ and $100D_h$ indicate the configured guard time respectively the life time factor. The life time factor multiplied with the guard time gives the life time for the life guarding protocol. The value is given in ms. The value of 0000h disable the life guarding.

6.3.9 $100D_h$ Life time factor

Index	Name			Mnemonic
$100D_h$	*Life_time_factor*			
	Data Type u8	Access Type rw	PDO Mapping	Note 🖫
	Default Value 0x00	Minimum	Maximum	Unit
	Description The life time factor multiplied with the guard time gives the life time for the life guarding protocol.			

6.3.10 1010_h Store Parameters

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

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

The defaults values can be saved or restored only with the motor disabled or with the NMT status *Stopped* or *Pre-operational*. Trying the operation with the motor enabled or in the NMT *Operational* status an error code answer is received.

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

Index	Name	Type
1010_h	*Store Parameters*	ARRAY

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

Sub-Index	Name			Mnemonic
01_h	*Save all Parameters*			
	Data Type u32	Access Type rw	PDO Mapping	Note
	Default Value	Minimum	Maximum	Unit
	Description			

It allows to save the dictionary objects with index between 1000_h and $9FFF_h$, specified with the symbol 🖫, in the non-volatile memory.

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

Sub-Index	Name			Mnemonic
02_h	*Save Communication Parameters*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw		
	Default Value	Minimum	Maximum	Unit
	Description			
	It allows to save the dictionary objects with index between 1000_h and $1FFF_h$, specified with the symbol 🖫, in the non-volatile memory. To start and save simply write the object with the value 0x65766173.			

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

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

6.3.11 1011_h Restore Default Parameters

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

The default values can be restored only with the motor disabled or with the NMT status *Stopped* or *Pre-operational*. Trying the operation with motor enabled or with the NMT status *Operational*, an error code is obtained in reply.

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

Index	Name	Type
1011_h	*Restore Default Parameters*	ARRAY

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

Sub-Index	Name			Mnemonic
01_h	*Restore all Default Parameters*			
	Data Type: u32	Access Type: rw	PDO Mapping	Note
	Default Value	Minimum	Maximum	Unit
	Description: It allows to save the dictionary objects with index between 1000_h and $9FFF_h$. To start and save simply write the object with the value 0x64616F6C.			

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

Sub-Index	Name			Mnemonic
03_h	*Restore Application Default Parameters*			
	Data Type: u32	Access Type: rw	PDO Mapping	Note
	Default Value	Minimum	Maximum	Unit
	Description			

	It allows to save the dictionary objects with index between 6000_h and $9FFF_h$. To start and save simply write the object with the value 0x64616F6C.

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

6.3.12 1014_h COB-ID EMCY message

Index	Name			Mnemonic
1014_h	*COB-ID_EMCY_message*			
	Data Type u32	Access Type rw	PDO Mapping	Note 🖫
	Default Value Node-ID + 80_h	Minimum	Maximum	Unit
	Description This object indicates the configured COB-ID for the EMCY write service.			

6.3.13 1015_h Inhibit time EMCY

Index	Name			Mnemonic
1015_h	*Inhibit_time_EMCY*			
	Data Type u16	Access Type rw	PDO Mapping	Note 🖫
	Default Value 0x0000	Minimum	Maximum	Unit 0.1 ms (Ex. 2500 = 250ms)
	Description This object indicates the configured inhibit time for the EMCY message. The value is given in 0.1ms. The value of 0000h disables the inhibit time.			

6.3.14 1016_h Consumer heartbeat time

Index	Name	Type
1016_h	*Consumer heartbeat time*	RECORD

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

Sub-Index	Name			Mnemonic
01_h...04_h	*Consumer_heartbeat_time_1…4*			
	Data Type: u32	Access Type: rw	PDO Mapping	Note: 💾
	Default Value: 0	Minimum	Maximum	Unit

Description:

It allows to set the time and address of the node to be monitored.

The objects bits are used as follows:

bits 31..24	bits 23..16	bits 15..0
Reserved, set to 0	Node address (Node-ID)	Time (Heartbeat time)

The time is expressed in ms (ex. 1500 is equal to 1,5s) and if set to 0 monitoring is interrupted.

6.3.15 1017_h Producer heartbeat time

Index	Name			Mnemonic
1017_h	*Producer_heartbeat_time*			
	Data Type: u16	Access Type: rw	PDO Mapping	Note: 💾
	Default Value: 0x0000	Minimum	Maximum	Unit: ms (Ex. 100 = 100ms)

Description:

The producer heartbeat time indicates the configured cycle time of the heartbeat.

The value is given in ms. The value of 0000h disables the producer heartbeat.

6.3.16 1018_h Identity object

This object provide general identification information of the CANopen device.

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

Sub-index 02h contains the unique value that identifies a specific device.

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

Sub-index 04h contains the device serial number.

Index	Name	Type
1018_h	*Identity object*	RECORD

Sub-Index	Name			Mnemonic
00_h	*Number_of_Entries*			
	Data Type: u8	Access Type: const	PDO Mapping	Note
	Default Value: 04_h	Minimum	Maximum	Unit
	Description: Highest sub-index supported.			

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

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

Description

Contains the unique value that identifies a specific device.

The following table shows the correspondence between code and device:

Code	Device	Code	Device
00604100_h	DDS6041	00604101_h	DDS6041A
00604400_h	DDS6044	00604401_h	DDS6044A
00604800_h	DDS6048	00604801_h	DDS6048A
00607400_h	DDS6074	00607401_h	DDS6074A
00607800_h	DDS6078	00607801_h	DDS6078A
00624100_h	DDS6241	00624101_h	DDS6241A
00624400_h	DDS6244	00624401_h	DDS6244A

00624800_h	DDS6248	00624801_h	DDS6248A
00627400_h	DDS6274	00627401_h	DDS6274A
00627800_h	DDS6278	00627801_h	DDS6278A

Sub-Index	Name			Mnemonic
03_h	*Revision_number*			
	Data Type: u32	Access Type: ro	PDO Mapping	Note
	Default Value	Minimum	Maximum	Unit
	Description: Contains the major revision number and the minor revision number of the revision of the device. The major revision number identifies a specific CANopen behavior. That means if the CANopen functionality is different, the major revision number is different. The minor revision number identifies different versions of device with the same CANopen behavior.			

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

Sub-Index	Name			Mnemonic
04_h	*Serial_number*			
	Data Type: u32	Access Type: ro	PDO Mapping	Note
	Default Value	Minimum	Maximum	Unit
	Description: Contains the device serial number.			

6.3.17 1029_h Error behavior

Index	Name	Type
1029_h	*Error behavior*	RECORD

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

Sub-Index	Name			Mnemonic
01_h	*Communication_error*			
	Data Type: u8	Access Type: rw	PDO Mapping	Note
	Default Value: 0x00	Minimum	Maximum	Unit
	Description: If a serious CANopen device failure is detected in NMT state Operational, the device changes autonomously the NMT state according to this object value. The table below shows the values definition:			

0	Change to NMT state Pre-operational (only if currently in NMT state Operational)
1	No change of the NMT state
2	Change to NMT state Stopped

6.3.18 1200_h SDO server parameter

Index	Name	Type
1200_h	*SDO server parameter*	RECORD

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

Sub-Index	Name			Mnemonic
01_h	*COB-ID_client_to_server (rx)*			
	Data Type: u32	Access Type: const	PDO Mapping	Note
	Default Value: Node-ID+0x600	Minimum	Maximum	Unit
	Description: Contains the COB-ID for the transmission from client to server.			

Sub-Index	Name			Mnemonic
02_h	*COB-ID_server_to_client (tx)*			
	Data Type: u32	Access Type: ro	PDO Mapping	Note
	Default Value: Node-ID+0x580	Minimum	Maximum	Unit
	Description: Contains the COB-ID for the transmission from server to client.			

6.3.19 1400_h, 1401_h, 1402_h, 1403_h RPDO communication parameter

Index	Name	Type
1400_h 1401_h 1402_h 1403_h	*RPDO communication parameter*	RECORD

Sub-Index	Name			Mnemonic
00_h	*Number_of_Entries*			
	Data Type	Access Type	PDO Mapping	Note
	u8	const		
	Default Value	Minimum	Maximum	Unit
	05_h 05_h 05_h 05_h			
	Description			
	Highest sub-index supported.			

Sub-Index	Name			Mnemonic
01_h	*COB-ID_used_by_RPDO*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw		🖫
	Default Value	Minimum	Maximum	Unit
	Node-ID + 0x00000200 Node-ID + 0x80000300 Node-ID + 0x80000400 Node-ID + 0x80000500			
	Description			
	Contains the COB-ID of the RPDO.			

31	30	29	Bits 28…11	Bits 10…0
valid	reserved	frame	00000_h	11-bit CAN-ID
			29-bit CAN-ID	

Sub-Index	Name			Mnemonic
02_h	*Transmission_type*			
	Data Type	Access Type	PDO Mapping	Note
	u8	rw		🖫
	Default Value	Minimum	Maximum	Unit
	FF_h FF_h FF_h FF_h			
	Description			
	Defines the reception behavior of the RPDO. The table below shows the values definition:			

Value	Description
00_h … $F0_h$	Synchronous
FF_h	Event-driven

Sub-Index	Name			Mnemonic
03_h	*Inhibit_time*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		🖫
	Default Value	Minimum	Maximum	Unit
	0 0 0 0			
	Description			
	Contains the inhibit time. The value is given in 0.1 ms. The value of 0000h disables the inhibit time.			

Sub-Index	Name			Mnemonic
05_h	*Event-timer*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		🖫
	Default Value	Minimum	Maximum	Unit
	0 0 0 0			ms (Ex. 50 = 50ms)
	Description			
	Contains the event-timer. The value is given in ms. The value of 0000h disables the event-timer.			

6.3.20 1600_h, 1601_h, 1602_h, 1603_h RPDO mapping parameter

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

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

Sub-Index	Name			Mnemonic
01_h	*Application_object_1*			
	Data Type u32	Access Type rw	PDO Mapping	Note 💾
	Default Value 0x60400010 0x60400010 0x60400010 0x607A0020	Minimum	Maximum	Unit

Description

Contains the information of the mapped application object 1.

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

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

Sub-Index	Name			Mnemonic
02_h	*Application_object_2*			
	Data Type u32	Access Type rw	PDO Mapping	Note 💾
	Default Value --- 0x607A0020 0x60FF0020 0x60810020	Minimum	Maximum	Unit

Description

Contains the information of the mapped application object 2.

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

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

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

Description

Contains the information of the mapped application objects.

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

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

6.3.21 1800_h, 1801_h, 1802_h, 1803_h TPDO communication parameter

Index	Name	Type
1800_h 1801_h 1802_h 1803_h	*TPDO communication parameter*	RECORD

Sub-Index	Name			Mnemonic
00_h	*Number_of_Entries*			
	Data Type	Access Type	PDO Mapping	Note
	u8	const		
	Default Value	Minimum	Maximum	Unit
	06_h 06_h 06_h 06_h			
	Description			
	Highest sub-index supported.			

Sub-Index	Name			Mnemonic
01_h	*COB-ID_used_by_TPDO*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw		🖫
	Default Value	Minimum	Maximum	Unit
	Node-ID + 0x00000180 Node-ID + 0x80000280 Node-ID + 0x80000380 Node-ID + 0x80000480			
	Description			

Contains the COB-ID of the TPDO.

31	30	29	Bits 28…11	Bits 10…0
valid	RTR	frame	00000_h	11-bit CAN-ID
			29-bit CAN-ID	

Sub-Index	Name			Mnemonic
02_h	*Transmission_type*			
	Data Type	Access Type	PDO Mapping	Note
	u8	rw		🖫
	Default Value	Minimum	Maximum	Unit
	FF_h FF_h FF_h FF_h			
	Description			

Defines the transmission behavior of the TPDO.

The table below shows the values definition:

Value	Description
00_h	Synchronous acyclic
01_h … $F0_h$	Synchronous every n sync (n=value)
FC_h	RTR-only synchronous
FD_h	RTR-only event-driven
FF_h	Triggered by *Statusword* change

Sub-Index	Name			Mnemonic
03_h	*Inhibit_time*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		
	Default Value	Minimum	Maximum	Unit
	0 0 0 0			0.1 ms (Ex. 3000 = 300ms)
	Description			
	Contains the inhibit time. The time is the minimum interval for PDO transmission if the transmission type is set to FE_h or FF_h. The value is given in 0.1 ms. The value of 0000h disables the inhibit time.			

Sub-Index	Name			Mnemonic
05_h	*Event-timer*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		
	Default Value	Minimum	Maximum	Unit
	0 100 100 0			ms (Ex. 50 = 50ms)
	Description			
	Contains the event-timer. The time is the maximum interval for PDO transmission if the transmission type is set to FE_h and FF_h. The value is given in ms. The value of 0000h disables the event-timer.			

Sub-Index	Name			Mnemonic
06_h	*SYNC_start_value*			
	Data Type	Access Type	PDO Mapping	Note
	u8	rw		
	Default Value	Minimum	Maximum	Unit
	0 0 0 0			
	Description			
	Contains the SYNC start value. The SYNC start value of 0 indicates that the counter of the SYNC message shall not be processed for this PDO. The SYNC start value 1 to 240 indicates that the counter of the SYNC message shall be processed for this PDO. The SYNC message of which the counter value equals the SYNC start value is used as first			

	received SYNC message.

6.3.22 1A00$_h$, 1A01$_h$, 1A02$_h$, 1A03$_h$ TPDO mapping parameter

Index	Name	Type
1A00$_h$ 1A01$_h$ 1A02$_h$ 1A03$_h$	*TPDO mapping parameter*	RECORD

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

Sub-Index	Name			Mnemonic
01$_h$	*Mapped_application_object_1*			
	Data Type u32	Access Type rw	PDO Mapping	Note 💾
	Default Value 0x60410010 0x60410010 0x60410010 ---	Minimum	Maximum	Unit
	Description Contains the information of the mapped application object 1. The object describes the content of the PDO by their index, sub-index and length (length of the application object in bit).			

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

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

Contains the information of the mapped application object 2.

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

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

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

Description

Contains the information of the mapped application objects.

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

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

6.4 Object with index between 2000_h and $5FFF_h$

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

6.4.1 2310_h Motor Data

Index	Name	Type
2310_h	*Motor Data*	RECORD

Sub-Index	Name			Mnemonic
00_h	*Number_of_Entries*			
	Data Type: u8	Access Type: const	PDO Mapping:	Note:
	Default Value: $0B_h$	Minimum:	Maximum:	Unit:
	Description: Number of record entries.			

Sub-Index	Name			Mnemonic
01_h	*CMC_MTRDT*			
	Data Type: u32	Access Type: rw	PDO Mapping:	Note: 🖫
	Default Value: 0	Minimum:	Maximum:	Unit:
	Description: It is a compact code that identifies the LAM Technologies motors and that allows the drive to configure the best control parameter for the motor. It is sufficient to initialize the object *CMC Motor Data* with the CMC code of the motor connected to the drive to instruct the drive about all the motor features. When the object is initialized with a value different from 0, the objects with Sub-Index from 02_h to $0A_h$ are ignored.			

Sub-Index	Name			Mnemonic
02_h	*Type_MTRDT*			
	Data Type: u8	Access Type: ro	PDO Mapping:	Note: 🖫 Valid only if *CMC_MTRDT* = 0
	Default Value: 12	Minimum:	Maximum:	Unit:
	Description: It indicates the type of motor connected to the drive according to the table below:			

12	Two-phase Stepper Motor

Sub-Index	Name			Mnemonic
03_h	*Pole_Pairs_MTRDT*			
	Data Type: u8	Access Type: rw	PDO Mapping:	Note: 🖫 Valid only if *CMC_MTRDT* = 0
	Default Value: 50	Minimum:	Maximum:	Unit:
	Description: It allows to set the poles number of the motor. The number of pole is the number of electrical cycles included in a complete motor revolution. For example a 200 steps/rev two-phase stepper motor has a number of poles equal to 50.			

Sub-Index	Name			Mnemonic
04h	*Wiring_MTRDT*			
	Data Type	Access Type	PDO Mapping	Note
	u8	ro		Valid only if *CMC_MTRDT* = 0
	Default Value	Minimum	Maximum	Unit
	0			
	Description			
	Motor connection. Currently not used.			

Sub-Index	Name			Mnemonic
05h	*Resistance_MTRDT*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		Valid only if *CMC_MTRDT* = 0
	Default Value	Minimum	Maximum	Unit
	100		60000	10mOhm (Ex. 240 = 2.4 Ohm)
	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.			

Sub-Index	Name			Mnemonic
06h	*Inductance_MTRDT*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		Valid only if *CMC_MTRDT* = 0
	Default Value	Minimum	Maximum	Unit
	300		60000	10µH (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 an optimal functioning it is very important to carefully set this object so that it indicates the real inductance of the connected motor.			

Sub-Index	Name			Mnemonic
07h	*Back_EMF_MTRDT*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		Valid only if *CMC_MTRDT* = 0
	Default Value	Minimum	Maximum	Unit
	2500		60000	10mV/1000rpm (Ex. 4500 = 45V/1000rpm)
	Description			
	It allows to set the back EMF of the motor connected to the drive. The set value must take into account the type of phase connection (parallel or series), if the motor allows multiple configurations.			

Sub-Index	Name			Mnemonic
08h	*Rated_Current_MTRDT*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		Valid only if *CMC_MTRDT* = 0
	Default Value	Minimum	Maximum	Unit

	100	10	60000	10mArms (Ex. 420 = 4.2 Arms)x

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 an optimal functioning it is very important to carefully set this object so that it indicates the real value of the rated current at which the motor was built.

Do not use this object to modify the motor current. The object *Rated_Current_MTRDT* must always be set with the rated current value indicated by the manufacturer of the motor. To modify the running or idle current of the motor, use instead the objects *Current_Max_MTRCNF* and *Current_Min_MTRCNF*.

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

Sub-Index	Name			Mnemonic
09_h	*Max_Current_MTRDT*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		Valid only if *CMC_MTRDT* = 0
	Default Value	Minimum	Maximum	Unit
	130	10	60000	10mArms (Ex. 550 = 5.5 Arms)

Description

It allows to set the maximum current 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 a LAM Technologies motor, it can be set at a value equal to the 130% of the parameter *Rated_Current_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.

Sub-Index	Name			Mnemonic
$0A_h$	*Rated_Torque_MTRDT*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		Valid only if *CMC_MTRDT* = 0
	Default Value	Minimum	Maximum	Unit
	50	1	60000	10mNm (Ex. 180 = 1.8 Nm)

Description

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

For an optimal functioning it is very important to carefully set this object so that it indicates the motor rated torque when supplied at the rated current set through the object *Rated_Current_MTRDT*.

Sub-Index	Name			Mnemonic
$0B_h$	*Max_Speed_MTRDT*			
	Data Type	Access Type	PDO Mapping	Note

	u16	rw		🖫
	Default Value	Minimum	Maximum	Unit
	30000	1	30000	0.1rpm (Ex. 668 = 66.8rpm)
	Description			
	The object 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*).			

6.4.2 2330h Motor Encoder Data

Index	Name	Type
2330h	*Encoder Motor*	RECORD

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

Sub-Index	Name			Mnemonic
01h	*Configuration_ENCMTR*			
	Data Type	Access Type	PDO Mapping	Note
	u8	rw		🖫
	Default Value	Minimum	Maximum	Unit
	0			
	Description			

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

Bits	Functions
0	Index 0 Encoder without Index signal 1 Encoder with Index signal
1	Pulse count inversion 0 Normal pulses count 1 Backwards pulses count (equivalent to change phase A and B between them)
2	Not used
3	Power Supply 0 Encoder power supply off 1 Encoder power supply on
7..4	Filter

If the encoder signals are affected by electric noise it is possible to use the digital filter present in the drive to eliminate the pulses shorter than a pre-defined value.

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

The equivalent maximum frequency must be calculated considering two times the set time, as it is necessary that such time elapses both for the active signal condition and for the inactive signal. Setting, for example, a Filter value of 1us you can apply a maximum frequency of 500KHz (1+1=2us i.e. 500KHz).

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

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

Description

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

By setting the value to 0 the encoder input is disables. The object *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 encoder native one. For example, by using a 400 pulses/rev encoder, the drive will be able to recognize 1600 different positions/rev.

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

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

6.4.3 2360h Holding Brake Setup

Index	Name	Type
2360h	*Holding Brake Setup*	RECORD

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

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

It allows to set the holding brake handling by the drive and manual control.
The object 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 — Handles, 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 object *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 command the activation and deactivation of the brake autonomously 1 — Manual, the brake control can take place only via a digital input or the object *Control_HBRKC*
2..7	Not used
8..11	Action It allows to define how the manual brake control (obtainable through a digital input or the object *Control_HBRKC*) interacts with the automatic control operated by the drive (if active through bit1=0). In the following description, “active signal” means the active state of the digital input chosen for the manual brake control or the bit0 of the object *Control_HBRKC* set to 1 (the two signals are in logic OR between them).

0	None
1	Release, with active signal the brake is always released independently from the drive control
2	Engage, with active signal the brake is always engaged independently from the drive control
3	Release / Engage, with active signal the brake is always released. With 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

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

Description

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

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

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

Description

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

When the bit0 of the object *Option_HBRKS* is set to 1 and the motor is requested to be enabled, the drive waits for the time specified in the object *Release_Time_HBRKS*, after the brake has been switched off, before going into the operating status.

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

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

6.4.4 $23A0_h$ Current Regulation

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

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

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

It defines the current supplying mode to the motor.

When in the static mode, the motor receives the current *Current_Max_MTRCNF* when it is moving, and the current *Current_Min_MTRCNF* when it is stopped (after the time defined by the object *Current_Idle_Delay_MTRCNF*).

Instead, when in the dynamic mode, the drive supplies to the motor a current value proportional to the load. The current variation always occurs between the minimum and maximum values defined by the object *Current_Min_MTRCNF* and *Current_Max_MTRCNF* respectively.

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

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

6.4.5 2410_h Motion Setup

Index	Name	Type
2410_h	*Motion Setup*	RECORD

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

Sub-Index	Name			Mnemonic
01_h	*Feedback_MTNSTP*			
	Data Type	Access Type	PDO Mapping	Note
	u8	rw	RPDO	🖫
	Default Value	Minimum	Maximum	Unit
	0			
	Description			

The object is useful to enable and configure the closed loop operation of the motor.

The following table resumes the use of the object bits.

Bit	Description
0	Encoder Feedback It allows you to enable closed loop operation using the motor encoder for feedback. Value / Description: 0 – Open loop (feedback disabled) 1 – Closed loop (feedback enabled)
1..2	

3	Encoder Timing It allows to choose Encoder Timing mode. Value / Description 0 – Only when necessary (ex. at the first motor enabling after device start) 1 – Always at every motor enabling
4..7	Not used, set to 0.

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

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

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

6.4.6 2440h Position Loop

Index	Name	Type
2440h	*Position Loop*	RECORD

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

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

6.4.7 2480_h Error Class Setup

Index	Name	Type
2480_h	*Error Class Setup*	RECORD

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

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

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

Class	Description	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.

6.4.8 2810$_h$ Digital Inputs Assignment

The objects 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 *Fault reset* contained in the object *Controlword.* Please note that a same input can be associated with more functions and actions.

In the following description the inputs numbering starts from 2 instead of 0 for consistency with other types of drives of the DDS series, where the digital inputs DI0 and DI1 have special properties.

Index	Name	Type
2810$_h$	*Digital Inputs Action*	RECORD

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

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

Sub-Index	Name			Mnemonic
02$_h$	*Fault_Reset_DIA*			
	Data Type: u16	Access Type: rw	PDO Mapping	Note
	Default Value: 0502$_h$	Minimum	Maximum	Unit

Description

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

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

high byte, active status	
Value	Description
4	Rising edge
5	Falling edge

Low byte, input number	
Value	Description
2	Digital input 2
3	Digital input 3
4	Digital input 4
5	Digital input 5
6	Digital input 6
7	Digital input 7

Sub-Index	Name			Mnemonic
03h	*Positive_Limit_DIA*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		
	Default Value	Minimum	Maximum	Unit
	0207h			

Description

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

The object 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

Sub-Index	Name			Mnemonic
04h	*Negative_Limit_DIA*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		
	Default Value	Minimum	Maximum	Unit
	0206h			

Description

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

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

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

7	Digital Input 7

Sub-Index	Name			Mnemonic
05_h	*Home_DIA*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		
	Default Value	Minimum	Maximum	Unit
	0205_h			

Description

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

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

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

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

Description

Reserved, do no use.

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

Description

Reserved, do not use.

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

				Description
	Reserved, do not use.			

Sub-Index			Name	Mnemonic
09_h				
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		🖫
	Default Value	Minimum	Maximum	Unit
				Description
	Reserved, do not use.			

Sub-Index			Name	Mnemonic
$0A_h$	*Holding_Brake_DIA*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		🖫
	Default Value	Minimum	Maximum	Unit
	0			
				Description

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

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

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

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

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

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

Sub-Index			Name	Mnemonic
$0C_h$				
	Data Type	Access Type	PDO Mapping	Note

	u16	rw		
	Default Value	Minimum	Maximum	Unit
	Reserved, do not use.			Description

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

6.4.9 2830_h Digital Outputs 0 Assignment

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

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

Sub-Index				Name	Mnemonic
01_h	*Source_DOA*				
	Data Type	Access Type	PDO Mapping		Note
	u16	rw	RPDO		
	Default Value	Minimum	Maximum		Unit
	8040_h				

Description

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

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

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

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

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

6.4.10 2831_h Digital Output 1 Assignment

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

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

Sub-Index	Name			Mnemonic
01_h	*Source_DOA*			
	Data Type: u16	Access Type: rw	PDO Mapping: RPDO	Note: 🖫
	Default Value: 0041_h	Minimum	Maximum	Unit

Description

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

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

8060_h...$806F_h$	*Statusword* bit n = 0 (8060_h=bit0, 8061_h=bit1, etc.)

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

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

6.4.11 2832_h Digital Output 2 Assignment

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

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

Sub-Index	Name			Mnemonic
01_h	*Source_DOA*			
	Data Type: u16	Access Type: rw	PDO Mapping: RPDO	Note: 🖫
	Default Value: $006A_h$	Minimum	Maximum	Unit

Description

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

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

	For example, setting the value 8040_h the output will be activated if there is no fault condition. Instead, setting the value 8067_h the output will be activated every time the bit 7 (bit *warning*) of the *Statusword* will have value 0. When the value 0000_h (General Purpose) is selected the output status is controlled by the bit 2 value of the object *Outputs_DOV*.

6.4.12 3080_h Sync monitoring

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

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

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

Index	Name	Type
3080_h	*Sync Guard*	RECORD

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

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

Description

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

Bits	Functions
0	Enable Sync monitoring. On the transition of the bit from 0 to 1 (transition from disabled to enabled state) the objects *Out_Of_Sync_Counter_SYGD*, *Peak_Out_Of_Sync_SYGD* and *Sync_Counter_SYGD* are set to 0.

	0 Monitoring disabled 1 Monitoring enabled
1	Enable emergency in case of violation of timing waited for Sync. 0 Emergency not generated in case of violation 1 Emergency generated in case of violation
2..4	Not used, set to 0.
5	*Out_Of_Sync_Counter_SYGD* reset. Reset occurs on the writing of the object *Option_SYGD*. The bit state after the writing has not effect. 0 No action 1 *Out_Of_Sync_Counter_SYGD* reset
6	*Peak_Out_Of_Sync_SYGD* object reset. Reset occurs on the writing of the object *Option_SYGD*. The bit state after the writing has no effect. 0 No action 1 *Peak_Out_Of_Sync_SYGD* object reset
7	*Sync_Counter_SYGD* reset. Reset occurs on the writing of the object *Option_SYGD*. The bit state after the writing has no effect. 0 No action 1 *Sync_Counter_SYGD* reset

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

Description

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

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

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

Sub-Index	Name			Mnemonic
03h	*Sync_Counter_SYGD*			
	Data Type: u32	Access Type: ro	PDO Mapping	Note
	Default Value: 0	Minimum	Maximum	Unit
	Description			

Counter of the received Sync independently from their timing.

The counter advances one unit each time a SYNC object is received.

The counter can be reset by the bit 7 of the object *Option_SYGD*.

Sub-Index	Name			Mnemonic
04h	*Sync_Expected_Period_SYGD*			
	Data Type: u32	Access Type: rw	PDO Mapping	Note: 🖫 Changeable only with monitoring disabled
	Default Value: 1000	Minimum: 1000	Maximum: 100000	Unit: 1us (Ex. 2500 = 2.5ms)
	Description			

It allows to set the nominal period with which the SYNC object is received.

Sub-Index	Name			Mnemonic
05h	*Sync_Max_Jitter_SYGD*			
	Data Type: u16	Access Type: rw	PDO Mapping	Note: 🖫 Changeable only with monitoring disabled
	Default Value: 200	Minimum: 1	Maximum: 10000	Unit: 1us (Ex. 200 = 200us)
	Description			

It allows to set the maximum allowable tolerance for the receiving the SYNC object with respect to the nominal period.

If the SYNC object is received in advance or in delay, with respect to the set nominal period, of a time greater than the value set in this object, the *Out_Of_Sync_Counter_SYGD* counter is incremented by 10 units.

The value of the object *Sync_Max_Jitter_SYGD* cannot be set to a value greater than 50% of the value set in the object *Sync_Expected_Period_SYGD*. Otherwise the activation of the monitoring will generate an error.

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

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

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

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

6.4.13 3140$_h$ Device Status

Index	Name	Type
3140$_h$	*Device Status*	RECORD

Sub-Index	Name			Mnemonic
00$_h$	*Number_of_Entries*			
	Data Type: u8	Access Type: const	PDO Mapping	Note
	Default Value: 03$_h$	Minimum	Maximum	Unit
	Description: Number of record entries.			

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

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

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

6.4.14 3210$_h$ Digital Inputs Value

Index	Name	Type
3210$_h$	*Digital Inputs Value*	RECORD

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

Sub-Index	Name			Mnemonic
01$_h$	*Inputs_DIV*			
	Data Type: u16	Access Type: ro	PDO Mapping: TPDO	Note
	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	Reserved, ignore the value
1	Reserved, ignore the value
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.

The inputs numbering starts from 2 instead of 0 for consistency with other types of drives of the DDS series, where the digital inputs DI0 and DI1 have special properties.

6.4.15 3230h Digital Outputs Value

Index	Name	Type
3230h	*Digital Outputs Value*	RECORD

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

Sub-Index	Name			Mnemonic
01h	*Outputs_DOV*			
	Data Type: u16	Access Type: rw	PDO Mapping: RPDO	Note
	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 object bits are assigned to the digital outputs as follows:

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

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

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

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

Description

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

To change one single bit of the object *Outputs_DOV* simply write this object with a value as follows: the bits from bit 0 to bit 2 must contain the number of bits of the object *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 object *Outputs_DOV* is put to 0 and vice versa.

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

set to 1.

The following table shows the use of the bits of the object *Set_Output_DOV.*

Bit	Description
0..3	Number of the bits of the object *Outputs_DOV* on which you want to act. Value / Description: 0 – Bit 0 1 – Bit 1 … – Bit n 15 – Bit 15
4..6	Not used, set to 0
7	Logic level to set in the bits of the object *Outputs_DOV* Value / Description: 0 – Bit = 1 1 – Bit = 0

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

6.4.16 3250$_h$ Analog Input 0 Value

Index	Name	Type
3250$_h$	*Analog Inputs 0 Value*	RECORD

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

Sub-Index	Name			Mnemonic
01$_h$	*Voltage_AIV*			
	Data Type: i16	Access Type: Ro	PDO Mapping: TPDO	Note
	Default Value	Minimum	Maximum	Unit: 1mV (Ex. 5302 = 5.302V)
	Description: It indicates the voltage at the analog input 0.			

6.4.17 3251$_h$ Analog Input 1 Value

Index	Name	Type
3251$_h$	*Analog Inputs 1 Value*	RECORD

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

Sub-Index	Name			Mnemonic
01$_h$	*Voltage_AIV*			
	Data Type: i16	Access Type: ro	PDO Mapping: TPDO	Note
	Default Value	Minimum	Maximum	Unit: 1mV (Ex. -2280 = -2.280V)
	Description: It indicates the voltage at the analog input 1.			

6.4.18 3260_h Analog Output 0 Value

Index	Name	Type
3260_h	*Analog Outputs 0 Value*	RECORD

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

Sub-Index	Name			Mnemonic
01_h	*Voltage_AOV*			
	Data Type	Access Type	PDO Mapping	Note
	i16	rw	RPDO	🖫
	Default Value	Minimum	Maximum	Unit
	0			1mV (Ex. 2450 = 2.450V)
	Description			
	It allows to set the voltage of the analog output 0.			

6.4.19 3261_h Analog Output 1 Value

Index	Name	Type
3261_h	*Analog Outputs 1 Value*	RECORD

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

Sub-Index	Name			Mnemonic
01_h	*Voltage_AOV*			
	Data Type	Access Type	PDO Mapping	Note
	i16	rw	RPDO	🖫
	Default Value	Minimum	Maximum	Unit
	0			1mV (Ex. 8520 = 8.520V)
	Description			
	It allows to set the voltage of the analog output 1.			

6.4.20 3310$_h$ Motor Configuration

Index	Name	Type
3310$_h$	*Motor Configuration*	RECORD

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

Sub-Index	Name			Mnemonic
01$_h$				
	Data Type: u8	Access Type: rw	PDO Mapping: RPDO	Note: 🖫
	Default Value	Minimum	Maximum	Unit
	Description: Reserved. Do not use.			

Sub-Index	Name			Mnemonic
02$_h$	*Current_Min_MTRCNF*			
	Data Type: u16	Access Type: rw	PDO Mapping: RPDO	Note: 🖫
	Default Value: 4000	Minimum	Maximum	Unit: 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 object *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 *Current_Idle_Delay_MTRCNF* from the stop

The value is expressed in percentage of the motor rated current set by the object *Rated_Current_MTRDT*. For example, if the motor has a rated current of 4Arms and a minimum current equal to 25%, the drive will never supply less than 1Arms to the motor.

Sub-Index	Name			Mnemonic
03$_h$	*Current_Max_MTRCNF*			
	Data Type: u16	Access Type: Rw	PDO Mapping: RPDO	Note: 🖫
	Default Value: 8000	Minimum	Maximum	Unit: 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 object *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 object *Current_Max_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 object *Rated_Current_MTRDT*. For example, if the motor has a rated current of 4Arms and a minimum current equal to 75%, the drive will never supply less than 3Arms to the motor.

Sub-Index	Name			Mnemonic
04_h	*Current_Idle_Delay_MTRCNF*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		🖫
	Default Value	Minimum	Maximum	Unit
	500	2	10000	1ms (Ex. 3500 = 3.5s)
	Description			
	It allows to set the motor stopping time before the current reaches the value defined by the object *Current_Min_MTRCNF.* When the dynamic current regulation mode is active this object has no effect.			

6.4.21 3312_h Motor Value

Index	Name	Type
3312_h	*Motor Value*	RECORD

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

Sub-Index	Name			Mnemonic
01_h	*Current_MTRV*			
	Data Type	Access Type	PDO Mapping	Note
	I16	ro	TPDO	
	Default Value	Minimum	Maximum	Unit
				10mArms (Ex. 225 = 2.25 Arms)
	Description			
	It indicates the actual current supplied to the motor expressed in effective value.			

Sub-Index	Name			Mnemonic
02_h	*Utilization_MTRV*			
	Data Type	Access Type	PDO Mapping	Note
	i16	ro	TPDO	
	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.			

6.4.22 3360$_h$ Holding Brake Control

Index	Name	Type
3360$_h$	*Holding Brake Control*	RECORD

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

Sub-Index	Name			Mnemonic
01$_h$	*Control_HBRKC*			
	Data Type: u8	Access Type: rw	PDO Mapping: RPDO	Note: 🖫
	Default Value: 0	Minimum	Maximum	Unit

Description

It allows you to manually control the holding brake.

The following table resumes the use of the object bits.

Bit	Description
0	Manual control It allows you to manually control the holding brake. Value / Description: 0 – Inactive manual control signal 1 – Active manual control signal The effect on the brake of the logic level assumed by this bit depends on the configuration of the object *Option_HBRKS,* therefore for a detailed description refer to chapter 6.4.3 2360$_h$ Holding Brake Setup. The bit0 is in logic OR with the digital input chosen for the manual control of the brake.
1..7	Not used, set to 0.

Sub-Index	Name			Mnemonic
02$_h$	*Status_HBRKC*			
	Data Type: u8	Access Type: ro	PDO Mapping: TPDO	Note
	Default Value	Minimum	Maximum	Unit
	Description			

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

The following table resumes the meaning of the object 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

6.4.23 3450$_h$ Profile Velocity Configuration

Index	Name	Type
3450$_h$	*Profile Velocity Configuration*	RECORD

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

Sub-Index	Name			Mnemonic
01$_h$	*Max_Slippage_TimeOut_PVCNF*			
	Data Type: u16	Access Type: rw	PDO Mapping: RPDO	Note: 🖫
	Default Value: 10	Minimum	Maximum: 30000	Unit: 1ms (Ex. 280 = 280ms)

Description

It allows to set the time after which the bit *Max slippage error* of the *Statusword* is set to 1 in the *Profile Velocity* mode.

The error *Max slippage error* occurs when the motor speed differs from the required speed beyond the value set through the object *Max_slippage*.

The drive is able to know the actual motor speed only if the motor is provided with encoder and the drive is configured in closed loop mode.

6.4.24 3470$_h$ Profile Torque Configuration

Index	Name	Type
3470$_h$	*Profile Torque Configuration*	RECORD

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

Sub-Index	Name			Mnemonic
01$_h$	*Mode_PTCNF*			
	Data Type: u8	Access Type: rw	PDO Mapping: RPDO	Note: 🖫
	Default Value: 0	Minimum	Maximum	Unit

Description

It allows to configure the options of the *Profile Torque* mode.

Normally in the torque control the motor speed is limited only by the load and by the characteristic of the motor. However, in many applications it is useful to set a maximum speed

to prevent the motor to reach high speeds without load. Setting to 1 the bit 0 of the object *Mode_PTCNF*, the drive limits the motor maximum speed to the value set through the object *Profile_velocity*.

The following table shows the use of the objet bits.

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

6.4.25 3490h Interpolated Profile Configuration

Index	Name	Type
3490h	*Interpolated Position Configuration*	RECORD

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

Sub-Index	Name			Mnemonic
01h	*Mode_IPCNF*			
	Data Type: u8	Access Type: rw	PDO Mapping: RPDO	Note: 🖫
	Default Value: 0	Minimum	Maximum	Unit
	Description: It allows to configure the options of the *Interpolated Position* mode. The following table resumes the use of the object bits.			

Bit	Description
0..5	Not used, set to 0.
6	Micro-interpolation. Between a set-point and the next, the drive is able to interpolate the path with intermediate micro positions in order to improve the smoothness of the movement. Value / Description: 0 – Micro-interpolation active 1 – Micro-interpolation disabled
7	Limitation of acceleration and deceleration.. In the *Interpolated Position* mode, usually it is the profile generator that limits the speed and the accelerations during the construction of the motion path, however, in some case, it can be advantageous to have a ramp limitation also from the drive. Value / Description: 0 – Unlimited acceleration and deceleration 1 – Limited acceleration and deceleration When the limitation is active, the acceleration and deceleration values are set through the objects *Profile_acceleration* and *Profile_deceleration* respectively.

Sub-Index	Name			Mnemonic
02_h	*Dampening_A_Value_IPCNF*			
	Data Type: i8	Access Type: rw	PDO Mapping: RPDO	Note:
	Default Value: 0	Minimum:	Maximum:	Unit:

Description

In the *Interpolated Position* mode if the set—points update will not take place with regular frequency, according to the period set through the object *Interpolation_time_period*, the movement of the motor becomes irregular. Through this object it is possible to prepare a filtering that makes tolerable modest changes in the set-points update frequency.

Positive values increase the effect of the filtering while negative values reduce it.

6.5 Object with index between 6000_h and $9FFF_h$

The object with index between 6000_h e $9FFF_h$ are specific to the profile /CiA402/. Their function, the access mode, etc. are described in the manuals of the DSP402 standard available on the official site www.can-cia.org, therefore in this manual there will be no detailed description of each object but a synthetic summary of the implemented ones.

6.5.1 $603F_h$ Error code

Index	Name			Mnemonic
$603F_h$	*Error_code*			
	Data Type	Access Type	PDO Mapping	Note
	u16	ro	TPDO	
	Default Value	Minimum	Maximum	Unit
	Description			
	This object provides the code of the last error occurred in the device. This object provides the same information as the lower 16-bit of sub-index 01_h of the pre-defined error field (1003_h).			

6.5.2 6040_h Controlword

Index	Name			Mnemonic
6040_h	*Controlword*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw	RPDO	
	Default Value	Minimum	Maximum	Unit

Description

This object indicates the received command controlling the device FSA.

It is structured in bits and the functionality of each bit changes according to the mode of operation chosen with the object *Modes of operation*.

Bit	Description		Bit	Description	
15	r	Reserved	7	fr	Fault reset
14	r	Reserved	6	abrl	Absolute / Relative
13	r	Reserved		rr	Reference ramp
12	r	Reserved	5	csi	Change set immediately
11	r	Reserved		ulkr	unlock ramp
10	r	Reserved	4	nsp	New set-point
9	cosp	Change on setpoint		hos	Homing operation start
8	h	Halt		ei	Enable interpolation
				er	enable ramp
			3	eo	Enable operation
			2	qs	quick stop
			1	ev	Enable voltage
			0	so	Switch on

Mode of operation

Profile position mode

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

Mode of operation

Velocity mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					r		h	fr	rr	ulkr	er	f	oe	so	rsto

Mode of operation

Profile velocity mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					r		h	fr	r	r	r	f	oe	so	rsto

Mode of operation

Profile torque mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					r		h	fr	r	r	r	f	oe	so	rsto

Mode of operation

Homing mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					r		h	fr	0	0	hos	f	oe	so	rsto

Mode of operation

Interpolated position mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					r		h	fr	0	0	ei	f	oe	so	rsto

Mode of operation

Cyclic synchronous position mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					r		h	fr				f	oe	so	rsto

Mode of operation

Cyclic synchronous velocity mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					r		h	fr				f	oe	so	rsto

Mode of operation

Cyclic synchronous torque mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					r		h	fr				f	oe	so	rsto

Mode of operation

Cyclic synchronous torque mode with commutation angle

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					r			fr				f	oe	so	rsto

6.5.3 6041_h Statusword

Index	Name			Mnemonic
6041_h	*Statusword*			
	Data Type	Access Type	PDO Mapping	Note
	u16	ro	TPDO	
	Default Value	Minimum	Maximum	Unit

Description

This object provides the status of the device FSA.

It is structured in bits and the functionality of each bit changes according to the mode of operation chosen with the object *Modes of Operation*.

Bit	Description		Bit	Description	
15	r	Reserved	7	w	Warning
14	r	Reserved	6	sod	Switch on disabled
13	fe	Following error	5	qs	Quick stop (0=active)
	he	Homing error	4	ve	Voltage enabled
	mse	Max slippage error	3	f	Fault
12	spa	Set-point acknowledge	2	oe	Operation enabled
	ha	Homing attained	1	so	Switched on
	ipa	Ip mode active	0	rsto	Ready to switch on
	spd	Speed is equal 0			
	dfcv	Drive follows the command value			
11	ila	Internal limit active			
10	tr	Target reached (or Velocity = 0)			
9	rm	Remote			
8	h	Halt			

Mode of operation

Profile position mode

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

Mode of operation

Velocity mode

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

Mode of operation

Profile velocity mode

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

Mode of operation

Profile torque mode

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

Mode of operation

Homing mode

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

Mode of operation

Interpolated position mode

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

Mode of operation

Cyclic synchronous position mode

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

Mode of operation

Cyclic synchronous velocity mode

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

Mode of operation

Cyclic synchronous torque mode

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

Mode of operation

Cyclic synchronous torque mode with commutation angle

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

6.5.4 6060_h Modes of operation

Index	Name			Mnemonic
6060_h	*Modes_of_Operation*			
	Data Type	Access Type	PDO Mapping	Note
	i8	rw	RPDO	
	Default Value	Minimum	Maximum	Unit
	1			

Description

This object indicates the requested operation mode.

The table below shows the values definition:

1	Profile Position
3	Profile Velocity
4	Torque Profile
6	Homing
7	Interpolated Position

6.5.5 6061_h Modes of operation display

Index	Name			Mnemonic
6061_h	*Modes_of_operation_display*			
	Data Type	Access Type	PDO Mapping	Note
	i8	ro	TPDO	
	Default Value	Minimum	Maximum	Unit

Description

This object provides the actual operation mode.

The table below shows the values definition:

1	Profile Position
3	Profile Velocity
4	Torque Profile
6	Homing
7	Interpolated Position

6.5.6 6062_h Position demand value

Index	Name			Mnemonic
6062_h	*Position_demand_value*			
	Data Type	Access Type	PDO Mapping	Note
	i32	ro	TPDO	
	Default Value	Minimum	Maximum	Unit
				0.0001rev (Ex. 45524 = 4.5524rev)

Description

This object provides the demanded position value.

The value is given in 1/10000 of revolution.

6.5.7 6064_h Position actual value

Index	Name			Mnemonic
6064_h	*Position_actual_value*			
	Data Type	Access Type	PDO Mapping	Note
	i32	ro	TPDO	
	Default Value	Minimum	Maximum	Unit
				0.0001rev (Ex. 5000 = 0.5000rev)

Description

This object provides the actual value of the position.

The value is given in 1/10000 of revolution.

6.5.8 6065_h Following error window

Index	Name			Mnemonic
6065_h	*Following_error_window*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw	RPDO	🖫
	Default Value	Minimum	Maximum	Unit
	10000			0.0001rev (Ex. 20000 = 2.0000rev)
	Description			
	This object indicates the configured range of tolerated position values symmetrically to the position demand value. If the position actual value is out of the following error window, a following error occurs. If the value of the following error window is $FFFFFFFF_h$, the following control is switched off. The value is given in 1/10000 of revolution.			

6.5.9 6066_h Following error time out

Index	Name			Mnemonic
6066_h	*Following_error_time_out*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw	RPDO	🖫
	Default Value	Minimum	Maximum	Unit
	10		30000	1ms (Ex. 250 = 250ms)
	Description			
	This object indicates the configured time for a following error condition, after that the bit 13 of the *Statusword* is set to 1. The value is given in ms.			

6.5.10 6067_h Position window

Index	Name			Mnemonic
6067_h	*Position_window*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw	RPDO	🖫
	Default Value	Minimum	Maximum	Unit
	10			0.0001rev (Ex. 20000 = 2.0000rev)
	Description			
	This object indicates the configured symmetrical range of accepted positions relative to the target position. If the actual value of the position is within the position window, this target position is regarded as having been reached. If the value of the position window is $FFFFFFFF_h$, the position window control is switched off. The value is given in 1/10000 of revolution.			

6.5.11 6068_h Position window time

Index	Name			Mnemonic
6068_h	*Position_window_time*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw	RPDO	🖫
	Default Value	Minimum	Maximum	Unit
	0		30000	1ms (Ex. 600 = 0.6s)
	Description			
	This object indicates the configured time, during which the actual position within the position window is measured. The value is given in ms.			

6.5.12 $606B_h$ Velocity demand value

Index	Name			Mnemonic
$606B_h$	*Velocity_demand_value*			
	Data Type	Access Type	PDO Mapping	Note
	i32	ro	TPDO	
	Default Value	Minimum	Maximum	Unit
				0.1rpm (Ex. 4525 = 452.5rpm)
	Description			
	This object shall provide the output value of the trajectory generator. The value is given in 0.1rpm.			

6.5.13 $606C_h$ Velocity actual value

Index	Name			Mnemonic
$606C_h$	*Velocity_actual_value*			
	Data Type	Access Type	PDO Mapping	Note
	i32	ro	TPDO	
	Default Value	Minimum	Maximum	Unit
				0.1rpm (Ex. 3850 = 385.0rpm)
	Description			
	This shall provide the actual velocity. The value is given in 0.1rpm.			

6.5.14 $606D_h$ Velocity window

Index	Name			Mnemonic
$606D_h$	*Velocity_window*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw	RPDO	🖫
	Default Value	Minimum	Maximum	Unit
	100			0.1rpm (Ex. 1000 = 100.0rpm)
	Description			
	This object indicates the configured velocity window. The value is given in 0.1rpm.			

6.5.15 606E$_h$ Velocity window time

Index	Name			Mnemonic
606E$_h$	*Velocity_window_time*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw	RPDO	🖫
	Default Value	Minimum	Maximum	Unit
	0		30000	1ms (Ex. 200 = 200ms)
	Description			
	This object indicates the configured velocity window time. The value is given in ms.			

6.5.16 606F$_h$ Velocity threshold

Index	Name			Mnemonic
606F$_h$	*Velocity_threshold*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw	RPDO	🖫
	Default Value	Minimum	Maximum	Unit
	60			0.1rpm (Ex. 500 = 50.0rpm)
	Description			
	This object indicates the configured velocity threshold. The value is given in 0.1rpm.			

6.5.17 6070$_h$ Velocity threshold time

Index	Name			Mnemonic
6070$_h$	*Velocity_threshold_time*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw	RPDO	🖫
	Default Value	Minimum	Maximum	Unit
	0		30000	1ms (Ex. 200 = 200ms)
	Description			
	This object indicates the configured velocity threshold time. The value is given in ms.			

6.5.18 6071$_h$ Target Torque

Index	Name			Mnemonic
6071$_h$	Target_Torque			
	Data Type	Access Type	PDO Mapping	Note
	i16	rw	RPDO	🖫
	Default Value	Minimum	Maximum	Unit
	0	-1300	1300	0.1% (Ex. 405 = 40.5%)
	Description			
	This object indicates the configured input value for the torque controller in profile torque mode. The value is given in 0.1% of the nominal motor torque.			

6.5.19 6077_h Torque actual value

Index	Name			Mnemonic
6077_h	Torque_actual_value			
	Data Type	Access Type	PDO Mapping	Note
	i16	ro	TPDO	
	Default Value	Minimum	Maximum	Unit
				0.1% (Ex. 405 = 40.5%)
	Description			
	This object provides the actual value of the torque. The value is given in 0.1% of the nominal motor torque.			

6.5.20 6078_h Current actual value

Index	Name			Mnemonic
6078_h	*Current_actual_value*			
	Data Type	Access Type	PDO Mapping	Note
	i16	ro	TPDO	
	Default Value	Minimum	Maximum	Unit
				0.1% (Ex. 800 = 80.0%)
	Description			
	This objects provides the actual value of the current supplied to the motor. The value is given in 0.1% of the nominal motor current.			

6.5.21 $607A_h$ Target position

Index	Name			Mnemonic
$607A_h$	Target_position			
	Data Type	Access Type	PDO Mapping	Note
	i32	rw	RPDO	🖫
	Default Value	Minimum	Maximum	Unit
				0.0001rev (Ex. 50000 = 5.0000rev)
	Description			
	This object indicates the commanded position that the drive should move to in position profile mode using the current settings of motion control parameters such as velocity, acceleration, deceleration, motion profile type etc. The value of this object shall be interpreted as absolute or relative depending on the abs/rel flag in the *Controlword*. The value is given in 1/10000 of revolution.			

6.5.22 $607C_h$ Home offset

Index	Name			Mnemonic
$607C_h$	Home_offset			
	Data Type	Access Type	PDO Mapping	Note
	i32	rw	RPDO	🖫
	Default Value	Minimum	Maximum	Unit
				0.0001rev (Ex. 20000 = 2.0000rev)
	Description			

	This object indicates the configured difference between the zero position for the application and the machine home position (found during homing). During homing, the machine home position is found and once the homing is completed, the object *Position_actual_value* is set to the value of the object *Home_offset*. The value is given in 1/10000 of revolution.

6.5.23 607D$_h$ Software position limit

To disable the software position limits, the *Min position limit* and *Max position limit* object shall be set both to 0.

Index	Name	Type
607D$_h$	*Software position limit*	RECORD

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

Sub-Index	Name			Mnemonic
01$_h$	*Min_position_limit*			
	Data Type: i32	Access Type: rw	PDO Mapping: RPDO	Note:
	Default Value: 0	Minimum:	Maximum:	Unit: 0.0001rev (Ex. -1000000 = -100rev)
	Description: This object indicates the configured minimal software position limits. These parameters define the absolute position limits for the position demand value. The value is given in 1/10000 of revolution.			

Sub-Index	Name			Mnemonic
02$_h$	*Max_position_limit*			
	Data Type: i32	Access Type: rw	PDO Mapping: RPDO	Note:
	Default Value: 0	Minimum:	Maximum:	Unit: 0.0001rev (Ex. 1000000 = 100rev)
	Description: This object indicates the configured maximal software position limits. These parameters shall define the absolute position limits for the position demand value. The value is given in 1/10000 of revolution.			

6.5.24 607F$_h$ Max profile velocity

Index	Name			Mnemonic
607F$_h$	*Max_profile_velocity*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw	RPDO	🖫
	Default Value	Minimum	Maximum	Unit
	30000	1	30000	0.1rpm (Ex. 5000 = 500.0rpm)
	Description			
	This object indicates the configured maximal allowed velocity in either direction during a profiled motion. The value is given in 0,1rpm.			

6.5.25 6081$_h$ Profile velocity

Index	Name			Mnemonic
6081$_h$	*Profile_velocity*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw	RPDO	🖫
	Default Value	Minimum	Maximum	Unit
	600		30000	0.1rpm (Ex. 4500 = 450.0rpm)
	Description			
	This object indicates the configured velocity normally attained at the end of the acceleration ramp during a profiled motion and is valid for both directions of motion. The value is given in 0.1rpm.			

6.5.26 6083$_h$ Profile acceleration

Index	Name			Mnemonic
6083$_h$	*Profile_acceleration*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw	RPDO	🖫
	Default Value	Minimum	Maximum	Unit
	100	5	300000	rpm/s (Ex. 100 = 100rpm/s)
	Description			
	This object indicates the configured acceleration. The value is given in rpm/s.			

6.5.27 6084$_h$ Profile deceleration

Index	Name			Mnemonic
6084$_h$	*Profile_deceleration*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw	RPDO	🖫
	Default Value	Minimum	Maximum	Unit
	300	5	300000	rpm/s (Ex. 300 = 300rpm/s)
	Description			
	This object indicates the configured deceleration. The value is given in rpm/s.			

6.5.28 6085_h Quick stop deceleration

Index	Name			Mnemonic
6085_h	*Quick_stop_deceleration*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw	RPDO	🖫
	Default Value	Minimum	Maximum	Unit
	5000	5	300000	rpm/s (Ex. 5000 = 5000rpm/s)
	Description			
	This object indicates the configured deceleration used to stop the motor when the quick stop function is activated. The quick stop deceleration is also used if fault is detected in the device. The value is given in rpm/s.			

6.5.29 6087_h Torque slope

Index	Name			Mnemonic
6087_h	*Torque_slope*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw	RPDO	🖫
	Default Value	Minimum	Maximum	Unit
				0.1%/s (Ex. 558 = 55.8%/s)
	Description			
	This object indicates the configured rate of change of torque. The value is given in units of per thousand of rated torque per second.			

6.5.30 6098_h Homing method

Index	Name			Mnemonic
6098_h	*Homing_method*			
	Data Type	Access Type	PDO Mapping	Note
	i8	rw	RPDO	🖫
	Default Value	Minimum	Maximum	Unit

Description

This object indicates the configured homing method that shall be used.

The table below shows the values definition:

Homing methods selectable through the object Homing_method

Cod.	Description	Sensors used			
		PLS	NLS	HS	IDX
1	At the start, if negative limit switch inactive counterclockwise direction up to the limit switch, then reverse and homing at the first index outside the negative limit switch. At the start, if negative limit switch active clockwise direction up to leave the limit switch, then homing at the first index outside the negative limit switch.		●		●
2	At the start, if positive limit switch inactive clockwise direction up to the limit switch, then reverse and homing at the first index outside the positive limit switch. At the start, if positive limit switch active counterclockwise direction up to leave the limit switch, then homing at the first index outside the positive limit switch.	●			●
3	At the start, if home switch inactive initial direction clockwise up to home switch, then reverse and homing at the first index outside the home switch. At the start, if home switch active initial direction counterclockwise up to leave the switch, then homing at the first index outside the home switch.			●	●
4	At the start, if home switch inactive initial direction clockwise up to home switch, then homing at the first index inside the home switch. At the start, if home switch active initial direction counterclockwise up to leave the switch, then reverse and homing at the first index inside the home switch.			●	●
5	At the start, if home switch active initial direction clockwise up to leave the switch, then homing at the first index outside the home switch. At the start, if home switch inactive initial direction counterclockwise up to find the switch, then reverse and homing at the first index outside the home switch.			●	●
6	At the start, if home switch active initial direction clockwise up to leave the switch, then reverse and homing at the first index inside the home switch. At the start, if home switch inactive initial direction counterclockwise up to find the switch, then reverse and homing at the first index inside the home switch.			●	●

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

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

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

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

37	Homing at the actual position.				

6.5.31 6099h Homing speeds

Index	Name	Type
6099h	*Homing speeds*	RECORD

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

Sub-Index	Name			Mnemonic
01h	*Speed_during_search_for_switch*			
	Data Type: u32	Access Type: rw	PDO Mapping: RPDO	Note: 🖫
	Default Value: 600	Minimum	Maximum: 30000	Unit: 0.1rpm (Ex. 600 = 60.0rpm)
	Description: This object indicates the speed during search for switch in homing procedure. The value is given in 0.1rpm.			

Sub-Index	Name			Mnemonic
02h	*Speed_during_search_for_zero*			
	Data Type: u32	Access Type: rw	PDO Mapping: RPDO	Note: 🖫
	Default Value: 60	Minimum	Maximum: 30000	Unit: 0.1rpm (Ex. 100 = 10.0rpm)
	Description: This object indicates the speed during search for zero in homing procedure. The value is given in 0.1rpm.			

6.5.32 60C1$_h$ Interpolation data record

Index	Name	Type
60C1$_h$	*Interpolation_data_record*	RECORD

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

Sub-Index	Name			Mnemonic
01$_h$	*Interpolation_data_record*			
	Data Type: i32	Access Type: rw	PDO Mapping: RPDO	Note: 🖫
	Default Value	Minimum	Maximum	Unit: 0.0001rev (Ex. 25500 = 2.55rev)
	Description: The object allows to set the position set-point used by the *Interpolated Position* mode. The value is given in 1/10000 of rev.			

6.5.33 $60C2_h$ *Interpolation Time Period*

When using the *Interpolated Position* mode it is very important to configure accurately the objects of this record so that they correspond to the period with which the set-point is updated (Sync period).

The objects can be modified only if the interpolation is not active (bit *ipa* in the *Statusword* set to 0).

Index	Name	Type
$60C2_h$	*Interpolation_time_period*	RECORD

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

Sub-Index	Name			Mnemonic
01_h	*Interpolation_time_period_value*			
	Data Type: u8	Access Type: rw	PDO Mapping: RPDO	Note:
	Default Value: 1	Minimum:	Maximum:	Unit: Defined by the object *Interpolation_time_index*

Description:

The object allows you to set the interpolation period, i.e. the time between two successive updates of the set-point.

For best operation it is important that the object value corresponds exactly to the period with which the SYNC object is received.

Please note how the measurement unit is not predefined but depends on the content of the object *Interpolation_time_index*. The period in seconds can be calculated as:
T = *Interpolation_time_period_value* * $10^{Interpolation\_time\_index}$

If, for example, the objects are configured *Interpolation_time_period_value* = 2 and *Interpolation_time_index* = -3, the interpolation period is set to 2ms ($2 * 10^{-3}$).

Sub-Index	Name			Mnemonic
02_h	*Interpolation_time_index*			
	Data Type: i8	Access Type: rw	PDO Mapping: RPDO	Note:
	Default Value: -3	Minimum: -128	Maximum: 63	Unit:

Description:

The object defines the measurement unit used for the object *Interpolation_time_period_value* and represents the exponent in base 10 that multiplies the object *Interpolation_time_period_value.*

6.5.34 60F4$_h$ Following error actual value

Index	Name			Mnemonic
60F4$_h$	*Following_error_actual_value*			
	Data Type	Access Type	PDO Mapping	Note
	i32	ro	TPDO	
	Default Value	Minimum	Maximum	Unit
				0.0001rev (Ex. 20000 = 2.0000rev)
	Description			
	This object provides the actual value of the following error. The value is given in 1/10000 of revolution.			

6.5.35 60F8$_h$ Max slippage

Index	Name			Mnemonic
60F8$_h$	*Max_slippage*			
	Data Type	Access Type	PDO Mapping	Note
	i32	ro	RPDO	🖫
	Default Value	Minimum	Maximum	Unit
	1000	0	30000	0.1rpm (Ex. 1000 = 100.0rpm)
	Description			
	This object indicates the configured maximal slippage. When the max slippage has been reached, the corresponding bit 13 max slippage error in the *Statusword* is set to 1. The value is given in 0.1rpm.			

6.5.36 60FF$_h$ Target velocity

Index	Name			Mnemonic
60FF$_h$	*Target_velocity*			
	Data Type	Access Type	PDO Mapping	Note
	i32	rw	RPDO	🖫
	Default Value	Minimum	Maximum	Unit
				0.1rpm (Ex. 1000 = 100.0rpm)
	Description			
	This object indicates the configured target velocity and shall be used as input for the trajectory generator. The value is given in 0.1rpm.			

6.5.37 6502_h Supported drive modes

Index	Name			Mnemonic
6502_h	*Supported_drive_modes*			
	Data Type	Access Type	PDO Mapping	Note
	u32	ro	TPDO	
	Default Value	Minimum	Maximum	Unit

Description

This object provides information about the supported drive modes.

It is structured in bits and each bit is associated with a specific mode. If the mode is supported the bit is 1, otherwise the bit is 0 if the drive mode is not supported by the device.

Bit	Description	
31	r	Reserved
…	r	Reserved
16	r	Reserved

Bit	Description		Bit	Description	
15	r	Reserved	7	csp	Cyclic synchronous profile mode
14	r	Reserved	6	ip	Interpolated position mode
13	r	Reserved	5	hm	Homing mode
12	r	Reserved	4	r	Reserved
11	r	Reserved	3	tq	Torque mode
10	cstca	Cyclic synchronous torque mode with commutation angle	2	pv	Profile velocity mode
9	cst	Cyclic synchronous torque mode	1	vl	Velocity mode
8	csv	Cyclic synchronous velocity mode	0	pp	Profile position mode

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10 cstca	9 cst	8 csv	7 csp	6 ip	5 hm	4	3 tq	2 pv	1 vl	0 pp

6.5.38 67FE$_h$ Version number

Index	Name			Mnemonic
67FE$_h$	*Version_number*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw	RPDO	
	Default Value	Minimum	Maximum	Unit
	0			

Description

The object supplies the version of the CiA 402 profile implemented in the drive.

bits 31..24	bits 23..16	bits 15..8	bits 7..0
Reserved	Main version	Secondary version	Sub version

7 Status and Diagnostics

In the Omni Automation IDE, 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 *Temperature* shows the power stage temperature value of the drive.

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

Under the section *Analog Input,* the field *Input 0* shows the voltage value applied to the analog input.

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

The field Position shows the actual position reached by the motor. The integer part of the value indicates the number of complete revolution, while the decimal part shows the fraction of the revolution reached by the motor with a resolution of 1/10000 rev. For example, the value 0.5000 indicates that the motor is half revolution forward with respect to 0 position, while the value -3.7500 indicates that the motor is 3 and 3 quarters of revolution backward with respect to the 0 position. Clicking with the right button of the mouse on the field you can reset the displayed value. This action does not change the physical position of the motor but simply reset the displayed value.

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 reached the set rated value.

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

Under the section *Motor Encoder,* the field *Value* indicates the cyclical position of the encoder 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 I 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 and the errors in the memory or active. When the error is active, the column *Active* of the table contains an exclamation mark, while when the error is stored but no more active the column *Active* is empty and the line background is red. When the error reset is executed the background becomes white. In the table are stored up to the latest 10 errors, then the latest replace the oldest. The column *Time* shows the moment when the error occurred after the power on of the 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.

LAM Technologies
Viale Ludovico Ariosto, 492/D
50019 Sesto Fiorentino
Firenze - ITALY

Ph. +39 055 4207746
Fax +39 055 4207651

www.lamtechnologies.com

General information info@lamtechnologies.com
Technical support support@lamtechnologies.com
Sales information sales@lamtechnologies.com

rev. 0.23 (preliminary)