Drives & Motors

CANopen Manual

Vector Step Positioners

CAN connected

DS-301 V4.01 - DSP-402 V1.1

Index

NOTES ON SAFETY

Products for automation manufactured by AEC must be handled, installed and maintained only by skilled and authorized personnel, that must be qualified and instructed to install components for automation. Devices must be installed only for the purposes described in the user's guide. The installer should pay particular attention to potential risks caused by mechanical and electrical hazards.

It is very important that all applications and installations meet all applicable safety requirements.

The installers must take responsibility to verify their knowledge and understanding of all applicable safety standards.

Installations which are not complying with safety requirements can damage equipment and injure the user.

AEC s.r.l. will not be liable and will not take any responsability for damages caused by products handled or installed improperly, or if the customer have given permission or performed modifications and/or repairs not authorized from AEC s.r.l.

AEC's motion control equipment are high-performances devices for automation, able to producing high forces and rapid movements.

Pay high attention, in particular during installation and development of applications.

Use properly sized equipments for the type of application.

AEC's devices must be considered as components for automation. They are sold as end-user products, and must be installed only by qualified personnel, in accordance with all applicable safety requirements.

Skilled staff must be able to recognize possible dangers that may result from programming, modifying parameter's values and, generally, that may result from using mechanical, electric and electronic equipment.

The drive must be installed in closed cabinets, so that any parts thereof is not reachable while system is powered on.

AEC s.r.l strongly recommends to always follow safety requirements and security rules. Failure to follow this instruction may cause and/or injuries.

General precautions

- The images contained in this manual are for demonstration purposes, and may differ from the products received.
- This manual is subject to changes due to improvement of the products, modification of specifications, or manual thereof improvement.
- AEC s.r.l. is not responsible for any damage to property or injury that could result from improper installation and/or not authorized modification to products.

AEC's drive systems are products for general use that conform to the state of the art in technology and are designed to prevent any dangers. However, drives and drive controllers that are not specifically designed for safety functions are not approved for applications where the functioning of the drive could endanger persons. ***The possibility of unexpected or unbraked movements can never be totally excluded without additional safety equipment****. For this reason personnel must never be in the danger zone of the drives unless additional suitable safety equipment prevents any personal danger. This applies to operation of the machine during production and also to all service and maintenance work on drives and the machine. The machine design must ensure personal safety. Suitable measures for prevention of property damage are also required.*

To prevent personal injury and damage to property, damaged drive systems must not be installed. Changes and modifications of the drive systems are not permitted, and if made all no warranty and liability will be accepted.

REVISIONS

Manuale_Canopen_EN_rev08

Version	Date	Notes
1.0	July 2009	First draft
1.2	02 October 2009	Addition of AEC proprietary objects
1.3	3 February 2010	Update of proprietary objects mapping
1.3.3	8 August 2012	Update of Data Type on Object 0x2052
1.3.1 Rev. 02	17 October 2013	Addition of register 0x6085
1.3.2	May 2014	Addition of object 0x23E8
1.3.3	October 2014	Update of registers section
V02	March 2017	General update
rev.03	May 2018	Update of registers section
rev.04	March 2019	Update of AEC registers section
rev.05	June 2019	Addition of AEC variables management
rev.06	August 2019	Update of registers 0079-80, 0358, from 0544 to 0558
rev. 07	July 2020	Update of AEC registers section
rev. 08	June 2023	Addition of AEC registers with address higher than 10.000

COPYRIGHTS

TRADEMARKS

GENERAL NOTES ON THE PROTOCOL

CAN (Controller Area Network) fieldbus had been originally developped for the automotive market, with the aim of reducing the complexity of the connections needed to put into communication the various electronic devices present in a normal car (ABS, Airbag, SRS etc).

Thanks to its characteristics, like easy wiring and reliability, it was increasigly used in the industrial sector to control complex machineries with distributed intelligence.

CAN must not be considered a replacement for big control networks like Ethernet, but at local level, it's increasignly used inside machineries, thanks to its speed and reliability.

The capability to work in real-time contributed to the quick diffusion of the protocol (communication networks are considered real-time with a baudrate higher than 200Kbit/s).

With the purposes to standardize the interfacement mode between the various devices, and to make simple the use of CAN fieldbus to the user, an organization named Cia has been established. The Cia (Can in automation, website www.can-cia.de) is responsible for the definition of the communication standards. Between several standards, the standard communication protocol named CANopen has been defined.This standard is explained in the Draft Standard DS301 for what it concerns the part common to all the devices that belong to CANopen world.

Besides the DS301, the devices defined as CANopen compatible must have a further particularization, depending on the segment of devices to which they belong, in order to conform the front-end of the communication on the fieldbus side.

These specializations are called "Device Profile", and they are defined in the DS4xx drafts.

As an example, DS401 for I/O modules, DS402 for motion control devices (drives), etc.

Moreover, a CANopen compatible device must comly with the defined type of connectors and their pinouts.

All this gives a significant advantage to the end user, that can switch between a type of drive to another, being sure that nothing will change for what concerns the CAN communication.

The CAN bus, as previously exlained, can support up to 127 nodes. The maximum distance without the need of signal repetitors is 500 meters.

The transmission speed can be selected from 20KHz to 1MHz.

Each CAN message is composed by data of 8 bytes.

CANopen protocol is responsible for the segmentation of information longer than 8 bytes, and the management of the interface of the device toward the CAN world.

The level 1 of the OSI levels is fully managed via hardware from the CAN control device, that must comply to the specifications defined by Bosch, which has defined the CAN fieldbus.

Therefore, it's impossible for anybdy to make proprietary modifications at this level. This brings to the user the significant advantage of the possibility to insert a device into a CAN network, being sure that this will not disturb the network.

CANopen protocol is at the level 7 of the OSI scale, and has the task to manage the incoming and outcoming messages to the CAN network.

In order to comply as closely as possible the Cia specifications, the application layer of the OSI communication model (DS402) of the AEC devices, has been developped internally from AEC s.r.l., while the data transfer layer (DS301) has been supplied from Vector Informatik, GMBH.

The comminication objects (defined by CAN open standard) for the configuration of COB-ID, SDO, PDO are described by the DS301 standard of Cia.

The software supplied by Vector provides sone key features that are called by the main management program of AEC devices. The device is seen as a SLAVE into a network defined Minimum Device.

The standard for the Motion Control device profile (DS402) has been followed at the application level to configure the operationg mode and, where possible, to supply several other proprietary commands.

All the MANDATORY objects needed for the stepper motor movement has been implemented.

All the other objects has been implemented only if strictly necessary.

Data exchange principles

At the moment of a transmission of the CAN data, the content of a communication is not addressed to specific stations, but it's marked by an inequivocable identificator in all the network. In addition to the contained message, the markidentifies the priority of the message.
This is very important for the assignment of the bus, when more than one stations compete for the right to access to the bus itself.
When the master intends to send a message to any station, or more stations, it transmits the data along with the identifier and the request of transmission to the CAN module.
This latter has the task to build the message and to send it to the line, soon as it is free.
When the message is inserted into the network, all other nodes become recipient of such message. It will be the task of each node to evaluate whether the message is relevant or not (if it is enabled to receive and to interpret the message)
In case of irrelevant messages, the messages are ignored.
The possibility to send messages identified only in the content and not in the destination, makes the system and the configuration very flexible, in addition to give the possibility to easily send the same information to several nodes simultaneously.

Arbitrage without distribution, bit per bit

In case of real-time communications, it's of primary importance that the data are transmitted quickly.

This imply not only a fast physical transmissione (e.g. 1Mbits), but also a fast assignment of the bus.

The priority with which a message is transmitted before another is defined by its identifier.

The priorities are defined in the phase of assignment of the nodes, and they are not dinamically editable.

To each node it is assigned an ID (identifier) corresponding to a binary number; the identifier with the lower number has the higher priority.

The access conflict is resolved with a bit per bit arbitrage; all the nodes that must access to the bus send the request of transmission with the identifier of the message to be sent.

In accordance with the "wired-and" mechanism, according to which the dominant state (0) overwrite the recessive (1), it's easy to comprehend that the ID with a lower value overwrites all the others in line, taking the priority.

All the "losers" become recipient of the message, awaiting to access to the bus as soon as it's free.

This type of arbitrage guarantees the transmission of important messages, also in case of overloaded bus.

Recognition and reporting of the errors

Unlike other protocols, CAN does not use confirm messages, but it reports possible errors occured.
To recognize the errors, the CAN implements three mechanisms at message level.

- *Cyclic Redundancy Check:* the sending device applies redundant control bits to the transmission, adding a checksum value calculated on the bits to be send. The same calculation is repeated by the receiver device on the basis of the bits received. If the checksums don't mach, a CRC error is detected.
- *Frame Check:* this mechanism checks that the structure of the transmitted block and the length of the same are correct. Frame check errors are called format errors.
- *ACK errors:* the incoming block are confirmed by all the recipients with a positive

acknowledge. If the transmitter doest not detect any acknowledhe (ACK error), this means thant there might have been a transmission error recognized only by the recipients, that the ACK field is falsified, or that the recipient is nor present.

In CAN protocol, there are also two implemented mechanism for the detection of errors at bit level:

- *Monitoring:* this is the capability of the maker of the message to recognize errors by basing on the control of the signals in the bus: each node whic transmits, simultanously monitors the status of the bus, recognizing in real-time the divergences between the sent bit and the received bit. In this way, it is possible to detect local or global error.
- *Bit-stuffing:* At bit level, the codingof the single bits itself is checked. For the representation of the bits, CAN protocol uses the NRZ coding (Non-return-to-zero), which guarantees the maximum efficiency during the coding. The stuffing control has the task to check the codes according to the bit-stuffing rule: after 5 bits with the same value, a stuff-bit with the oppo-site value is sent. This bit will be immediately discarded by the recipient. If bit-stuff errors are detected, the transmission in progress is interrupted ans a “error flag” is trasnmitted, thus avoiding that the message would be interpreted and accepted by other nodes. After the interruption of the transmission, there will be a new trasmission of the message, starting from the arbitrage phase. In case of a defective station, this method might result in blocking all the messages (also the corrected ones) and the blocking of the bus. Therefore, CAN proceeds to distinguish between transient or permanent erros. In the second scenario, CAN disables the node itself.

SECURITY OF THE DATA

In relation to bus systems, the term “security of the data” means the characteristics of identifying and recognizing falsified data from transmission errors.
The probability of residual error is a statistic dimension for the security violation of the data.
It indicates, in fact, the probability with which the data might be falsified and not being identified as wrong.
The probability of residual errors should be so low not to leave unrecognized falsified data during all the average lifespan of the system
In a system with 10 nodes, and an error rate of 1/1000 (1 disturbed message every 1000), the re-sidual error probability of the CAN network can be determined in relation to the probability of a bit error for messages with a length of 80/90 bitsassuming a bit error probability of approximately 0,2, the probability of residual error has a maximum value that is about 10-13.
Starting from this maximum value, it is possible to calculate the maximum number of unrecognizable errors for a determined CAN network.
As an exemple, if a CAN network has a communication speed of 1Mbit/s, an average solicitation of the bus of 50%, operationg duration of 4000 hours and average length of messages of 80bits, the approximate number of transmitted messages would be $9x10^4$.
The statistic number of unrecognized errors during the total operating time is therefore about 10^{-2} .
This means that in a management time of 8 hours a day, 365 days a year, and an error rate o 0,7, in average there is an unrecognized error every 1000 years.

PHYSICAL LAYER

The physical support for CAN protocol is a differential two-wires line with common return, as determined in the ISO11898 specifications for the high-speed transmissions.
Bitrates and bus length estimation (worst case)
High communication speeds decrease the immunity to the network disturbs, therefore also the length of the same must be reduced.
The following chart shows the estimated maximum length in the worst operating conditions, on the basis of a propagation delay of 5ns/m and a total propagation delay of the device (device internal in/out delay) of:

1 Mbit/s - 800 Kbit/s	:	210 ns
500 Kbit/s - 250 Kbit/s	:	300 ns(2*40 ns for the opto included)
125 Kbit/s	:	450 ns(2*100 ns for the opto included)

Bit rate (Kbit/s)	Bus length (m)
1000	25
800	50
500	100
250	250
125	500
50	1000
20	2500
10	5000

For line length over 200m, it ia advisable to use optocouplers, while it may be necessary to use bridges or repetitors for line lengths of over 1Km.

In order to reduce the line reflection, terminate the both sides of the communication bus with a resistance having a value equal to 120 Ohm. The possible shield must be connected to the ground, only in one side, in a single point.

General characteristics

CANopen characteristics

Following you can find the most important features regarding CANopen implementation:

Standards	ISO 11898 (1-2)
Physical layer	CAN bus Ver. 2.0
Bus topology	NRZ differential (Non Return to Zero)
Cable type	Shielded twisted pairs conductors
Connector	Male D-Sub 9 pin
Number of nodes on the bus	max. 127
NMT	Slave
Error Control	Node Guarding, Life Guarding, Heartbeat
Node ID	Hardware/Software
Nr. of PDOs	4 Rx - 4 Tx
PDO modes	Event triggered, Sync (cyclic), Sync (acyclic)
PDO linking	No
PDO mapping	Variable (granularity 8bit)
Nr. of SDOs	1 Server, 0 Client
Emergency Message	Yes
CANopen Version	DS301 V4.01
Framework	No
Device Profile	DSP-402 V1.1
AEC's Vendor ID	00 00 00 BC

Products characteristics

Name	Family	File EDS	Interface	Profile
SMD1104xxC	VectorStep	Smd1104xIC_24.eds	CAN bus	CANopen
SMD3006xxC	VectorStep	Smd3006xIC_24.eds	CAN bus	CANopen
SMD5006xxC	VectorStep	Smd5006xIC_24.eds	CAN bus	CANopen
SMD5106xxC	VectorStep	Smd5106xIC_24.eds	CAN bus	CANopen

SMD2204 MULTIAXIS DRIVE COMMUNICATION

For **multiaxis drives (MA from now on)** CANopen/EtherCAT communication is similar to the typical **single axis drive (SA from now on)**, and it is compliant with the CiA301 and CiA402 specifications

To command the axis 1 of a MA, the communication registers are exactly the same as for a SA: all the registers from H6000 to H607FF are used to control the axis, so much so that in case of a replacement of a SA with an MA, the full compatibility is guaranteed for the command of the axis 1 of the MA.

For a MA, to command the other two axes, it is sufficient to shift the registers of the location H6000..H67FF of H0800 for the axis 2, and of H1000 for the axis 3.

As an example, the classic register H6060 (Mode of operation) for the axis 1 simply becomes H6860 for the axis 2 and H7060 for the axis 3.

Another example is the H6040 (Controlword) for the axis 1, which becomes H6840 for the axis 2 and H7040 for the axis 3.

Examples:

603F	Error Code (Axis1)	683F	Error Code (Axis2)	703F	Error Code (Axis3)
6040	Controlword (Axis1)	6840	Controlword (Axis2)	7040	Controlword (Axis3)
6041	Statusword (Axis1)	6841	Statusword (Axis2)	7041	Statusword (Axis3)
605A	Quick stop option code (Axis1)	685A	Quick stop option code (Axis2)	705A	Quick stop option code (Axis3)
605B	Shutdown option code (Axis1)	685B	Shutdown option code (Axis2)	705B	Shutdown option code (Axis3)
605C	Disable operation option code (Axis1)	685C	Disable operation option code (Axis2)	705C	Disable operation option code (Axis3)
605D	Stop option code (Axis1)	685D	Stop option code (Axis2)	705D	Stop option code (Axis3)
605E	Fault reaction option code (Axis1)	685E	Fault reaction option code (Axis2)	705E	Fault reaction option code (Axis3)
6060	Modes of operation (Axis1)	6860	Modes of operation (Axis2)	7060	Modes of operation (Axis3)
6061	Modes of operation display (Axis1)	6861	Modes of operation display (Axis2)	7061	Modes of operation display (Axis3)
6064	Position actual value (Axis1)	6864	Position actual value (Axis2)	7064	Position actual value (Axis3)
6065	Following error window (Axis1)	6865	Following error window (Axis2)	7065	Following error window (Axis3)
6066	Following error timeout (Axis1)	6866	Following error timeout (Axis2)	7066	Following error timeout (Axis3)
6067	Position window (Axis1)	6867	Position window (Axis2)	7067	Position window (Axis3)
6068	Position window time-out (Axis1)	6868	Position window time-out (Axis2)	7068	Position window time-out (Axis3)
6069	VelocitySensorActualValue (Axis1)	6869	VelocitySensorActualValue (Axis2)	7069	VelocitySensorActualValue (Axis3)
606C	Velocity Actual Value (Axis1)	686C	Velocity Actual Value (Axis2)	706C	Velocity Actual Value (Axis3)
6071	Target torque (Axis1)	6871	Target torque (Axis2)	7071	Target torque (Axis3)

For the specific AEC registers, namely for the zone H2000…H5FFF, instead, the philosophy changes.

For a **SA**

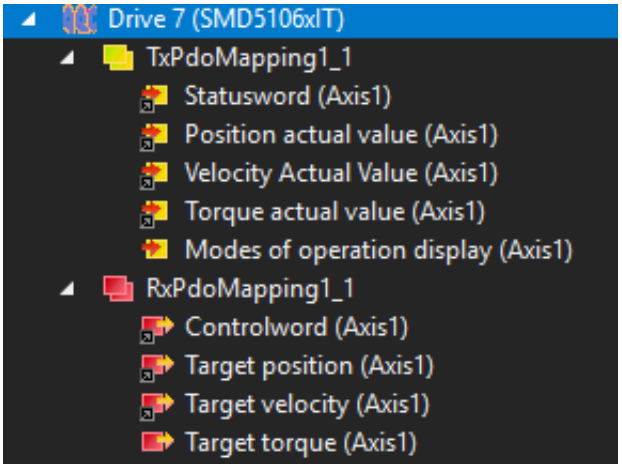
2000:0	Posact		> 1 <
2000:01	Axis 1 Posact	RW P	3849
2004:0	Posactreq		> 1 <
2004:01	Axis 1 Posactreq	RW P	3791
2008:0	Postarg		> 1 <
2008:01	Axis 1 Postarg	RW P	3634

For a **MA** becomes

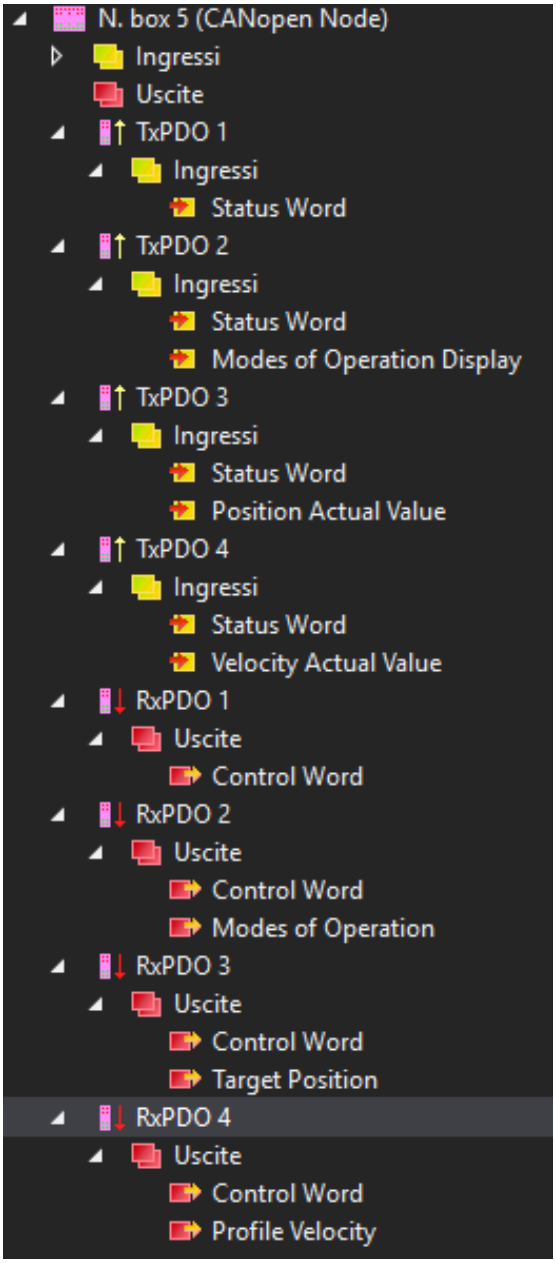
2000:0	Posact		> 3 <
2000:01	Axis 1 Posact	RW P	0
2000:02	Axis 2 Posact	RW P	0
2000:03	Axis 3 Posact	RW P	0
2004:0	Posactreq		> 3 <
2004:01	Axis 1 Posactreq	RW P	0
2004:02	Axis 2 Posactreq	RW P	0
2004:03	Axis 3 Posactreq	RW P	0
2008:0	Postarg		> 3 <
2008:01	Axis 1 Postarg	RW P	0
2008:02	Axis 2 Postarg	RW P	0
2008:03	Axis 3 Postarg	RW P	0

Similarly, for the default PDOTX and PDORX, for a **SA** you have:

EtherCAT

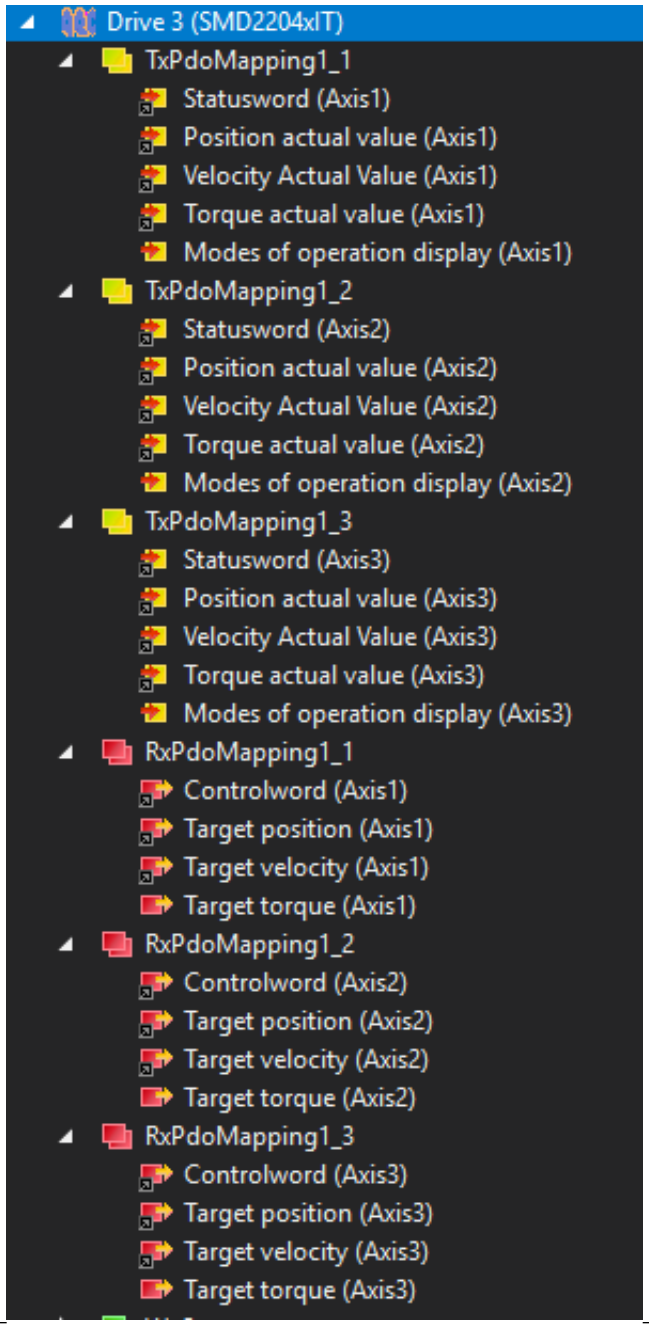


CANopen

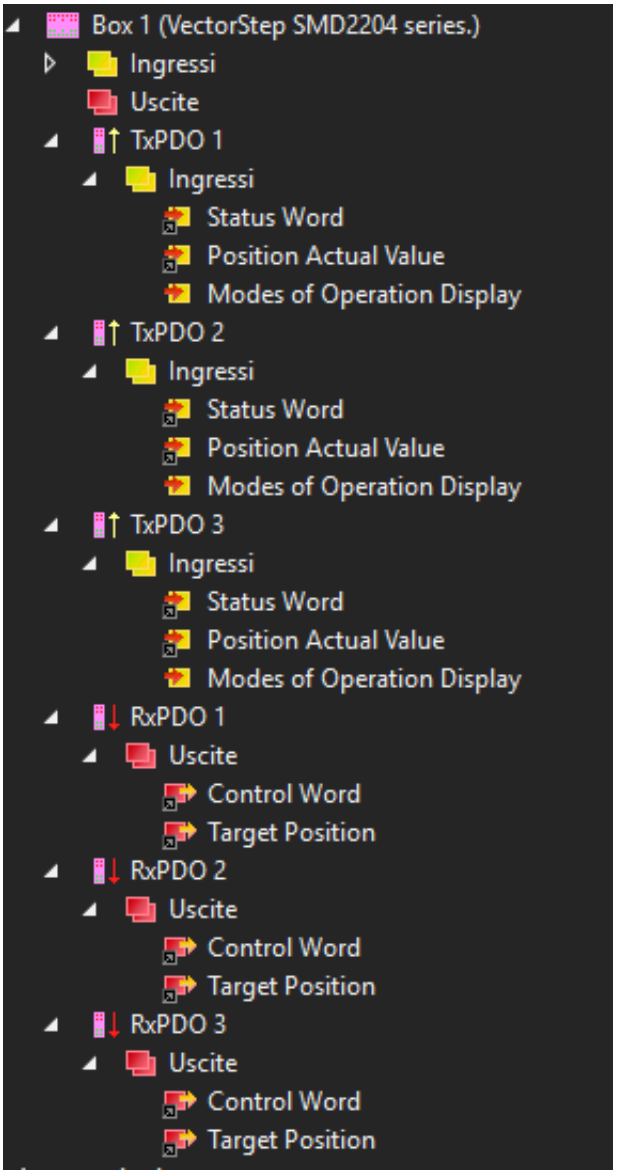


instead, for a **MA**

EtherCAT



CANopen



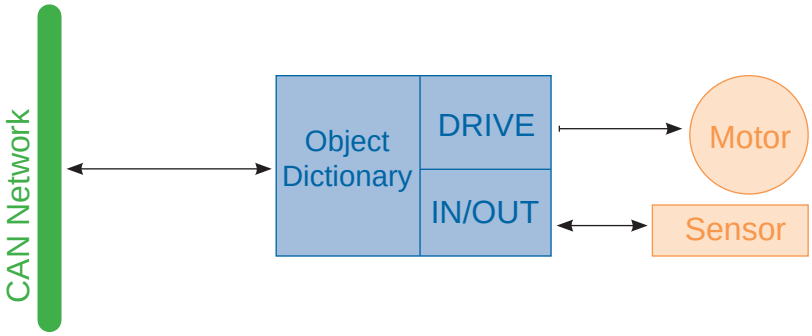
Definition of a device

Objects and dictionary

The main purpose of a CANopen communication network is to read and write status information from a connected device.

For this reason, every device connected into the network defines some parameter lists to which is possible to access in read and/or in write; this list of information is called dictionary, and each element is called Object.

Each device connected into a CANopen network must define an Object dictionary that indicates which information can be exchanged.



Dictionary layout

As already stated, the dictionary is a table which contains all the objects supported by the device. Each of the objects is identified by a 16 bit index and a 8 bit sub-index.

Most of the objects contains simple data, 16 bit integer, 32 bit integer or strings, to which is possible to access directly by using the 16 bit index.

Some objects, instead, contain a subgroup of data (structure); in this case, it is necessary to use the index to access to the object, and the sub-index to define the data to be read and/or to be written.

The layout of the dictionary is defined by the CANopen profile (DSP402), as follows:

Index range	Objects description
0000	Not used
0001 - 001F	Static Data types
0020 - 003F	Complex Data Types
0040 - 005F	Manufacturer Specific Complex Data Types
0060 - 007F	Device Profile Specific Static Data Types
0080 - 009F	Device Profile Specific Complex Data Types
00A0 - 0FFF	Reserved for future use
1000 - 1FFF	Communication Profile Area (DS 301)
2000 - 5FFF	Manufacturer Specific Profile Area
6000 - 9FFF	Standardized Device Profile Area
A000 - FFFF	Reserved for further use

ACCESS TO THE DICTIONARY

TYPES OF MESSAGES

CANopen protocol uses two types of messages to access to the connected devices:

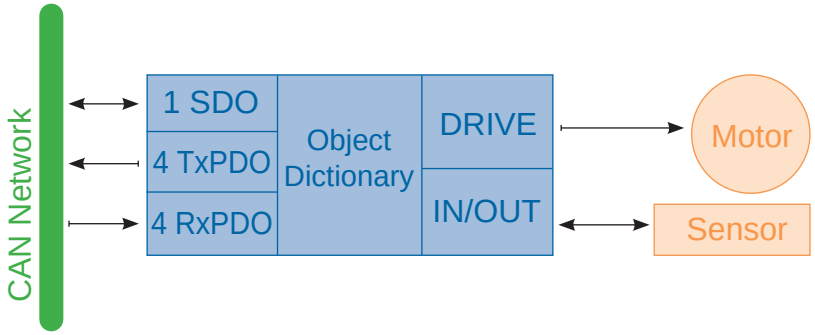
- Service Data Objects (SDO)
- Process Data object (PDO)

Both can be used to read and write information into the CAN objects.

SDO	PDO
The SDO protocol permits to access to any object defined by the object dictionary, independently from its type or dimension.	A PDO message can transfer a maximum of 8 byte data.
The transfer is always confirmed.	The transfer of a PDO does not provide confirmation.
It allows to access directly and without limitations to any object of the dictionary..	The access of an object using a PDO requires a preliminar parameterization (mapping), where it's indicated the number of parameters transferred, the type of data and the reference object/objects.
It uses a client/server model in which the master is the only device that can access to the object dictionary of the connected devices.	It uses a peer-to-peer communication model, where each node can start a communication, and more than one node can simoultaneously reveice the same message.
A SDO has two identifiers: one in transmission (Slave -> Master) and one in receiving. (Master -> Slave)	The PDO in transmission are used by the device to send data. The PDO in receiving are used to receive data.
It is used to configure the device, to map PDO and to map sporadic communications or low priority communications between Master and Slave.	It is used to trnasfer high priority data (like setpoints, velocity or actual position) or all the frequent and synchronous communications.

SDO AND PDO

The VectorStep drives implement 1 SDO and 8 PDOs (4 in transmission and 4 in receiving).



How to map a PDO

Each PDO is defined by two objects of the dictionary of the device:

- *PDO Communication Object:* defines the identifier of the PDO (PDO-ID), the communication mode (synchronous, asynchronous) and the type of trigger (event driver or cyclic).
- *PDO Mapping Object:* associates the data bytes with one or more objects of the dictionary.

The map a PDO means to configure the PDO Communication Object and the PDO Mapping Object.

The following table provide the procedure to map, or re-map, a new PDO.

	Operations	Descriptions	Example
1	Disable the PDO	To disable the PDO, write at 0 the sub-index zero of the mapping object.	0x1601:00 = 0
2	Set the communication parameters	If necessary, define the new identifier(COB-ID) into the sub-index 1 of the PDO Communication Object. Choose the type of transmission by writing the sub-index 2 of the same object ([0-240] define a synchronous transmission, [254,255] define an asynchronous transmission.)	0x1401:01 = 0x181 0x1401:02 = 0xFE
3	Map the data	Specify the object/objects to be accessed, and the type of data transferred, into the sub-indexes from 1 to 4 of the PDO Mapping Object. • The bits [0..7], of the mapping value, define the dimension (in bit) of the transferred data. • the bits [8..15] contain the sub-index of the mapped object. • the bits [16..31] contain the index of the mapped object.. It is possible to map up to 4 objects into the same PDO.	Sub-index 1 = Controlword 0x1601:01 = 0x604000010 Sub-index 2 = Modes of Operation 0x1601:02 = 0x60600008
4	Set the number of mapped objects	Write the number of mapped objects in the sub-index 0 of the PDO Mapping Object. This operation re-enable the PDO.	0x1601:00 = 02

The default COB-ID are:

Object	COB-ID		PDO Communication Object
	Min (Node 1)	Max (node 127)	
TxPDO1	0x181	0x1FF	0x1800
RxPDO1	0x201	0x27F	0x1400
TxPDO2	0x281	0x2FF	0x1801
RxPDO2	0x301	0x37F	0x1401
TxPDO3	0x381	0x3FF	0x1802
RxPDO3	0x401	0x47F	0x1402
TxPDO4	0x481	0x4FF	0x1803
RxPDO4	0x501	0x57F	0x1403
SDO	0x601	0x67F	--

DS-301 V4.1 Objects

Summary table

Object	Object Name	Sub-index	Data type	Attributes
0x1000	Device Type	0	Unsigned 32	RO
0x1001	Error Register	0	Unsigned 8	RO / MAP
0x1003	Pre-defined Error Field	3	Unsigned 32 Array	Sub0 = RW Sub1-2 = RO
0x1005	COB-ID SYNC Message	0	Unsigned 32	RW
0x1006	Communication Cycle Period	0	Unsigned 32	RW
0x1007	Synchronous window length	0	Unsigned 32	RW
0x1008	Manufacturer Device Name	0	Visible String	RO
0x1009	Manufacturer Hardware Version	0	Visible String	RO
0x100A	Manufacturer Software Version	0	Visible String	RO
0x100C	Guard time	0	Unsigned 16	RW
0x100D	Life-time factor	0	Unsigned 8	RW
0x1014	COB-ID Emergency Object	0	Unsigned 32	RW
0x1015	Inhibit Time Emergency	0	Unsigned 16	RW
0x1016	Consumer Heartbeat Time	3	Unsigned 32 Array	RW
0x1017	Producer Heartbeat Time	0	Unsigned 16	RW
0x1018	Identity Object	4	Unsigned 32 Array	RO
0x1200	SDO1 Parameter	3	Record	RW
0x1400	RxPDO1 Communication Parameters	5	Record	RW
0x1401	RxPDO2 Communication Parameters	5	Record	RW
0x1402	RxPDO3 Communication Parameters	5	Record	RW
0x1403	RxPDO4 Communication Parameters	5	Record	RW
0x1600	RxPDO1 Mapping Parameters	5	Unsigned 32 Array	RW
0x1601	RxPDO2 Mapping Parameters	9	Unsigned 32 Array	RW
0x1602	RxPDO3 Mapping Parameters	9	Unsigned 32 Array	RW
0x1603	RxPDO4 Mapping Parameters	9	Unsigned 32 Array	RW
0x1800	TxPDO1 Communication Parameters	9	Record	RW
0x1801	TxPDO2 Communication Parameters	6	Record	RW
0x1802	TxPDO3 Communication Parameters	6	Record	RW
0x1803	TxPDO4 Communication Parameters	6	Record	RW
0x1A00	TxPDO1 Mapping Parameters	9	Unsigned 32 Array	RW
0x1A01	TxPDO2 Mapping Parameters	9	Unsigned 32 Array	RW
0x1A02	TxPDO3 Mapping Parameters	9	Unsigned 32 Array	RW
0x1A03	TxPDO4 Mapping Parameters	9	Unsigned 32 Array	RW

Device Type 0x1000

This object contains the information about the type of device an its features. It is composed by 2 fields of 16 bits. The 16 low bits bassi include information regarding the device profile, the 16 high bits provide additional information.

Index	0x1000				
Symbol	device_type	**Length (byte)**	4	**Min value**	0x000000000
Object Code	Variable	**Elements**		**Max value**	0xFFFFFFFF
Data Type	Unsigned32	**Access**	Read only	**Default value**	0x00040192
		PDO mapping	No		

Details:

Bit	Description	Code
[0..15]	Device Profile Number	DS402
[16..23]	Drive Type bit encoding	Stepper Motor
[24..32]	Manufacturer Specific	0

Error Register 0x1001

This object is the error register of the device. The device can map the internal errors into this byte.

Index	0x1001				
Symbol	error_register	**Length (byte)**	1	**Min value**	0x00
Object Code	Variable	**Elements**		**Max value**	0xFF
Data Type	Unsigned8	**Access**	Read only	**Default value**	0x00
		PDO mapping	Yes		

Details:

Bit	Description	Code
0	Generic error	-
1	Non implemented	-
2	Non implemented	-
3	Non implemented	-
4	Communication error (overrun, error state)	-
5	Device Profile error	-
6	Reserved (always 0)	-
7	Non implemented	-

Pre-defined Error Field 0x1003

This object holds the errors that occurred in the device and which have been signaled by the emergency object (EMCY), in order to provide an alarm log.
The VectorStep drives are able to store up to 2 errors. Further errors will overwrite the current ones.
By writing 0 in 0x1003:0, the log will be reset.

Index	0x1003				
Symbol	pre_defined_error_field	**Length (byte)**		**Min value**	
Object Code	Array	**Elements**	3 max	**Max value**	
Data Type		**Access**		**Default value**	
		PDO mapping			

Sub-indexes:

Index	0x1003:00				
Symbol	number_of_errors	**Length (byte)**	1	**Min value**	0x00
Object Code	Variable	**Elements**		**Max value**	0x02
Data Type	Unsigned8	**Access**	Read/Write	**Default value**	0x00
		PDO mapping	No		

Index	0x1003:01				
Symbol	standard_error_field	**Length (byte)**	4	**Min value**	0x00000000
Object Code	Variable	**Elements**		**Max value**	0xFFFFFFFF
Data Type	Unsigned32	**Access**	Read	**Default value**	0x00000000
		PDO mapping	No		

Index	0x1003:02				
Symbol	standard_error_field	**Length (byte)**	4	**Min value**	0x00000000
Object Code	Variable	**Elements**		**Max value**	0xFFFFFFFF
Data Type	Unsigned32	**Access**	Read	**Default value**	0x00000000
		PDO mapping	No		

Each EMCY has a 5 byte Vendor Specific Data that can be used to pass specific information to the Master. The zone HFFxx is specific for the manufacturer. The possible errors generated by the device are listed below:

Code	Description
0x0000	Generic error
0x5400	Error in the power stage of the drive
0x8110	Overload of the bus (lost message)
0x8120	CAN controller in “Error passve” state (CAN controller disabled)
0x8130	Error in message control
0x8140	Exit from the Bus state off
0x8150	Collision COB-ID
0x8210	Length of the PDO greater than the mapped value
0x8220	Length of the PDO lower than the mapped value
FF01 + E.Cod	Reading/writing error into the NVRAM memory + Error code (Reepsts)
FF02	Error in executing of the HOME command (for SMD2204 - axis 1). The VSD[0] indicates the least significant byte of the register Rhsts

Codice	Descrizione
FF06	Error in executing of the HOME command (for SMD2204 - axis 2 if present). The VSD[0] indicates the least significant byte of the register Rhsts
FF07	Error in executing of the HOME command (for SMD2204 - axis 3 if present). The VSD[0] indicates the least significant byte of the register Rhsts
FF03	Lack of data to be processed in interpolation mode. VSD[0]==1, VSD[1]==0 fixed.
FF04	Interpolation alarm: Sync too early. VSD[0]==1, VSD[1]==0 fixed.
FF05	Interpolation alarm: Sync too late. VSD[0]==1, VSD[1]==0 fixed.
FF13 FF14 FF15	Drive power stage error for axes 1, 2 and 3, present only for multiaxis drive (SMD2204). This EMCY is generated in addition to the standard 0x5400 in order to give detailed information for each axis.


Sub-indexes 01 and 02 are present only in case errors occurr.

COB-ID SYNC Message 0x1005

The 0x1005 object defines the COB-ID of the synchronization object (SYNC).
It also defines if the device generates the SYNC message.

Index	0x1005				
Symbol	COD_ID_SYNC	**Length (byte)**	4	**Min value**	0X00000000
Object Code	Variable	**Elements**		**Max value**	0xFFFFFFFF
Data Type	Unsigned32	**Access**	Read/Write	**Default value**	0x80 or 0x80000080
		PDO mapping	No		

Details:

Bit	Value	Code
31	X	Not relevant
30	0	The device doesn't generate the SYNC messagge
	1	The device generates the SYNC messagge
29	0	11-bit ID (CAN 2.0A)
	1	29-bit ID (Full CAN 2.0B)
[28..11]	0	if bit29 = 0 then bit [28..11] are ignored
	x	if bit29 = 1 then bit [28..11] become relevant
[10..0]	x	SYNC COB-ID

If you attempt to modify the bits 29 and 30, the device replies with an abort message (abort code: 0x06090030).

Communication cycle period 0x1006

This object defi nes the communication time in microseconds, which defines the SYNC interval. It is set at 0 if not being used. If a new value different from 0 is inserted in the communication cycle period, the SYNC object transmission restarts within a sync cycle of the new value.

Index	0x1006				
Symbol	Communication_cycle_period	**Length (byte)**	4	**Min value**	0X00000000
Object Code	Variable	**Elements**		**Max value**	0xFFFFFFFF
Data Type	Unsigned32	**Access**	Read/Write	**Default value**	0X00000000
		PDO mapping	No		

Synchronous window length 0x1007

This object contains the length of the temporal window in microseconds for the synchronous PDOs. It is set at 0 if not being used.

Index	0x1007				
Symbol	Synchronous_window_length	**Length (byte)**	4	**Min value**	0X00000000
Object Code	Variable	**Elements**		**Max value**	0xFFFFFFFF
Data Type	Unsigned32	**Access**	Read/Write	**Default value**	0X00000000
		PDO mapping	No		

Manufacturer device name 0x1008

This object holds the name of the device.

Index	0x1008				
Symbol	Manufacturer_device_name	**Length (byte)**	25	**Min value**	
Object Code	Const	**Elements**		**Max value**	
Data Type	Visible string	**Access**	Read	**Default value**	SMD50.06LIC
		PDO mapping	No		

Manufacturer hardware version 0x1009

This object contains the hardware version of the connected device.

Index	0x1009				
Symbol	Manufacturer_hardware_version	**Length (byte)**	11	**Min value**	
Object Code	Const	**Elements**		**Max value**	
Data Type	Visible string	**Access**	Read	**Default value**	Hw ver.X.XX
		PDO mapping	No		

Manufacturer software version 0x100A

This object contains the software version of the connected device.

Index	0x100A				
Symbol	Manufacturer_software_version	**Length (byte)**	11	**Min value**	
Object Code	Const	**Elements**		**Max value**	
Data Type	Visible string	**Access**	Read	**Default value**	Sw ver.X.XX
		PDO mapping	No		

Guard time 0x100C

The objects 0x100C and 0x100D defi ne respectively the guard_time in milliseconds and the life_time factor.

The life_time factor multiplied by guard_time gives life time for the Life Guarding Protocol.

This object is set at 0 if not being used.

Index	0x100C				
Symbol	Guard_time	**Length (byte)**	2	**Min value**	0X0000
Object Code	Variable	**Elements**		**Max value**	0xFFFF
Data Type	Unsigned16	**Access**	Read/Write	**Default value**	0X0000
		PDO mapping	No		

LIFE TIME FACTOR 0x100D

This object defines the multiplicative factor that, united to the guard time, gives the Life Time for the Node Guarding Protocol. This object is set at 0 if not being used.

Index	0x100D				
Symbol	Life_time_factor	**Length (byte)**	1	**Min value**	0X00
Object Code	Variable	**Elements**		**Max value**	0xFF
Data Type	Unsigned8	**Access**	Read/Write	**Default value**	0X00
		PDO mapping	No		

COB-ID EMERGENCY MESSAGE 0x1014

The object 0x1014defines the COB-ID of the emergency object (EMCY).

Index	0x1014				
Symbol	COD_ID_EMCY	**Length (byte)**	4	**Min value**	0X00000001
Object Code	Variable	**Elements**		**Max value**	0xFFFFFFFF
Data Type	Unsigned32	**Access**	Read/Write	**Default value**	0x80 + Node_ID
		PDO mapping	No		

Details:

Bit	Value	Code
[31..30]	X	Not relevant
29	0	11-bit ID (CAN 2.0A)
	1	29-bit ID (Full CAN 2.0B)
[28..11]	0	if bit29 = 0 then bit [28..11] are ignored
	x	if bit29 = 1 then bit [28..11] become relevant
[10..0]	x	EMCY COB-ID

If you attempt to modify the bits 29 and 30, the device replies with an abort message (abort code: 0x06090030).

INHIBIT TIME EMERGENCY 0x1015

The inhibition time of the emergency messagge can be adjusted by using this object. The time is indicated as a multiple of 100 microseconds.

Index	0x1015				
Symbol	Inhibit_time_emergency	**Length (byte)**	2	**Min value**	0X0000
Object Code	Variable	**Elements**		**Max value**	0xFFFF
Data Type	Unsigned16	**Access**	Read/Write	**Default value**	0X0000
		PDO mapping	No		

CONSUMER HEARTBEAT TIME 0x1016

The Consumer Hearbeat Time defines the Heartbeat cycle control time. The value set should be higher than the time set in the Hearbeat message generator. Monitoring starts after the first Heartbeat message has been received. The time is indicated in multiples of 1 millisecond.
If the field of Consumer Hearbeat Time is set at 0, the corresponding field is not used.

Index	0x1016				
Symbol	Consumer_heartbeat_time	**Length (byte)**		**Min value**	
Object Code	Array	**Elements**	3	**Max value**	
Data Type		**Access**		**Default value**	
		PDO mapping			

Sub-indexes:

Index	0x1016:00				
Symbol	number_of_entries	**Length (byte)**	1	**Min value**	0x00
Object Code	Variable	**Elements**		**Max value**	0x02
Data Type	Unsigned8	**Access**	Read	**Default value**	0x00
		PDO mapping	No		

Index	0x1016:01				
Symbol	Consumer_heartbeat_time	**Length (byte)**	4	**Min value**	0x00000000
Object Code	Variable	**Elements**		**Max value**	0x007FFFFF
Data Type	Unsigned32	**Access**	Read/Write	**Default value**	0x00000000
		PDO mapping	No		

Index	0x1016:02				
Symbol	Consumer_heartbeat_time	**Length (byte)**	4	**Min value**	0x00000000
Object Code	Variable	**Elements**		**Max value**	0x007FFFFF
Data Type	Unsigned32	**Access**	Read/Write	**Default value**	0x00000000
		PDO mapping	No		

Details:

Bit	Value	Description
[31..24]	0	Reserved
[23..16]	x	Node-ID
[15..0]	x	Heart beat time

If you attempt to configure more than one Consumer Heartbeat Time different from 0 for the same Node-ID, the device generates an abort message (code 0x06040043).

PRODUCER HEARTBEAT TIME 0x1017

It defines the cycle time for the generation of a heartbeat signal. The value of the object producer heartbeat time is set at 0 if the function is not used.
The time should be a multiple of 1 millisecond.

Index	0x1017				
Symbol	Producer_heartbeat_time	**Length (byte)**	2	**Min value**	0X0000
Object Code	Variable	**Elements**		**Max value**	0xFFFF
Data Type	Unsigned16	**Access**	Read/Write	**Default value**	0X0000
		PDO mapping	No		

The object at index 0x1018 contains general information about the device, as listed below:

- Vendor ID (sub-index 1) holds a single value for each manufacturer of devices with CANOpen protocol.
- Specific product code, defi ned by the manufacturer (sub-index 2).
- Specific revision code, defi ned by the manufacturer (sub-index 3). It holds a greater (MSW) and a lower (LSW) revision number. The greater revision number identifies a specific CANOpen behaviuor. If a function is added, the greater code is increased. The lower number identifies dfirmware changes, but without modifying the object dictionary.
- Serial number of the prodcut.

Index	0x1018				
Symbol	pre_defined_error_field	**Length (byte)**		**Min value**	
Object Code	Array	**Elements**	5	**Max value**	
Data Type		**Access**		**Default value**	
		PDO mapping			

Sub-indexes:

Index	0x1018:00				
Symbol	number_of_entries	**Length (byte)**	1	**Min value**	0x01
Object Code	Variable	**Elements**		**Max value**	0x04
Data Type	Unsigned8	**Access**	Read	**Default value**	0x04
		PDO mapping	No		

Index	0x1018:01				
Symbol	vendor_ID	**Length (byte)**	4	**Min value**	0x00000000
Object Code	Variable	**Elements**		**Max value**	0xFFFFFFFF
Data Type	Unsigned32	**Access**	Read	**Default value**	0x000000BC
		PDO mapping	No		

Index	0x1018:02				
Symbol	product_code	**Length (byte)**	4	**Min value**	0x00000000
Object Code	Variable	**Elements**		**Max value**	0xFFFFFFFF
Data Type	Unsigned32	**Access**	Read	**Default value**	0x0000138E or 0x00000BBE
		PDO mapping	No		

Index	0x1018:03				
Symbol	revision_number	**Length (byte)**	4	**Min value**	0x00000000
Object Code	Variable	**Elements**		**Max value**	0xFFFFFFFF
Data Type	Unsigned32	**Access**	Read	**Default value**	
		PDO mapping	No		

Index	0x1003:02				
Symbol	serial_number	**Length (byte)**	4	**Min value**	0x00000000
Object Code	Variable	**Elements**		**Max value**	0xFFFFFFFF
Data Type	Unsigned32	**Access**	Read	**Default value**	
		PDO mapping	No		

SDO1 Parameters 0x1200

This object contains the communication parameters for the SDOs supported by the device.
The drives accept 1 SDO server type, and no client type SDOs.
Sub-indexes 1 and 2 hold the allocation addresses of the transmitting and receiving SDOs.

Index	0x1200				
Symbol	Server SDO parameter	**Length (byte)**		**Min value**	
Object Code	Record	**Elements**	3	**Max value**	
Data Type	SDO Parameter	**Access**		**Default value**	
		PDO mapping			

Sub-indexes:

Index	0x1200:00				
Symbol	number_of_entries	**Length (byte)**	1	**Min value**	0x02
Object Code	Variable	**Elements**		**Max value**	0x02
Data Type	Unsigned8	**Access**	Read	**Default value**	0x02
		PDO mapping	No		

Index	0x1200:01				
Symbol	COD-ID RxSDO (Cl -> Sv)	**Length (byte)**	4	**Min value**	0x00000000
Object Code	Variable	**Elements**		**Max value**	0xFFFFFFFF
Data Type	Unsigned32	**Access**	Read/Write	**Default value**	0x00000600+ NodeID
		PDO mapping	No		

Index	0x1200:02				
Symbol	COD-ID TxSDO (Sv -> Cl)	**Length (byte)**	4	**Min value**	0x00000000
Object Code	Variable	**Elements**		**Max value**	0xFFFFFFFF
Data Type	Unsigned32	**Access**	Read/Write	**Default value**	0x00000580+ NodeID
		PDO mapping	No		

Legend:
- Sv = Server
- Cl = Client

This object contains the communication parameters for the PDOs that the devis is enabled to receive. AEC's devices support 4 PDOs in reception. The PDOs are sent without the presence of supplementary characters caused by the protocol, which gives very fast communication. The PDOs are ideal for transmitting and receiving data in real time. They can be programmed to be cyclic or asynchronous.

Index	0x1400+(No.PDO-1)				
Symbol	Receive PDO parameter	**Length (byte)**		**Min value**	
Object Code	Record	**Elements**	3	**Max value**	
Data Type	PDOCommPar	**Access**		**Default value**	
		PDO mapping			

Sub-indexes:

Index	0x1400+(No.PDO-1):00				
Symbol	number_of_entries	**Length (byte)**	1	**Min value**	0x02
Object Code	Variable	**Elements**		**Max value**	0x02
Data Type	Unsigned8	**Access**	Read	**Default value**	0x02
		PDO mapping	No		

Index	0x1400+(No.PDO-1):01				
Symbol	RxPDO COB-ID	**Length (byte)**	4	**Min value**	0x00000000
Object Code	Variable	**Elements**		**Max value**	0xFFFFFFFF
Data Type	Unsigned32	**Access**	Read/Write	**Default value**	Object 0x1400 0x200+ NodeID Object 0x1401 0x300+ NodeID Object 0x1402 0x400+ NodeID Object 0x1403 0x500+ NodeID
		PDO mapping	No		

Details:

Bit	Value	Description
31 MSB	0	Valid PDO
	1	Not valid PDO
30	0	RTR enabled on this PDO
	1	RTR disabled on thisPDO
29	0	11-bit ID (CAN 2.0A)
	1	29-bit ID (CAN 2.0B)
[28..11]	0	if bit29 = 0
	x	if bit29 = 1
[10..0] LSB	x	COB-ID

Index	0x1400+(No.PDO-1):02				
Symbol	Transmission_type	**Length (byte)**	1	**Min value**	0x00
Object Code	Variable	**Elements**		**Max value**	0xFF
Data Type	Unsigned8	**Access**	Read/Write	**Default value**	0xFE
		PDO mapping	No		

Transmission type (see the table)

Transmission type	PDO transmission				
	cyclic	acyclic	synchronous	asynchronous	RTR only
0		X	X		
1 - 240	X		X		
241 - 251	Reserved				
252			X		X
253				X	X
254				X	
255				X	

The synchronous transmissions (0 - 240) u use the SYNC object as a trigger for the sending of PDOs.

In case that type of transmission is 0, the sending of PDOs occurs in synchronous manner (SYNC triggered) but not periodic, while for transmissions between 1 and 240, the sending occurs in synchronous and periodic manner.

The type of transmission also defines how many SYNC messages must be received before the start of the PDOs transmission (if transmission_type = 10, than it's necessary to receive 10 SYNC messages before sending the PDOs).

With transmissions type 252 and 253, the sending of the PDOs only after the receipt of a RTR data request remote frame.
In case of 252, the update of the data (not the sending) occurs in synchronous manner (managed by the SYNC message). In case of 253, the update of the data occurs at the receipt of the RTR.

In 254 and 255 modes, the transmission of PDOs is "event triggered": the transmission occurs an asynchronous manner, when a determined event occurs.

In this modes, in case an event timer is set, the transmission occurs at the expiry of each event timer (also if no other events have occurred), in addition to the moment when an event occurs.

When an event occurs, the event timeris re-initialized.

This object defi nes the map of the values to be sent through the PDOs.
The minimum granularity is 8 bits, therefore the maximum number of mappable objects is 8.
NSub-index 0 is used to set the number of objects mapped in the PDO.
Firstly, the sub-index 0 should be set at 0. Then set the indexes from 1 to 8 with the codes of the objects to be mapped. Finally, set sub-index 0 with the number of mapped objects.
If the length of the mapped objects exceeds 64 bits, one of the following abort messages is generated by the SDO: 0x06020000, 0x06040041 o 0x06040042.
If mapped object does not exist, one of the following abort messages is generated by the SDO: 0x06020000 or 0x06040041.

Index	0x1600+(No.PDO-1)				
Symbol	Receive PDO mapping	**Length (byte)**		**Min value**	
Object Code	Record	**Elements**	9	**Max value**	
Data Type	PDO Mapping	**Access**		**Default value**	
		PDO mapping			

Sub-indexes:

Index	0x1600+(No.PDO-1):00				
Symbol	number_of_entries	**Length (byte)**	1	**Min value**	0x00
Object Code	Variable	**Elements**		**Max value**	0x08
Data Type	Unsigned8	**Access**	Read	**Default value**	
		PDO mapping	No		

Index	0x1600+(No.PDO-1):01..08				
Symbol	PDO mapping	**Length (byte)**	4	**Min value**	0x00000000
Object Code	Variable	**Elements**		**Max value**	0xFFFFFFFF
Data Type	Unsigned32	**Access**	Read/Write	**Default value**	
		PDO mapping	No		

Default mapping:

Index	**Sub-Index**	**Mapped value**	
		Code	**Description**
1600	1	0x60400010	Controlword
	2	0x60600008	Modes of operation
1601	1	0x60400010	Controlword
	2	0x607A0020	Target position
1602	x	-	-
1603	x	-	-

Mapping parameters details:

Object ndex	**Object Sub-Index**	**Number of bits**
bit [31..16]	bit [15..8]	bit [7..0]

This object contains the communication parameters for the PDOs that the devis is enabled to transmit. AEC's devices support 4 PDOs in transmission. The PDOs are sent without the presence of supplementary characters caused by the protocol, which gives very fast communication. The PDOs are ideal for transmitting and receiving data in real time. They can be programmed to be cyclic or asynchronous.

Index	0x1800+(No.PDO-1)				
Symbol	Transmit PDO parameter	**Length (byte)**		**Min value**	
Object Code	Record	**Elements**	3	**Max value**	
Data Type	PDOCommPar	**Access**		**Default value**	
		PDO mapping			

Sub-indexes:

Index	0x1800+(No.PDO-1):00				
Symbol	number_of_entries	**Length (byte)**	1	**Min value**	0x02
Object Code	Variable	**Elements**		**Max value**	0x05
Data Type	Unsigned8	**Access**	Read	**Default value**	0x05
		PDO mapping	No		

Index	0x1800+(No.PDO-1):01				
Symbol	TxPDO COB-ID	**Length (byte)**	4	**Min value**	0x00000000
Object Code	Variable	**Elements**		**Max value**	0xFFFFFFFF
Data Type	Unsigned32	**Access**	Read/Write	**Default value**	Object 0x1800 0x180+ NodeID Object 0x1801 0x280+ NodeID Object 0x1802 0x380+ NodeID Object 0x1803 0x480+ NodeID
		PDO mapping	No		

Details:

Bit	Valore	Descrizione
31 MSB	0	Valid PDO
	1	Not valid PDO
30	0	RTR enabled on this PDO
	1	RTR disabled on this PDO
29	0	11-bit ID (CAN 2.0A)
	1	29-bit ID (CAN 2.0B)
[28..11]	0	if bit29 = 0
	x	if bit29 = 1
[10..0] LSB	x	COB-ID

Index	0x1800+(No.PDO-1):02				
Symbol	Transmission_type	**Length (byte)**	1	**Min value**	0x00
Object Code	Variable	**Elements**		**Max value**	0xFF
Data Type	Unsigned8	**Access**	Read/Write	**Default value**	0xFE
		PDO mapping	No		

Transmission type (see the table)

Typer of transmission	PDO transmission				
	cyclic	acyclic	synchronous	asynchronous	RTR only
0		X	X		
1 - 240	X		X		
241 - 251	Riservati				
252			X		X
253				X	X
254				X	
255				X	

The synchronous transmissions (0 - 240) u use the SYNC object as a trigger for the sending of PDOs.
In case that type of transmission is 0, the sending of PDOs occurs in synchronous manner (SYNC triggered) but not periodic, while for transmissions between 1 and 240, the sending occurs in synchronous and periodic manner.
The type of transmission also defines how many SYNC messages must be received before the start of the PDOs transmission (if transmission_type = 10, than it's necessary to receive 10 SYNC messages before sending the PDOs).
With transmissions type 252 and 253, the sending of the PDOs only after the receipt of a RTR data request remote frame.
In case of 252, the update of the data (not the sending) occurs in synchronous manner (managed by the SYNC message). In case of 253, the update of the data occurs at the receipt of the RTR.
In 254 and 255 modes, the transmission of PDOs is "event triggered": the transmission occurs an asynchronous manner, when a determined event occurs.
In this modes, in case an event timer is set, the transmission occurs at the expiry of each event timer (also if no other events have occurred), in addition to the moment when an event occurs. When an event occurs, the event timeris re-initialized.

Index	0x1800+(No.PDO-1):03				
Symbol	inhibit_time	**Length (byte)**	2	**Min value**	0x0000
Object Code	Variable	**Elements**		**Max value**	0xFFFF
Data Type	Unsigned16	**Access**	Read/Write	**Default value**	0x0000
		PDO mapping	No		

Inhibits the transmission of PDOs for the set value.(value expressed in multiples of 100 microseconds)

Index	0x1800+(No.PDO-1):04				
Symbol	Reserved	**Length (byte)**		**Min value**	
Object Code		**Elements**		**Max value**	
Data Type		**Access**		**Default value**	
		PDO mapping			

Reserved object.

Index	0x1800+(No.PDO-1):05				
Symbol	event_timer	**Length (byte)**	2	**Min value**	0x0000
Object Code	Variable	**Elements**		**Max value**	0xFFFF
Data Type	Unsigned16	**Access**	Read/Write	**Default value**	0x0000
		PDO mapping	No		

Event timer for sending timed PDO (managed only with trasmission types 254 or 255).
Value expressed in milliseconds.

This object defi nes the map of the values to be sent through the PDOs.
The minimum granularity is 8 bits, therefore the maximum number of mappable objects is 8.
NSub-index 0 is used to set the number of objects mapped in the PDO.
Firstly, the sub-index 0 should be set at 0. Then set the indexes from 1 to 8 with the codes of the objects to be mapped. Finally, set sub-index 0 with the number of mapped objects.
If the length of the mapped objects exceeds 64 bits, one of the following abort messages is generated by the SDO: 0x06020000, 0x06040041 o 0x06040042.
If mapped object does not exist, one of the following abort messages is generated by the SDO: 0x06020000 or 0x06040041.

Index	0x1A00+(No.PDO-1)				
Symbol	Transmit PDO mapping	**Length (byte)**		**Min value**	
Object Code	Record	**Elements**	9	**Max value**	
Data Type	PDO Mapping	**Access**		**Default value**	
		PDO mapping			

Sub-indexes:

Index	0x1A00+(No.PDO-1):00				
Symbol	number_of_entries	**Length (byte)**	1	**Min value**	0x00
Object Code	Variable	**Elements**		**Max value**	0x08
Data Type	Unsigned8	**Access**	Read	**Default value**	
		PDO mapping	No		

Index	0x1A00+(No.PDO-1):01..08				
Symbol	PDO mapping	**Length (byte)**	4	**Min value**	0x00000000
Object Code	Variable	**Elements**		**Max value**	0xFFFFFFFF
Data Type	Unsigned32	**Access**	Read/Write	**Default value**	
		PDO mapping	No		

Default mapping:

Index	**Sub-Index**	**Mapped value**	
		Code	**Description**
1A00	1	0x60410010	Statusword
	2	0x60610008	Modes of operation display
1A01	1	0x60410010	Statusword
	2	0x60640020	Position actual value
1A02	x	-	-
1A03	x	-	-

Mapping parameters details:

Object index	**Object sub-Index**	**Numbers of bits**
bit [31..16]	bit [15..8]	bit [7..0]

DSP-402 V1.1 Objects

Summary table

Object	Object Name	Sub-index	Data type	Attributes
0x603F	Error code	0	Unsigned 16	RO
0x6040	Controlword	0	Unsigned 16	RW / WMAP
0x6041	Statusword	0	Unsigned 16	RO / MAP
0x605A	Quick-Stop option code	0	Integer16	RW
0x605B	Shutdown option code	0	Integer16	RW
0x605C	Disable operation option code	0	Integer16	RW
0x605D	Stop option code	0	Integer16	RW
0x605E	Fault reaction option code	0	Integer16	RW
0x6060	Modes of operation	0	Integer8	WO / WMAP
0x6061	Modes of operation display	0	Integer 8	RO / MAP
0x6064	Position actual value	0	Integer 32	RO / MAP
0x6065	Following error window	0	Unsigned 32	RW / WMAP
0x6066	Following error time-out	0	Unsigned 16	RW / WMAP
0x6067	Position window	0	Unsigned 32	RW / WMAP
0x6068	Position window time-out	0	Unsigned 16	RW / WMAP
0x606C	Velocity actual value	0	Integer 32	RO / MAP
0x607A	Target position	0	Integer 32	RW / WMAP
0x607C	Home offset	0	Integer 32	RW / WMAP
0x6081	Profile velocity	0	Unsigned 32	RW / WMAP
0x6082	End Velocity	0	Unsigned 32	RW / WMAP
0x6083	Profile acceleration	0	Unsigned 32	RW / WMAP
0x6084	Profile deceleration	0	Unsigned 32	RW / WMAP
0x6085	Quick stop deceleration	0	Unsigned 32	RW / WMAP
0x6086	Motion profile type	0	Integer 16	RW / WMAP
0x6098	Homing method	0	Integer 8	RW / WMAP
0x6099	Homing speeds	3	Unsigned 32 Array	RW / WMAP
0x609A	Homing acceleration	0	Unsigned 32	RW / WMAP
0x60C0	Interpolation Submode Select	0	Integer 16	RW
0x60C1	Interpolation Data Record	2	Array	RW / WMAP
0x60C2	Interpolation Time Period	3	Array	RW / WMAP
0x60C3	Interpolation Sync Definition	3	Array	RW
0x60C4	Interpolation Data Configuration	7	Array	RW / WMAP
0x60FD	Digital inputs	0	Unsigned 32	RO / MAP
0x60FE	Digital outputs	3	Unsigned 32 Array	RW / WMAP
0x6402	Motor type	0	Unsigned 16	RW
0x6502	Supported drive modes	0	Unsigned 32	RO
0x6504	Drive manufacturer	0	Visible String	RW
0x6505	http drive catalog address	0	Visible String	RW

Legend:

RO → Read only RW → Read/Write MAP → Mapping read
WO → Write only WMAP → Mapping write

Error Code 0x603F

The error_code captures the code of the last error that occurred to the drive. It corresponds to the value saved in the 16 low bits of the 0x1003 object pre_defined_error_field.

Index	0x603F				
Symbol	error_code	**Length (byte)**	2	**Min value**	0x0000
Object Code	Variable	**Elements**		**Max value**	0xFFFF
Data Type	Unsigned16	**Access**	Read only	**Default value**	0x0000
		PDO mapping	No		

Details:

Bit	Description
[15..8]	Error code
[7..0]	Error details

Controlword 0x6040

The modification of some bits in the ControlWord and the external signals (transitions) result in the generation of a Device control command.
The ControlWord is always mapped in the first two bytes of PDO messages being received.

Index	0x6040				
Symbol	Controlword	**Length (byte)**	2	**Min value**	0x0000
Object Code	Variable	**Elements**		**Max value**	0xFFFF
Data Type	Unsigned16	**Access**	Read only	**Default value**	0x0000
		PDO mapping	Write		

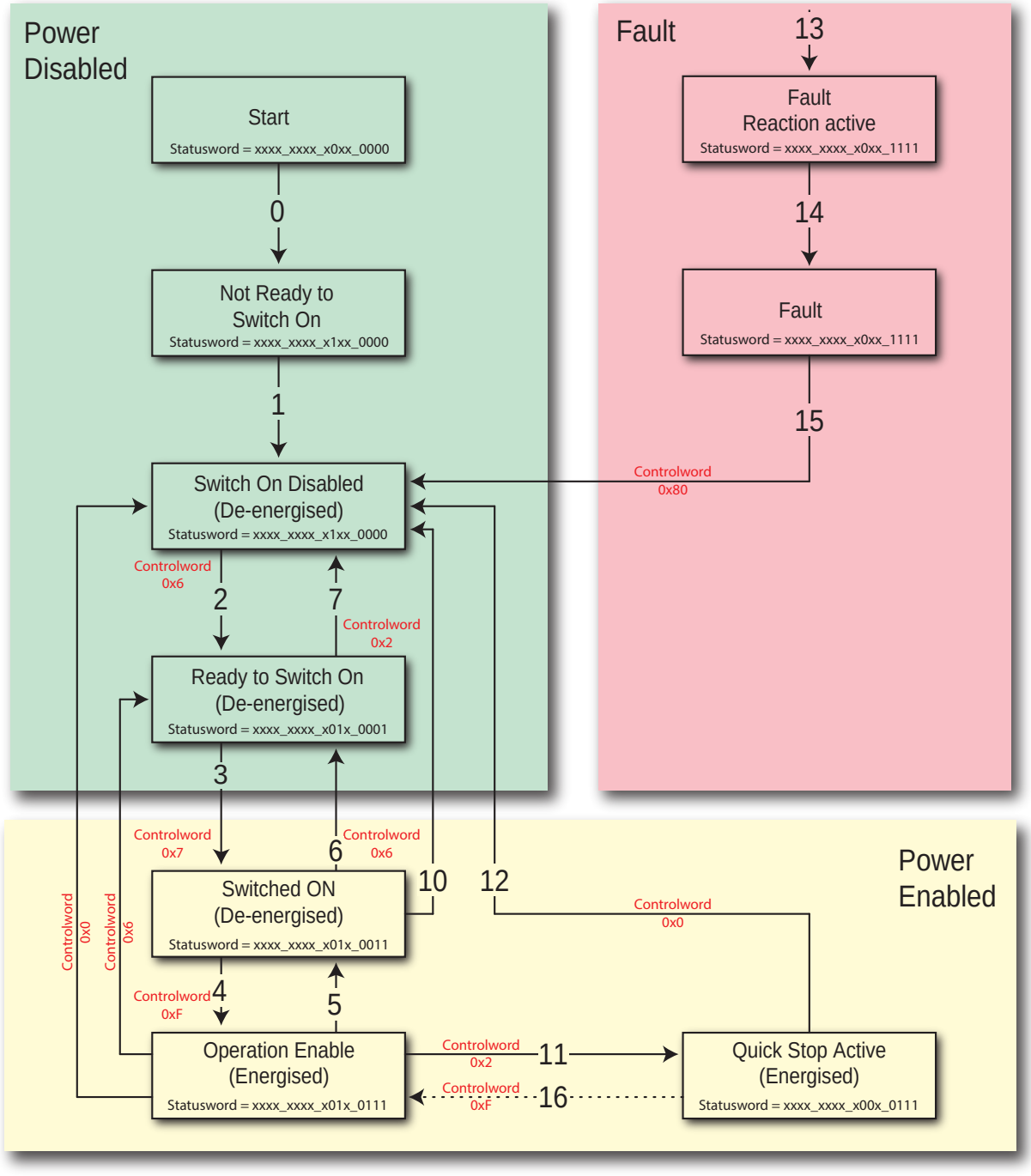
Details:

Bit	Profile position mode	Homing mode	AEC Velocity mode (JOG)
0	Switch ON		
1	Disable voltage		
2	Quick stop		
3	Enable operation		
4	1(re) = New setpoint	1(re) = Homing operation start	1 = Start
			0 = Stop
5	1 = Change setpoint immediately	Reserved	0 = CW
			1= CCW
6	0 = Absolute position	Reserved	0 = Speed update on Start jog
	1 = Relative position		1 = Continuous speed update
7	Reset fault		
8	Reserved		
9	Reserved		
10	Reserved		
[11..15]	Manufacturer specific		

(re) = rising edge

Command	Bit of Controlword					Transitions
	b7	b3	b2	b1	b0	
Shutdown	0	X	1	1	0	2, 6, 8
Switch ON	0	X	1	1	1	3
Disable Voltage	0	X	X	0	X	7, 9, 10, 12
Quick Stop	0	X	0	1	X	7, 10, 11
Disable Operation	0	0	1	1	1	5
Enable Operation	0	1	1	1	1	4, 16
Fault Reset	↑	X	X	X	X	15

DSP-402 State Machine



STATUSWORD 0x6041

The StatusWord indicates the current state of the drive and it is always mapped in the first two bytes of transmitted PDO messages.

Index	0x6041				
Symbol	Statusword	**Length (byte)**	2	**Min value**	0x0000
Object Code	Variable	**Elements**		**Max value**	0xFFFF
Data Type	Unsigned16	**Access**	Read only	**Default value**	0x0000
		PDO mapping	Read		

Details:

Bit	Profile position mode	Homing mode	AEC Velocity mode (JOG)
0	Ready to switch ON		
1	Switched ON		
2	Operation Enabled		
3	Fault		
4	Voltage disabled		
5	Quick stop		
6	Switched on disabled		
7	Warning		
8	Motor moving		
9	Remote		
10	Target reached		
11	Internal limit active		
12	Setpoint acquired	Home carried out	1 = Motor stopped
13	Following error	Error in homing sequence	Reserved
14	Manufacturer specific		
15	Manufacturer specific		

Command	Bit of Statusword					
	b6	b5	b3	b2	b1	b0
	Switch ON disabled	Quick Stop	Fault	Operation Enabled	Switched ON	Ready to Switch ON
Not ready to switch ON	0	X	0	0	0	0
Switch ON disabled	1	X	0	0	0	0
Ready to switch ON	0	1	0	0	0	1
Switched ON	0	1	0	0	1	1
Operation Enabled	0	1	0	1	1	1
Fault	0	X	1	1	1	1
Fault Reaction Active	0	X	1	1	1	1
Quick Stop Active	0	0		1	1	1

Quick Stop option code

0x605A

The Quick-Stop option code defines the type of action to be carried out in case a Quick-Stop is requested.

Index	0x605A				
Symbol	quick_stop_option_code	**Length (byte)**	2	**Min value**	0x8000
Object Code	Variable	**Elements**		**Max value**	0x7FFF
Data Type	Integer16	**Access**	Read / Write	**Default value**	0x0002
		PDO mapping	No		

Details:

Bit	Description
[-32768 ..-1]	Manufacturer spedific
0	Disable drive function
1	Slow down on slow down ramp and go in “Switch On Disabled”
2	Slow down on quick stop ramp and go in “Switch On Disabled”
3	Reserved
4	Reserved
5	Slow down on slow down ramp and stay in Quick-stop
6	Slow down on quick stop ramp and stay in Quick-stop
7	Reserved
8	Reserved
[9..32767]	Reserved

Shutdown option code

0x605B

The Shutdown option code defines the type of action to be carried out during the transition 8 of the state machine (OPERATION ENABLE → READY TO SWITCH ON)

Index	0x605B				
Symbol	shutdown_option_code	**Length (byte)**	2	**Min value**	0x8000
Object Code	Variable	**Elements**		**Max value**	0x7FFF
Data Type	Integer16	**Access**	Read / Write	**Default value**	0x0000
		PDO mapping	No		

Details:

Bit	Description
[-32768 ..-1]	Manufacturer spedific
0	Disable drive function
1	Slow down on slow down ramp
[2..32767]	Reserved

Disable operation option code 0x605C

The Disable Operation option code ddefines the type of action to be carried out during the transition 5 of the state machine (OPERATION ENABLE → SWITCHED ON)

Index	0x605C				
Symbol	disable_operation_option_code	**Length (byte)**	2	**Min value**	0x8000
Object Code	Variable	**Elements**		**Max value**	0x7FFF
Data Type	Integer16	**Access**	Read / Write	**Default value**	0x0001
		PDO mapping	No		

Details:

Bit	Description
[-32768 ..-1]	Manufacturer spedific
0	Disable drive function
1	Slow down on slow down ramp
[2..32767]	Reserved

Stop option code 0x605D

The Stop option code defines the type of action to be carried out in case a Stop is requested.

Index	0x605D				
Symbol	stop_option_code	**Length (byte)**	2	**Min value**	0x8000
Object Code	Variable	**Elements**		**Max value**	0x7FFF
Data Type	Integer16	**Access**	Read / Write	**Default value**	0x0001
		PDO mapping	No		

Details:

Bit	Description
[-32768 ..-1]	Manufacturer spedific
0	Disable drive function
1	Slow down on slow down ramp
2	Slow down on quick stop ramp
[3..32767]	Reserved

Fault Reaction option code 0x605E

The Fault reaction option code defines the type of action to be carried out in case of Fault.

Index	0x605E				
Symbol	Fault_reaction_option_code	**Length (byte)**	2	**Min value**	0x8000
Object Code	Variable	**Elements**		**Max value**	0x7FFF
Data Type	Integer16	**Access**	Read / Write	**Default value**	0x0002
		PDO mapping	No		

Details:

Bit	Description
[-32768 ..-1]	Manufacturer spedific
0	Disable drive function
1	Slow down on slow down ramp
2	Slow down on quick stop ramp
[3..32767]	Reserved

MODES OF OPERATION 0x6060

The parameter modes_of_operation changes the actual operation mode of the drive.

Index	0x6060				
Symbol	modes_of_operation	**Length (byte)**	4	**Min value**	0x80
Object Code	Variable	**Elements**		**Max value**	0x7F
Data Type	Integer8	**Access**	Write only	**Default value**	0x01
		PDO mapping	Write		

Details:

Modo operativo	Codifica
[-128..-2]	Reserved
-1	AEC Velocity mode (JOG)
0	Reserved
1	Profile position mode
[2..5]	Reserved
6	Homing mode
7	Interpolated mode
[8..127]	Reserved

If you attempt to change the mode of operation with motor in movement, an Abort is commanded (Stop with emergency ramp) without the modification of the status of the state machine.

MODES OF OPERATION DISPLAY 0x6061

This parameter shows the actual operation mode of the drive. The meaning of the returned value corresponds to that of the modes_of_operation (index 6060h).

Index	0x6061				
Symbol	Modes_of_operation_display	**Length (byte)**	1	**Min value**	0x00
Object Code	Variable	**Elements**		**Max value**	0xFF
Data Type	Integer8	**Access**	Read only	**Default value**	0x01
		PDO mapping	Read		

Details:

Modo operativo	Code
[-128..-2]	Reserved
-1	AEC Velocity mode (JOG)
0	Reserved
1	Profile position mode
[2..5]	Reserved
6	Homing mode
7	Interpolated mode
[8..127]	Reserved

POSITION ACTUAL VALUE 0x6064

The absolute position of the axis is given by this object. The value is expressed in steps.

Index	0x6064				
Symbol	Position_actual_value	**Length (byte)**	4	**Min value**	0x80000000
Object Code	Variable	**Elements**		**Max value**	0x7FFFFFFF
Data Type	Integer32	**Access**	Read only	**Default value**	0x00000000
		PDO mapping	Read		

VELOCITY ACTUAL VALUE 0x606C

This object defines the instantaneous speed of the stepper motor. The value is expressed in hundredths of a revolution per second (rps x100).

Index	0x606C				
Symbol	Velocity_actual_value	**Length (byte)**	4	**Min value**	0x00000001
Object Code	Variable	**Elements**		**Max value**	0x00004E20
Data Type	Integer32	**Access**	Read only	**Default value**	0x000003E8
		PDO mapping	Read		

TARGET POSITION 0x607A

The Target_positionis the position required in ‘position profile’ operation mode, by using the current movement parameters as speed, acceleration, deceleration, movement profile etc. The quota to be reached in interpreted as absolute or relative, according to the the status of the bit ‘absolute_relative flag ‘ (bit 6) of the ControlWord.

Index	0x607A				
Symbol	Target_position	**Length (byte)**	4	**Min value**	0x80000000
Object Code	Variable	**Elements**		**Max value**	0x7FFFFFFF
Data Type	Integer32	**Access**	Read/Write	**Default value**	0x00000000
		PDO mapping	Write		

HOME OFFSET 0x607C

Forced quota after homing sequence. It’s the value, expressed in steps, that is overwritten on the current position, after executing the homing sequence.

Index	0x607C				
Symbol	Home_offset	**Length (byte)**	4	**Min value**	0x80000000
Object Code	Variable	**Elements**		**Max value**	0x7FFFFFFF
Data Type	Integer32	**Access**	Read/Write	**Default value**	0x00000000
		PDO mapping	Write		

Profile velocity 0x6081

Set the speed of the motor during a movement. The speed is expressed in hundredths of a revolution per second (rps x100).

Index	0x6081				
Symbol	Profile_velocity	**Length (byte)**	4	**Min value**	0x00000000
Object Code	Variable	**Elements**		**Max value**	0x7FFFFFFF
Data Type	Integer32	**Access**	Read/Write	**Default value**	0x00000064
		PDO mapping	Write		

End velocity (start/stop speed) 0x6082

Set the start/stop speed of the motor during a movement. The speed is expressed in hundredths of a revolution per second (rps x100)

Index	0x6082				
Symbol	End_velocity	**Length (byte)**	4	**Min value**	0xFFFFEC78
Object Code	Variable	**Elements**		**Max value**	0x00001388
Data Type	Unsigned32	**Access**	Read/Write	**Default value**	0x00000000
		PDO mapping	Write		

Profile acceleration 0x6083

Set the acceleration of the motor during a movement. The acceleration is expressed rps^2 x10 (tenths of a revolution per second^2).

Index	0x6083				
Symbol	Profile_acceleration	**Length (byte)**	4	**Min value**	0x00000001
Object Code	Variable	**Elements**		**Max value**	0x00030D40
Data Type	Unsigned32	**Access**	Read/Write	**Default value**	0x000003E8
		PDO mapping	Write		

Profile deceleration 0x6084

Set the deceleration of the motor during a movement. The deceleration is expressed rps^2 x10 (tenths of a revolution per second^2).

Index	0x6084				
Symbol	Profile_deceleration	**Length (byte)**	4	**Min value**	0x00000001
Object Code	Variable	**Elements**		**Max value**	0x00030D40
Data Type	Unsigned32	**Access**	Read/Write	**Default value**	0x000003E8
		PDO mapping	Write		

Set the emergency deceleration of the motor during a movement. The deceleration is expressed rps^2 x10 (tenths of a revolution per $second^2$).

Index	0x6085				
Symbol	Quick_stop_deceleration	**Length (byte)**	4	**Min value**	0x00000001
Object Code	Variable	**Elements**		**Max value**	0x00030D40
Data Type	Unsigned32	**Access**	Read/Write	**Default value**	0x000003E8
		PDO mapping	No		

MOTION PROFILE TYPE

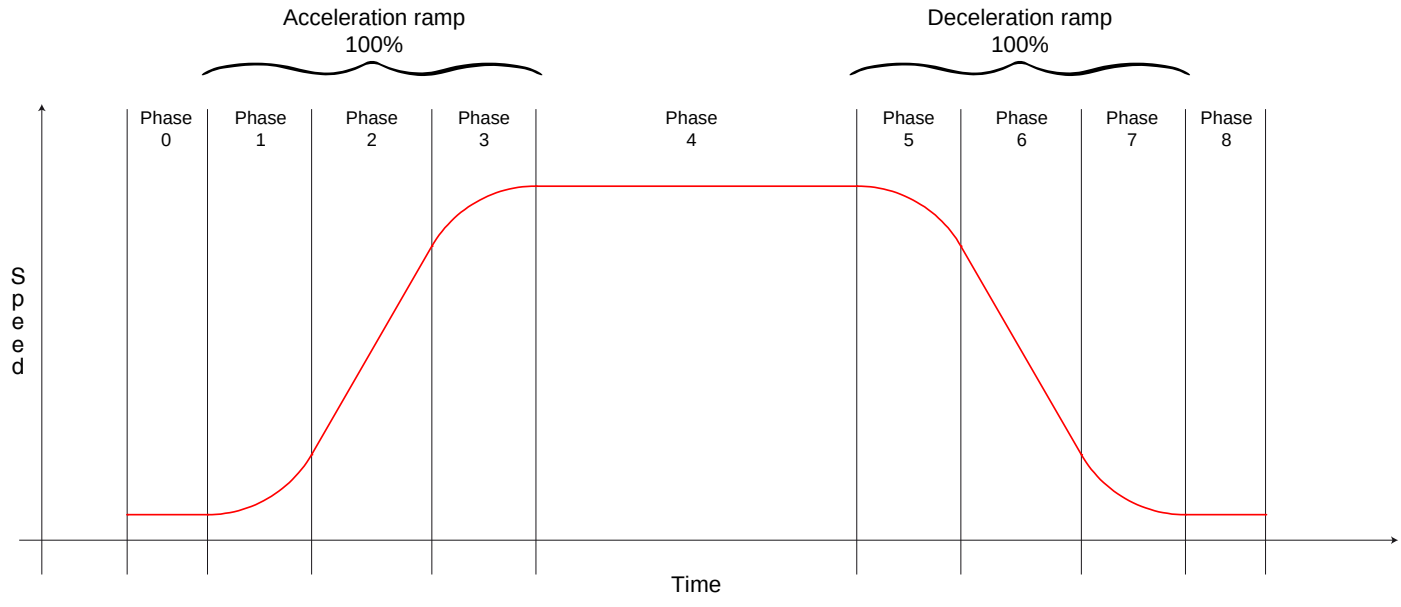
0x6086

This object sets the acceleration/deceleration ramp profile. It is possible to select either linear or S ramps.

Index	0x6086				
Symbol	Motion_profile_type	**Length (byte)**	2	**Min value**	0x8000
Object Code	Variable	**Elements**		**Max value**	0x7FFF
Data Type	Integer16	**Access**	Read/Write	**Default value**	0x0000
		PDO mapping	Write		

Details:

Operation mode	Code
[-32768..-11]	Reserved
-10	Pure S ramp (phase 2 duration / phase 6 = 0%)
-9	S ramp (phase 2 duration / phase 6 = 10%)
-8	S ramp (phase 2 duration / phase 6 = 20%)
-7	S ramp (phase 2 duration / phase 6 = 30%)
-6	S ramp (phase 2 duration / phase 6 = 40%)
-5	S ramp (phase 2 duration / phase 6 = 50%)
-4	S ramp (phase 2 duration / phase 6 = 60%)
-3	S ramp (phase 2 duration / phase 6 = 70%)
-2	S ramp (phase 2 duration / phase 6 = 80%)
-1	S ramp (phase 2 duration / phase 6 = 90%)
0	Trapezoidal ramp phase 2 duration 2 / phase 6 = 100%)
1	Not used
2	Not used
3	Not used
[4..32768]	Reserved for further use



This object sets the requested type of homing. It is possible to select various homing methods: only with backward limit switch, or backward limit switch + Top, or just Top.

Index	0x6098				
Symbol	Homing_method	**Length (byte)**	1	**Min value**	0x80
Object Code	Variable	**Elements**		**Max value**	0x7F
Data Type	Integer8	**Access**	Read/Write	**Default value**	0x00
		PDO mapping	Write		

Details:

Value		Reset quota	Synchro flag
[-128..-11]	Reserved		
-16	Homing on FLS + motor encoder TOP, positive direction		
-15	Homing on FLS + motor encoder TOP, negative direction		
-14	Homing on FLS, positive direction		
-13	Homing on FLS, negative direction		
-12	Homing with forward mechanical limit + encoder TOP	Yes	Yes
-11	Homing with backward mechanical limit + encoder TOP	Yes	Yes
-10	Homing with forward mechanical limit	Yes	Yes
-9	Homing with backward mechanical limit	Yes	Yes
-8	Homing with forward mechanical limit + axis measuring	Yes	Yes
-7	Homing with backward mechanical limit + axis measuring	Yes	Yes
-6	Homing only with TOP in positive direction	Yes	Yes
-5	Homing only with TOP in negative direction	Yes	Yes
-4	Homing with BLS + TOP rising edge, positive direction	Yes	Yes
-3	Homing with BLS + TOP rising edge, negative direction	Yes	Yes
-2	Homing only with BLS in positive direction	Yes	Yes
-1	Homing only with BLS in negative direction	Yes	Yes
0	Homing only with BLS in negative direction	Yes	Yes
35	Homing on place (only in CanOpen, for DS402 compatibility)	Yes	Yes
37	Homing on place (only in CanOpen, for DS402 compatibility)	Yes	Yes

HOMING SPEEDS 0x6099

Set the homing parameters, like: speed during search for switch and speed during search for zero. The speeds are expressed in hundredths of a revolution per second (rps x100).

Index	0x6099				
Symbol	Homing_speeds	**Length (byte)**		**Min value**	
Object Code	Array	**Elements**	3	**Max value**	
Data Type	Unsigned32	**Access**		**Default value**	
		PDO mapping			

Sub-indexes:

Index	0x6099:00				
Symbol	number_of_entries	**Length (byte)**	1	**Min value**	0x02
Object Code	Variable	**Elements**		**Max value**	0x02
Data Type	Unsigned8	**Access**	Read	**Default value**	0x02
		PDO mapping	No		

Index	0x6099:01				
Symbol	Speed_during_search_for_switch	**Length (byte)**	4	**Min value**	0x00000000
Object Code	Variable	**Elements**		**Max value**	0x00002710
Data Type	Unsigned32	**Access**	Read/Write	**Default value**	0x00000064
		PDO mapping	Write		

Index	0x6099:02				
Symbol	Speed_during_search_for_zero	**Length (byte)**	4	**Min value**	0x00000000
Object Code	Variable	**Elements**		**Max value**	0x00002710
Data Type	Unsigned32	**Access**	Read/Write	**Default value**	0x0000000A
		PDO mapping	Write		

If the Speed_during_search_for_switch is less than or equal to the Speed_during_search_for_zero, the ramps are disabled.

HOMING ACCELERATION 0x609A

This object sets the acceleration/deceleration ramp during the homing sequence. The value is expressed in rps^2 x10 (tenths of a revolution per $second^2$).

Index	0x609A				
Symbol	Homing_acceleration	**Length (byte)**	4	**Min value**	0x00000001
Object Code	Variable	**Elements**		**Max value**	0x0007A120
Data Type	Unsigned32	**Access**	Read/Write	**Default value**	0x00000001
		PDO mapping	Write		

Interpolation Submode Select 0x60C0

Defines the active interpolation mode.

Index	0x60C0				
Symbol	Interpolation_submode_select	**Length (byte)**	2	**Min value**	0x8000
Object Code	Variable	**Elements**		**Max value**	0x7FFF
Data Type	Integer16	**Access**	Read/Write	**Default value**	0x0000
		PDO mapping	No		

Details:

Operation mode	
[- 32768 .. -1]	Reserved
0	Linear interpolation
[1 .. 32768]	Reserved

Interpolation Data Record 0x60C1

This is a two positions buffer which contains the target quotas sent by the interpolator..

Index	0x60C1				
Symbol	Interpolation_data_record	**Length (byte)**		**Min value**	
Object Code	RECORD	**Elements**	2	**Max value**	
Data Type	Unsigned32	**Access**		**Default value**	
		PDO mapping			

Sub-indexes:

Index	0x60C1:01				
Symbol	1st_ip_function_parameter	**Length (byte)**	4	**Min value**	0x80000000
Object Code	Variable	**Elements**		**Max value**	0x7FFFFFFF
Data Type	Integer32	**Access**	Read/Write	**Default value**	0x00000000
		PDO mapping	Write		

Index	0x60C1:02				
Symbol	2nd_ip_function_parameter	**Length (byte)**	4	**Min value**	0x80000000
Object Code	Variable	**Elements**		**Max value**	0x7FFFFFFF
Data Type	Integer32	**Access**	Read/Write	**Default value**	0x00000000
		PDO mapping	Write		

The interpolated movement of the drive starts after the interpolation profile has been enabled, and both of the buffer elements have been written.

Interpolation Time Period 0x60C2

These parameters are used to synchronize Master and Slave during the interpolation.

Index	0x60C2				
Symbol	Interpolation_time_period	**Length (byte)**		**Min value**	
Object Code	RECORD	**Elements**	2	**Max value**	
Data Type		**Access**		**Default value**	
		PDO mapping			

Sub-indexes:

Index	0x60C2:01				
Symbol	ip_time_units	**Length (byte)**	1	**Min value**	0x01
Object Code	Variable	**Elements**		**Max value**	0x0A
Data Type	Unsigned8	**Access**	Read/Write	**Default value**	0x04
		PDO mapping	Write		

Index	0x60C2:02				
Symbol	ip_time_index	**Length (byte)**	1	**Min value**	0xFD
Object Code	Variable	**Elements**		**Max value**	0xFD
Data Type	Unsigned8	**Access**	Read/Write	**Default value**	0xFD
		PDO mapping	No		

Interpolation Sync Definition 0x60C3

Defines the synchronization parameters of the device.

Index	0x60C3				
Symbol	Interpolation_sync_definition	**Length (byte)**		**Min value**	
Object Code	ARRAY	**Elements**	2	**Max value**	
Data Type		**Access**		**Default value**	
		PDO mapping			

Sub-indexes:

Index	0x60C3:01				
Symbol	syncronize_on_group	**Length (byte)**	1	**Min value**	0x00
Object Code	Variable	**Elements**		**Max value**	0x00
Data Type	Unsigned8	**Access**	Read/Write	**Default value**	0x00
		PDO mapping	No		

Index	0x60C3:02				
Symbol	ip_sync_every_n_event	**Length (byte)**	1	**Min value**	0x01
Object Code	Variable	**Elements**		**Max value**	0x01
Data Type	Unsigned8	**Access**	Read/Write	**Default value**	0x01
		PDO mapping	No		

Defines method to store position data record.

Index	0x60C4				
Symbol	Interpolation_data_config	**Length (byte)**		**Min value**	
Object Code	RECORD	**Elements**	6	**Max value**	
Data Type		**Access**		**Default value**	
		PDO mapping			

Sub-indexes:

Index	0x60C4:01				
Symbol	max_buffer_size	**Length (byte)**	1	**Min value**	0x02
Object Code	Variable	**Elements**		**Max value**	0x02
Data Type	Unsigned8	**Access**	Read	**Default value**	0x02
		PDO mapping	No		

Index	0x60C4:02				
Symbol	actual_size	**Length (byte)**	1	**Min value**	0x00
Object Code	Variable	**Elements**		**Max value**	0x02
Data Type	Unsigned8	**Access**	Read/Write	**Default value**	0x00
		PDO mapping	No		

Index	0x60C4:03				
Symbol	buffer_organisation	**Length (byte)**	1	**Min value**	0x00
Object Code	Variable	**Elements**		**Max value**	0x00
Data Type	Unsigned8	**Access**	Read/Write	**Default value**	0x00
		PDO mapping	No		

Index	0x60C4:04				
Symbol	buffer_position	**Length (byte)**	1	**Min value**	0x00
Object Code	Variable	**Elements**		**Max value**	0x01
Data Type	Unsigned8	**Access**	Read/Write	**Default value**	0x00
		PDO mapping	Write		

Index	0x60C4:05				
Symbol	size_of_data_record	**Length (byte)**	1	**Min value**	0x01
Object Code	Variable	**Elements**		**Max value**	0x01
Data Type	Unsigned8	**Access**	Read	**Default value**	0x01
		PDO mapping	No		

Index	0x60C4:06				
Symbol	buffer_clear	**Length (byte)**	1	**Min value**	0x00
Object Code	Variable	**Elements**		**Max value**	0x01
Data Type	Unsigned8	**Access**	Write	**Default value**	0x00
		PDO mapping	No		

Details:

Operation modes	
0	Clear the buffer (access to the buffer disabled)
1	Enable the access to the buffer
[2 .. 256]	Reserved

Digital Inputs 0x60FD

Defines some simple digital inputs on board the drive. The inputs on LSB show the limit switches. The two most significant bytes give the 16 digital inputs on board the drive, for generic purposes.

Index	0x60FD				
Symbol	Digital_inputs	**Length (byte)**	4	**Min value**	0x00000000
Object Code	Variable	**Elements**		**Max value**	0xFFFF00F7
Data Type	Unsigned32	**Access**	Read only	**Default value**	0x00000000
		PDO mapping	Read		

Details:

Bit	Codifica
0	Negative limit switch (BLS active high)
1	Positive limit switch (FLS active high)
2	Home switch (TOP active high)
[3..15]	Reserved
[16..31]	Digital inputs (digital input [0..15]) on board the drive

Digital Outputs 0x60FE

Defines some simple digital outputs on board the driver. The third byte gives the 8 digital outputs on board the drive. The sub-index 2 defines the mask of the editable outpts.

Index	0x60FE				
Symbol	Digital_outputs	**Length (byte)**		**Min value**	
Object Code	Record	**Elements**	2	**Max value**	
Data Type		**Access**		**Default value**	
		PDO mapping			

Sub-indexes:

Index	0x60FE:01				
Symbol	Physical_output	**Length (byte)**	4	**Min value**	0x00000000
Object Code	Variable	**Elements**		**Max value**	0x00FF0000
Data Type	Unsigned32	**Access**	Read/Write	**Default value**	0x00000000
		PDO mapping	Write		

Index	0x60FE:02				
Symbol	bitmask	**Length (byte)**	4	**Min value**	0x00000000
Object Code	Variable	**Elements**		**Max value**	0xFFFFFFFF
Data Type	Unsigned32	**Access**	Read/Write	**Default value**	0x00FF0000
		PDO mapping	Write		

The bitmask parameter permits to select which outputs are editable. By setting a bit to 1, you enable the output; by setting a bit to 0, you disable the output, which will remain unchanged.

Details:

Bit	Code
[0..15]	Reserved
[16..23]	Digital outputs (digital output [0.. 8]) on board the drive
[24..31]	Reserved

MOTOR TYPE 0x6402

Shows the type of motor driven by the device (type 9, Microstep motor). The parameter can be changed, but it isn't stored into the NVRAM, so when the drive is re-switched on, the value returns to 9.

Index	0x6402				
Symbol	Motor_type	**Length (byte)**	2	**Min value**	0x0000
Object Code	Variable	**Elements**		**Max value**	0xFFFF
Data Type	Unsigned32	**Access**	Read/Write	**Default value**	0x0008
		PDO mapping	Write		

Details:

Value	Code	Supported
0	Non-Standard Motor	○
1	Phase Modulated DC Motor	○
2	Frequency Controlled DC Motor	○
3	PM Synchronous motor	○
4	FC synchronous motor	○
5	Switched Reluctance Motor	○
6	Wound Rotor Induction Motor	○
7	Squirrel Cage Induction Motor	○
8	Stepper Motor	●
9	Micro-Step Stepper Motor	●
10	Sinusoidal PM BL Motor	○
11	Trapezoidal PM BL Motor	○

○ Not supported / ● Supported

SUPPORTED DRIVE MODE 0x6502

Defines the operation modes implemented by the drive.

Index	0x6502				
Symbol	Supported_drive_mode	**Length (byte)**	4	**Min value**	0x00010021
Object Code	Variable	**Elements**		**Max value**	0x00010021
Data Type	Unsigned32	**Access**	Read only	**Default value**	0x00010021
		PDO mapping	Read		

Details:

Bit	Code	Supported
0	Profile Position Mode	●
1	Velocity Mode	○
2	Profile Velocity Mode	○
3	Profile Torque Mode	○
4	Reserved	○
5	Homing Mode	●
6	Interpolated Position Mode	●
[7..15]	Reserved	○
16	AEC Velocity Mode	●
[17..31]	Reserved	○

○ Not supported / ● Supported

DRIVE MANUFACTURER 0x6504

Shows the name of the manufacturer of the drive.
This object can be accessed either in reading and writing, but it is not possible to store new values into the non volatile memory, so it returns to the default value when the drive is re-switched on.

Index	0x6504				
Symbol	Drive_manufacturer	**Length (byte)**	27	**Min value**	
Object Code	Variable	**Elements**		**Max value**	
Data Type	Visible String	**Access**	Read/Write	**Default value**	Aec srl...Italy
		PDO mapping	No		

HTTP DRIVE CATALOG ADDRESS 0x6505

Contains the website where it's possible to download the datasheet of the drive.
This object can be accessed either in reading and writing, but it is not possible to store new values into the non volatile memory, so it returns to the default value when the drive is re-switched on.

Index	0x6505				
Symbol	http_drive_catalog_address	**Length (byte)**	14	**Min value**	
Object Code	Variable	**Elements**		**Max value**	
Data Type	Visible String	**Access**	Read/Write	**Default value**	www.aec-smd.it
		PDO mapping	No		

STATE MACHINE DSP-402 V1.1

INTRODUCTION

The state machine implemented in the drives for stepper motor with the CANOpen option is based on the Draft Standard 402 (V.1.1) specifi cations for movement control.

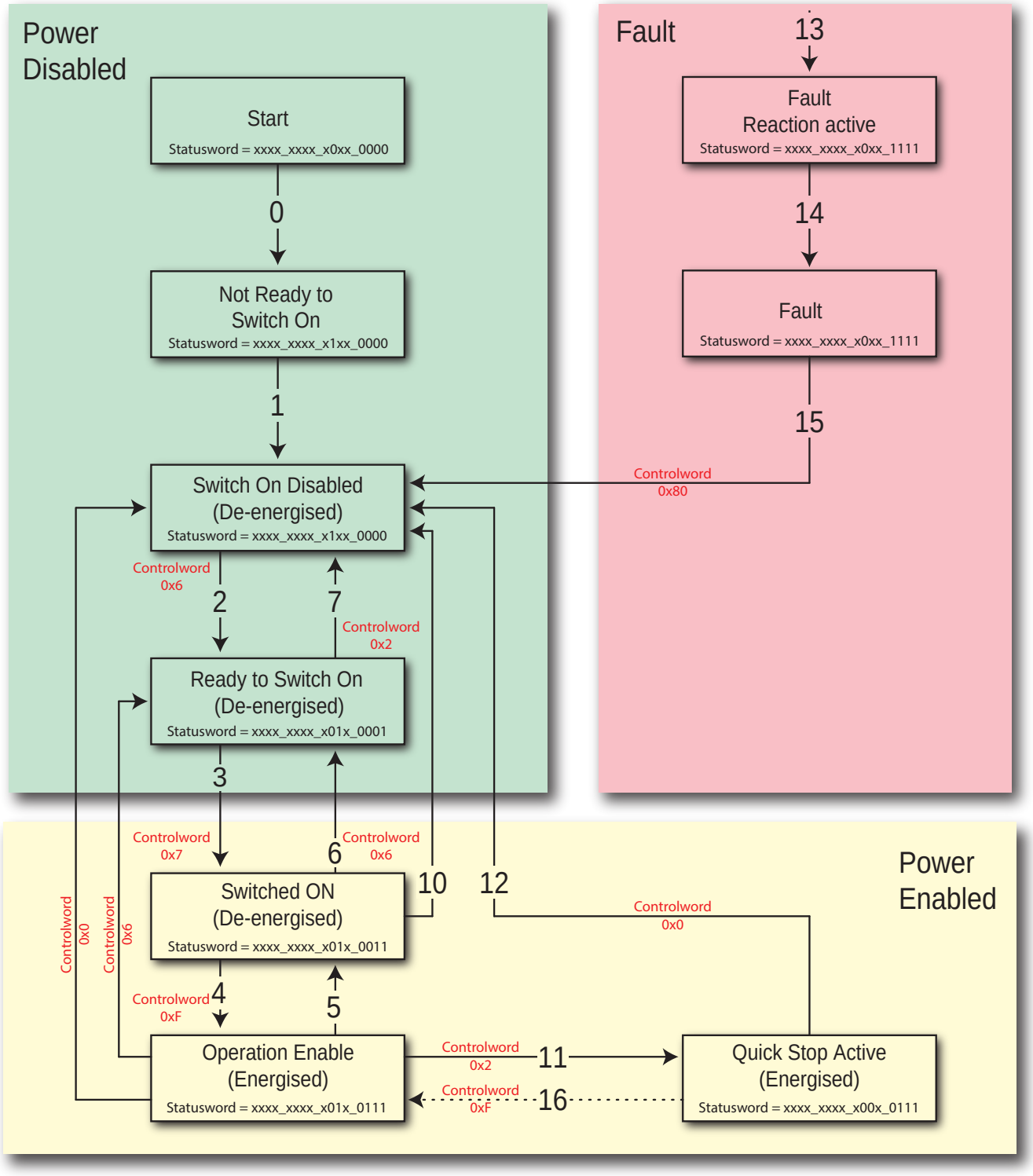
This standard defi nes the operation sequence and the states to enable the driver to switch on safely and in a well-defi ned manner. The standard also defi nes the states for re-enabling the driver after an error or an alarm.

The various states of the state machine of the device can be set manipulating the bits of the ControlWord object (index 0x6040) and the state of the state machine can be read through the StatusWord object (index 0x6041).

Only the mandatory states of the state machine have been implemented.

Transition definitions:

Transition	Controlword																
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	Value
1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	-
2	X	X	X	X	X	X	X	X	0	X	X	X	X	1	1	0	0x6
3	X	X	X	X	X	X	X	X	0	X	X	X	X	1	1	1	0x7
4	X	X	X	X	X	X	X	X	0	X	X	X	1	1	1	1	0xF
5	X	X	X	X	X	X	X	X	0	X	X	X	0	1	1	1	0x7
6	X	X	X	X	X	X	X	X	0	X	X	X	X	1	1	0	0x6
7	X	X	X	X	X	X	X	X	0	X	X	X	X	X	0	X	0x0
8	X	X	X	X	X	X	X	X	0	X	X	X	X	1	1	0	0x6
9	X	X	X	X	X	X	X	X	0	X	X	X	X	X	0	X	0x0
10	X	X	X	X	X	X	X	X	0	X	X	X	X	X	0	X	0x0
11	X	X	X	X	X	X	X	X	0	X	X	X	X	0	1	X	0x2
12	X	X	X	X	X	X	X	X	0	X	X	X	X	X	0	X	0x0
13	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	-
14	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	-
15	X	X	X	X	X	X	X	X	1	X	X	X	X	X	X	X	-
16	X	X	X	X	X	X	X	X	0	X	X	X	1	1	1	1	0xF


Power
Disabled
Start
Statusword = xxxx_xxxx_x0xx_0000
0
Not Ready to
Switch On
Statusword = xxxx_xxxx_x1xx_0000
1
Switch On Disabled
(De-energised)
Statusword = xxxx_xxxx_x1xx_0000
Controlword
0x6
2
7
Controlword
0x2
Ready to Switch On
(De-energised)
Statusword = xxxx_xxxx_x01x_0001
3
Controlword
0x7
6
Controlword
0x6
Switched ON
(De-energised)
Statusword = xxxx_xxxx_x01x_0011
10
12
Controlword
0x0
Controlword
0x0
Controlword
0x6
Controlword
0xF
4
5
Operation Enable
(Energised)
Statusword = xxxx_xxxx_x01x_0111
Controlword
0x2
11
Controlword
0xF
16
Quick Stop Active
(Energised)
Statusword = xxxx_xxxx_x00x_0111
Power
Enabled
Fault
13
Fault
Reaction active
Statusword = xxxx_xxxx_x0xx_1111
14
Fault
Statusword = xxxx_xxxx_x0xx_1111
15
Controlword
0x80

PROPRIETARY REGISTERS

REGISTERS

The registers are memory locations inside the drives, each one with specific functions. When a determined value is written in these locations, a function defined by the correspondent register is carried out.
Some of the registers are at 32 bit, and the access to these registers is done by reading or writing two consecutive registers, according to Modbus protocol.

N.B.: ALL THE INTERNAL RESOURCES, REGISTERS E VARIABLES OF THE AEC DRIVES ARE MAPPED IN THE MEMORY LOCATION 4 "HOLDING REGISTER".

N.B.: Verify if the Modbus addresses of the master start from 0 or 1. AEC Modbus addresses start from 0, in case of using a master in which they start from 1 (e.g. Siemens or Weintek) it is necessary to add 1 to the AEC Modbus address.
Example: Rposact= 0000+1= 0001-2

POSITION REGISTERS

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0000-1	0x2000	**Rposact**	Actual position of the axis	Step	0x80000000	0x7FFFFFFF	Signed32	LS – MSWORD MAP READ
0004-5	0x2004	**Rposactreq**	Theoric position os the axis	Step	0x80000000	0x7FFFFFFF	Signed32	LS -MSWORD MAP-READ
0008-9	0x2008	**Rpostarg**	Target position for GO/GOR functions	Step	0x80000000	0x7FFFFFFF	Signed32	LS – MSWORD ANA_T MAP WRITE
0012-13	0x200C	**Rupplim**	Upper limit quota	Step	0x80000000	0x7FFFFFFF	Signed32	LS – MSWORD SAVE MAP WRITE
0016-17	0x2010	**Rlowlim**	Lower limit quota	Step	0x80000000	0x7FFFFFFF	Signed32	LS – MSWORD SAVE MAP WRITE
0055-56	0x2037	**Rposactsa-ved**	Saved quota at switch-off of the drive	Step			Signed32	LS – MSWORD SAVE NO MAP RO
0057	0x2039	**Rposactsa-vedflag**	B0: 1= Quota saved correctly				Unsig-ned16	WORD RO

Automatic storing of the actual quota at the switch off of the drive

The drive is equipped with a circuit that automatically detect when the supply voltage of the logic stage drops below about 20Vdc.

When this event occurrs, the firmware interpretes it like a voltage loss, immediately cuts the current to the motor and, thanks to the residual load of the condensers, tries to store the actual quota (register Rposact) in the non-volatile memory.

When the voltage is again supplied to the logic stage, the firmware restarts and a dedicated function checks if the quota saved in the non-volatile memory is valid by executing some controls on data congruency.
If the stored quota is valid, this value is loaded in the register Rposactsaved (register 55) and the bit 0 of the register Rposactsavedflag (register 57) is set to 1.

If the stored quota is not valid, both Rposactsaved and Rposactsavedflag registers are load with the value 0.

The user has the possibility to check if there is an available valid switch off quota, and so pass the value of Rposactsaved in Rposact.

Please note that the motor must be in standstill at the moment of the voltage loss, else the saved quota will be not valid, because it may be affected by possible movements due to the inertia.

In case of voltage oscillations, the save of the quota is made only during the first signal of voltage loss. The saving is enabled again after the drive is powered again. In case the drive hasn't been completely switched off, if the quota Rposact is varied with respect to the saved quota, it is assumed that the control program had been able to put the motor in current and to move it in a controlled way.

It is essential to take account that this function uses the residual load of the condensers to execute the operations necessary to write the data into the non-volatile memory.
It may occur that the residual load of the condensers is not sufficient to complete the saving of the data.
In this case, when the drive is powered again, the data saved will be invalid. It will be necessary to decide the operations to be carried out, like an example an homing of the axis.

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0199	0x20C7	**Rstsflg**	Flag status register Bit 0: Drive enabled Bit 1: Drive in alarm Bit 2: Axis homed Bit 3: Motor in movement (theoric) Bit 4: Motor in acceleration Bit 5: Motor at constant speed Bit 6: Motor in deceleration Bit 7: Information contained in register Rstscllp Bit 8: Home executed with errors Bit 9: Status of current (0=CurOff/1=CurON) Bit 10: 1=Motor in position Bit 11: Following error Bit 12: Motor moved while in “disable” state (only with encoder). The range is given by the value of the Rdeadpos register. Bit 13: Counterclockwise rotation direction Bit 14: Actual quota out of software limits range Bit 15: Home in progress				Unsig-ned16	WORD MAP-READ RO
0200	0x20C8	**Rstsflg1**	Flag status 1 register Bit 0: Alarms are present in the buffer Bit 1: Warning is present Bit 2: Power OFF signal Bit 3: STOP in progress Bit 4: Task in progress Bit 5: Lower SW intervention limit Bit 6: Upper SW intervention limit Bit 7: BLS intervention (memory) Bit 8: FLS intervention (memory) Bit 9: Operation in voltage limit Bit 10: Saturated regulators Bit 11: Current limit is active Bit 12: Encoder phasing in progress Bit 13: 1 = Register 51 (Rextencvel) updated Bit 14: 1 = Register 89 (Rextenctopvel) updated Bit 15: STO active				Unsig-ned16	WORD MAP READ RO
0203	0x20CB	**Rstscllp**	Closed-loop flag status register Bit 0: Phased encoder Bit 1: Motor in theoric movement Bit 2: Motor in position Bit 3: Positioning alarm Bit 4: Following alarm Bit 5: Reserved Bit 6: Positioning time out Bit 7: Motor in real movement Bit 8: Current limit Bit 9: Speed limit Bit 10: Acceleration limit B11: B12: B13: B14: B15: N.B. bits 0-1-2 are used to set the bit 7 of Rstsflg				Unsig-ned16	WORD RO

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0058	0x203A	**Rflag1**	Operation mode flag 1 register (not savable) B0 = Wait end of movement B1 = Update JOG immediately B2= Enable the CAM table for function of positioning from table with strart from master quota + digital outputs. B3: B4: B5: B6: B7: B8: B9: B10: B11: B12: On the rising edge, resets quotas in closed loop (real quota and theoric quota) (equal to bit 12 of STW1 of Profibus) B13: B14: B15:		0	65535	Unsig-ned16	WORD

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0102-03	0x2066	**Rflag**	Operation mode flag register B0 = Wait end of movement B1 = Update velocity JOG immediately (see also BIT1 of Rflag1) B2 = Enable SW low limit B3 = Enable SW high limit B4 = Enable Back Limit Switch B5 = Enable Forward Limit Switch B6 = Reset Posact after Home B7 = Reset Motenc after Home B8 = Reset Motext after Home B9 = Reset Posact after Home offset B10 = Reset Motenc after Home offset B11 = Reset Motext after Home offset (0 = Reset quota / 1 = No reset quota) B12 = 1: Enable automatic offset correction in closed loop B13 = 1: Enable automatic PID current correction in closed loop B14 = 1: Unidirectional gear B15 = 1: Forward only gear B16 = 1: Back only gear B17 = 1: In Smart Mode, at “Current ON”, update the requested actual position with the actual quota obtained by the encoder B18 = 1: In Smart Mode, at “Current OFF”, doesn’t update the requested actual position with the actual quota obtained by the encoder, in order to permit the recovery of the quota at “Current ON” B19 = 0: USB port without slave address and fixed parameters at 9600,N,8,1 /1: USB port with parameters set by registers but without slave address. B20 = B21 = B22 = B23 = B24 = B25 = B26 = B27 = B28 = B29 = B30 = B31 =		0	0x7FFFFFFF	Unsig-ned32	LS – MSWORD SAVE NO MAP
0278	0x2116	**Rhwconfig**	Hardware configuration B0 (1): 0= Not used B1 (2): Motor encoder direction (from firmware 6,15 moved to 10031) B2 (4): Motor rotation direction B3 (8): Phased encoder in FC reset flag B4 (16): Configure An. Inp 0 SMD1104 B5 (32): 0=Motor encoder in quadrature / 1=Motor encoder pulse/direction (from firmware 6,15 moved to 10031) B6 (64): 1=Preset encoder SSI B7 (128): 0=Normal SSI encoder / 1=Complement SSI encoder B8 (256): 0=Gray SSI encoder / 1=Binary SSI encoder (from firmware 6,15 moved to 10034-35) B9-10: 00=SSI encoder with right alignment data / 01=SSI encoder with left alignment data / 10=SSI encoder with centered data (from firmware 6,15 moved to 10034-35) B11 (2048): 0=DMD with PWM 50% / 1=DMD with PWM 0-100% B12 (4096): 1=Realign entering quota in interpolation mode (Master CanOpen Nardi) B13 (8096): Rialign quotas Rposact Rposactreq in interpolation mode (Master CanOpen Sipro)		0	65535	Unsig-ned16	WORD SAVE

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0063-64	0x203F	**Rvel**	Maximum translation speed	rev*100 /s	-10000	10000	Signed32	LS – MSWORD ANA MAP WRITE
0065-66	0x2041	**Rvss**	Start and stop speed (Start/Stop)	rev*100 /s	0	10000	Signed32	LS – MSWORD MAP WRITE
0067-68	0x2043	**Racc**	Acceleration ramp. If =0, ramp is disabled.	rev*10/ s²	0	200000	Unsig-ned32	LS – MSWORD ANA MAP WRITE
0069	0x2045	**Raccpro**	Acceleration profile 0=S 10=Linear		0	10	Unsig-ned16	WORD SAVE NO MAP
0070-71	0x2046	**Rdec**	Deceleration ramp. If =0, ramp is disabled.	rev*10/ s²	0	200000	Unsig-ned32	LS – MSWORD ANA MAP WRITE
0072	0x2048	**Rdecpro**	Deceleration profile 0=S 10=Linear		0	10	Unsig-ned16	WORD SAVE NO MAP
0073-74	0x2049	**Rdeceme**	Emergency deceleration ramp	rev*10/ s²	0	200000	Unsig-ned32	LS – MSWORD SAVE MAP WRITE
0075-76	0x204B	**Rvelact**	Actual speed of the motor	rev*100 /s			Signed32	LS – MSWORD MAP READ RO
0077-78	0x204D	**Rvelactreq**	Actual speed requested by the motor	rev*100 /s			Signed32	LS – MSWORD MAP READ RO
0079-80	0x204F	**Rvelmax**	Maximum speed during quota recovery	rev*100 /s	0	10000	Signed32	LS – MSWORD SAVE NO MAP
0100	0x2064	**Rdefum**	Define the velocity and acceleration divider. The following values can be taken: 1= 1:1 Ratio between the set value and the real value 10= 10:1 Ratio between the set value and the real value 100= 100:1 Ratio between the set value and the real value		1	100	Unsig-ned16	WORD NO MAP SAVE
0327	0x2147	**Rpwmacc**	Only for DMD. Acc/dec PWM ramp	bit/s²			Unsig-ned16	WORD
0328	0x2148	**Rpwm**	Only for DMD. PWM opening in PWM mode, expres-sed in bit. It can range from 0 (100%) to 1250 (100%)	%	0	1250	Unsig-ned16	WORD
0366	0x216E	**Rveladjpicur**	Maximum velocity for current PI reduction at low velocity		0	32767	Unsig-ned16	WORD SAVE
0406	0x2196	**Rdlyadjpicur**	PI current correction activation delay (0=disable correction)	ms	0	32767	Unsig-ned16	WORD SAVE
0410	0x219A	**Rzerovellim**	Minimum value of the motor encoder steps for axis in movement. Used to detect the motor in stop in homing in mecha-nical stop (hard-stop) in SmartMode	Enc. Pulses	0	65535	Unsig-ned16	WORD SAVE NO MAP
0411	0x219B	**Rzeroveltim**	Motor encoder sample time for stopped motor reading. Used to detect the motor in stop in homing in mecha-nical stop in SmartMode	ms	0	65535	Unsig-ned16	WORD SAVE NO MAP
0412	0x219C	**Rcurlimtim**	Filter time for the signal of current limit in FOC_CLO-SE	ms	0	65535	Unsig-ned16	WORD SAVE NO_MAP

0493-494	0x21ED	**Rveltarg**	Velocity target in function mode 9 (CSV). Only for protocols CANopen and EtherCAT.		-20000	+20000	Signed32	D WORD MAP WRITE

MOVEMENT COMMANDS

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0059	0x203B	**Rcmdwr**	PC writing/indexer reading command register B0 (1) : DISABLE DRIVE B1 (2) : ENABLE DRIVE B2 (4) : ABORT B3 (8) : STOP B4 (16) : ESTOP B5 (32) : JOG CW B6 (64) : JOG CCW B7 (128) : New setpoint GO B8 (256) : New setpoint GOR B9 (512) : HOME B10 (1024) : GEAR B11 (2048) : CAM Start Stop B12 (4096) : CAM B13 (8192) : TASK B14 (16384): BESTOP B15 (32768): SHIFT STOP **CAUTION! The bits of this register must be set one at a time; the drive, once the action is performed, will clear the the bit.**		0	65535	Unsig-ned16	WORD MAP WRITE WO
0060	0x203C	**Rcmd1wr**	PC writing/indexer writing command 1 register B0: Current pulse on phase A B1: Motor phases wiring test B2: Winding mode B3: Smit mode B4: Realign quotas in SmartMode (Remove possible thrust of the motor) B5: Not used B6: Set step/dir cam mode (function as step/dire mode, but without the need to pass from “Position” to “Step Dir”) B7: Recover the motor quota in SmartMode (if the motor has been shifted with the drive disabled) B8: B9: B10: B11: B12: B13: B14: B15:		0	65535	Unsig-ned16	WORD MAP WRITE WO

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0061	0x203D	**Rcmdrd**	PC reading/indexer writing command register B0: B1: B2: Positioning ABORT B3: Ramp movement STOP B4: ESTOP active B5: JOG + B6: JOG - B7: New setpoint GO B8: New setpoint GOR B9: Home B10: GEAR B11: CAM Start Stop Mode B12: CAM mode B13: TASK mode B14: BESTOP active B15:				Unsig-ned16	WORD MAP READ RO
0062	0x203E	**Rcmd1rd**	PC reading/indexer writing command 1 register B0: 1=Current step executed B1: 1=Motor phases wiring test executed (reset by writing 0 on Rcmd1wr or at the start of another test) B3: 1=Winding mode active B4: 1= Smit mode active B5: 1= Smart mode quotas realignment executed B6: Not used B7: 1= Step/dir cam mode active B8: 1= SmartMode motor quota recover executed B9: B10: B11: B12: B13: B14: B15:				Unsig-ned16	WORD MAP READ RO
0106	0x206A	**Rtasknum**	Number of the task to be executed with serial command		0	63	Unsig-ned16	WORD
0450	0x21C2	**Rptroldcmd**	Executed commands buffer pointer 0= Newest 7= Oldest		0	7	Unsig-ned16	WORD NO MAP
0451	0x21C3	**Rbufcmdsta-tus**	Status of the last command. Indicates where the command came from: User program Serial Fieldbus				Unsig-ned16	WORD NO MAP RO
0452-53	0x21C4	**Rbufcmdcom-mand**	Last command executed				Unsig-ned32	H-LWORD NO MAP RO
0454-55	0x21C6	**Rbufcmdpara-meter**	Parameter of the last command executed				Signed32	H-LWORD NO_MAP RO

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0032-33	0x2020	**Rhmaxspc**	Maxumum space in Home	Step	0	0x7FFFFFFF	Unsig-ned32	LS – MSWORD SAVE NO MAP
0034	0x2022	**Rdefinpbls**	Definition of input number for BLS function (back limit switch) 255 = Standard BLS input 254 = BLS input disabled 0-15 = Digital input used for BLS (if the input is not associated to a specific function)		0	255	Unsig-ned16	WORD SAVE
0035	0x2023	**Rdefinpfls**	Definition of input number for FLS function (forward limit switch) 255 = Standard FLS input 254 = FLS input disabled 0-15 = Digital input used for FLS (if the input is not associated to a specific function)		0	255	Unsig-ned16	WORD
0036-37	0x2024	**Rhofs**	Homing offset (Shift of the axis after home routine)	Step	0x80000000	0x7FFFFFFF	Signed32	LS – MSWORD SAVE NO MAP
0038-39	0x2026	**Rhpos**	Axis quota forced after the execution of homing function	Step	0x80000000	0x7FFFFFFF	Signed32	WORD SAVE
0040	0x2028	**Rhcurcoll**	% of nominal current to detect a collision during homing sequence with mechanical limit (FOC_CLOSE)	%	1	100	Unsig-ned16	WORD SAVE MAP-WRITE
0041	0x2029	**Rhtimcoll**	Filter time to detect a collision during homing sequence with mechanical limit (FOC_CLOSE)	ms	0	10000	Unsig-ned16	WORD NOMAP SAVE
0081	0x2051	**Rhtinv**	Stop time of the axis before inverting direction during homing in mS (default=512mS)	ms	0	32767	Unsig-ned16	WORD SAVE NO MAP

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0082	0x2052	**Rhmode**	Homing method (Type of home routine): -16 = Homing on FLS + encoder 1 TOP, positive direction -15 = Homing on FLS + encoder 1 TOP, negative direction -14 = Homing on FLS, positive direction -13 = Homing on FLS, negative direction -12 = Homing with forward mechanical limit + encoder 1 TOP (only SmartMode and Closed Loop) -11 = Homing with backward mechanical limit + encoder 1 TOP (only SmartMode and Closed Loop) -10 = Homing with forward mechanical limit -9 = Homing with backward mechanical limit -8 = Homing with forward mechanical limit + axis measure (Resets the registers Rlowlim and Rupplim) -7 = Homing with backward mechanical limit + axis measure (Resets the registers Rlowlim and Rupplim) -6 = Homing only with encoder 1 TOP in positive direction -5 = Homing only with encoder 1 TOP in negative direction -4 = Homing with BLS + encoder 1 TOP rising edge, positive direction -3 = Homing with BLS + encoder 1 TOP rising edge, negative direction -2 = Homing only with BLS in positive direction -1 = Homing only with BLS in negative direction 0 = Homing on place 35 = Homing on place (only in CanOpen, for DS402 compatibility) 37 = Homing on place (only in CanOpen, for DS402 compatibility)		-16	37	Signed16	WORD MAP WRITE
0083-84	0x2053	**Rhvh**	Homing speed during the limit switch search	rev*100 /s	0	10000	Signed32	LS– MSWORD MAP WRITE
0085-86	0x2055	**Rhvl**	Homing speed during the 0 point search (Must be a speed in the start/stop range)	rev*100 /s	0	10000	Signed32	LS – MSWORD MAP WRITE
0087-88	0x2057	**Rhacc**	Acceleration/deceleration during homing sequence	rev*10/ s^2	0	200000	Unsig-ned32	LS – MSWORD MAP WRITE
0202	0x20CA	**Rhsts**	Homing sequence error code Bit 0: Home in progress Bit 1: Drive not enabled Bit 2: Maximum space for homing sequence Bit 3: BLS intervention error Bit 4: FLS intervention error Bit 5: Home interrupted Bit 6: Homing sequence not recognized Bit 7: Homing with mechanical limit (with FOC_OPEN) Bit 8: Bit 9: Bit 10: Bit 11: Bit 12: Bit 13: Bit 14: Bit 15:				Unsig-ned16	WORD NO MAP RO
0575-76	0x223F	**Rhindexspc**		Step			Signed32	D WORD

STOP INSTRUCTION

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0024-25	0x2018	**Rshstop**	Shift space in STOP instruction	Step	0	0x7FFFFFFF	Unsig-ned32	LS– MSWORD SAVE NO MAP
0028-29	0x201C	**Rspcstop**	Fixed stop space	Step	0	0x7FFFFFFF	Unsig-ned32	LS – MSWORD SAVE NO MAP
0030-31	0x201E	**Rspcstopcalc**	Stop space of the last stop	Step			Unsig-ned32	LS – MSWORD NO MAP

BESTOP INSTRUCTION (STOP ON BIT EVENT)

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0049	0x2031	**Rbestpflg**	BESTOP command enabling [B0 ..B4]: Bit Number B5: Condition (0 = Low, 1 = High) B6: Type of data (0 = Variable, 1= Register) B7 = Stop on the rising edge of TOP mot B8 = Stop on the falling edge of TOP mot B9 = Stop on the rising edge of TOP ext B10= Stop on the falling edge of TOP ext				Unsig-ned16	WORD MAP WRITE
0050	0x2032	**Rbestppar**	BESTOP parameter, contains the register or variable that generates ESTOP				Unsig-ned16	WORD MAP WRITE

N.B. In order to activate the BEstop functions, it is necessary to raise the bit 14 of the register Rcmdwr (address 0059).

ESTOP INSTRUCTION (STOP ON REGISTER OR VARIABLE VALUE)

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0020-21	0x2014	**Rshestop**	Shift space in ESTOP instruction	Step	0	0x7FFFFFFF	Unsig-ned32	LS – MSWORD ANA_T - SAVE NO MAP
0045	0x202D	**Restpflg**	ESTOP command enabling B7: Type of destination data (0 = Variabie, 1= Regi-ster) B6-B5: Source: 00=Var / 01=Reg / 10=direct value / 11=not allowed. B4-B3-B2-B1: Jump condition 0000 = Equal 0001 = Not Equal 0010 = Higher 0011 = Lower 0100 = Higher or Same 0101 = Lower or Same B0: Free				Unsig-ned16	WORD MAP WRITE
0046-47	0x202E	**Restppar1**	Parameter 1 for ESTOP function				Unsig-ned32	WORD MAP WRITE
0048	0x2030	**Restppar2**	Parameter 2 for ESTOP function				Unsig-ned16	WORD MAP WRITE

N.B. In order to activate the Estop functions, it is necessary to raise the bit 4 of the register Rcmdwr (address 0059).

GEAR INSTRUCTION

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0104	0x2068	**Rgearmul**	Reduction ratio for GEAR instruction (multiplier)		1	32767	Unsig-ned16	WORD SAVE NO MAP
0105	0x2069	**Rgeardiv**	Reduction ratio for GEAR instruction (divider)		1	32767	Unsig-ned16	WORD SAVE NO MAP

ALARMS AND WARNINGS

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0224	0x20E0	**Rpostimeout**	Time for positioning Time-out	ms	0	65535	Unsig-ned16	WORD SAVE
0225	0x20E1	**Rdeadpos**	Dead band in position	Step motore	0	32767	Unsig-ned16	WORD SAVE NO MAP
0226	0x20E2	**Rsettim**	Settling time in closed loop. Time in the range of motor position with requested theoric speed = 0, before the signal of motor in posi-tion.	ms	0	1000	Unsig-ned16	WORD SAVE NO MAP
0227	0x20E3	**Ralarm**	Drive alarms Bit 0: Overcurrent HW (not maskable) Bit 1: Overcurrent SW (not maskable) Bit 2: I2T Bit 3: Positioning error (Closed Loop. Position out of the DeadBand for time Rposalmtime with requested velocity=0) (disabled by default) Bit 4: Following error (Open Loop. Encoder pulses – normalized motor pul-ses greater than Rflwmax) (disabled by default) (Closed Loop. With requested velocity <>0 (Encoder pulses – normalized motor pulses) > Rflwmax for the time Rposalmtime) Bit 5: Overload digital output (not maskable) Bit 6: Overtemperature (not maskable) Bit 7: Overvoltage (not maskable) Bit 8: Undervoltage Bit 9: Motor encoder phasing error (not maskable) Bit 10: Motor phase A disconnected (not maskable) Bit 11: Motor phase B disconnected (not maskable) Bit 12: Positioning Timeout Bit 13: Homing Error Bit 14: Inverted encoder in Smart Mode or Closed Loop Bit 15: Encoder anomaly Note: write 0 into this register to reset the alarms.		0	65535	Unsig-ned16	WORD MAP READ

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0228	0x20E4	**Rwarning**	Drive pre-alarm notifications Bit 0: Overcurrent HW (= alarm) (not maskable) Bit 1: Overcurrent SW (not maskable) Bit 2: I2T Bit 3: Positioning error (Closed Loop. Position out of the DeadBand for time Rposalmtime with requested velocity=0) (disabled by default) Bit 4: Following error (Open Loop. Encoder pulses – normalized motor pulses greater than Rflwwrn) (disabled by default) (Closed Loop. With requested velocity <>0 (Encoder pulses – normalized motor pulses) > Rflwmax for the time Rposwrntime) Bit 5: Overload digital output (not maskable) Bit 6: Overtemperature (not maskable) Bit 7: Overvoltage (not maskable) Bit 8: Undervoltage Bit 9: Current limitated by voltage Bit 10: Saturated regulator Bit 11: Current limit is active Bit 12: Positioning Timeout Bit 13: Free Bit 14: Free Bit 15: Free		0	65535	Unsig-ned16	WORD MAP READ RO
0229	0x20E5	**Rbufalm0**	Alarm buffer 0				Unsig-ned16	WORD NO MAP RO
0230	0x20E6	**Rbufalm1**	Alarm buffer 1				Unsig-ned16	WORD NO MAP RO
0231	0x20E7	**Rbufalm2**	Alarm buffer 2				Unsig-ned16	WORD NO MAP RO
0232	0x20E8	**Rbufalm3**	Alarm buffer 3				Unsig-ned16	WORD NO MAP RO
0233	0x20E9	**Rbufalm4**	Alarm buffer 4				Unsig-ned16	WORD NO MAP RO
0234	0x20EA	**Rbufalm5**	Alarm buffer 5				Unsig-ned16	WORD NO MAP RO
0235	0x20EB	**Rbufalm6**	Alarm buffer 6				Unsig-ned16	WORD NO MAP RO
0236	0x20EC	**Rbufalm7**	Alarm buffer 7				Unsig-ned16	WORD NO MAP RO
0237	0x20ED	**Ralmcnt**	Fault counter		0	65535	Unsig-ned16	WORD NO MAP RO
0238	0x20EE	**Ralmack**	Last alarm acknowledge Bit 0: Alarm Acknowledge Bit 1: Reset alarm counter		0	3	Unsig-ned16	WORD NO MAP
0239	0x20EF	**Rtempalm**	Temperature limit to be reached to activate the alarm. Exceeded this value, an alarm is generated.	°C	0	150	Unsig-ned16	WORD SAVE NO MAP
0240	0x20F0	**Rtensmax**	Maximum voltage limit. Exceeded this value, an alarm is generated.	Volt	0	200	Unsig-ned16	WORD SAVE NO MAP
0241	0x20F1	**Rtensmin**	Minimum voltage limit. Below this value, an alarm is generated.	Volt	0	200	Unsig-ned16	WORD SAVE NO MAP
0242	0x20F2	**Rcurmax**	Maximum current limit.	mA	0	20000	Unsig-ned16	WORD SAVE NO MAP

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0243-244	0x20F3	**Rflwmax**	Maximum delta for the generation of a following alarm.	Imp. Enco-der	0	32000	Unsig-ned32	LS – MSWORD SAVE NO MAP
0245-246	0x20F5	**Ri2tmax**	I2T for alarm		0	9999999	Unsig-ned32	LSWORD – MSWORD SAVE NO MAP
0247	0x20F7	**Rmaskalm**	Alarms mask. If bit=1 the corresponding alarm is masked. Bit 0: Not maskable Bit 1: Not maskable Bit 2: I2T Bit 3: Positioning error (Closed Loop. Position out of the DeadBand for time Rposalmtime with requested velocity=0) (disabled by default) Bit 4: Following error (Open Loop. Encoder pulses – normalized motor pulses greater than Rflwmax) (disabled by default) (Closed Loop. With requested velocity <>0 (Encoder pulses – normalized motor pulses) > Rflwmax for the time Rposalmtime) Bit 5: Not maskable Bit 6: Not maskable Bit 7: Not maskable Bit 8: Undervoltage Bit 9: Not maskable Bit 10: Not maskable Bit 11: Not maskable Bit 12: Positioning Timeout Bit 13: Not maskable Bit 14: Not maskable Bit 15: Encoder anomaly				Unsig-ned16	WORD SAVE
0248	0x20F8	**Rposalmti-me**	Time before positioning alarm in Closed Loop.	ms	0	32000	Unsig-ned16	WORD SAVE
0249	0x20F9	**Rtempwrn**	Value for overtemperature warning	°C	0	150	Unsig-ned16	WORD SAVE NO MAP
0250	0x20FA	**Rovvwrn**	Value for overvoltage warning	Volt	0	200	Unsig-ned16	WORD SAVE NO MAP
0251	0x20FB	**Runvwrn**	Value for undervoltage warning	Volt	0	200	Unsig-ned16	WORD SAVE NO MAP
0252	0x20FC	**Rovcwrn**	Value for overcurrent warning	mA	0	20000	Unsig-ned16	WORD SAVE NO MAP
0253-254	0x20FD	**Rflwwrn**	Value for following warning	Step	0	32000	Unsig-ned32	LSWORD - MSWORD SAVE NO MAP
0255-256	0x20FF	**Ri2twrn**	Value for I2T warning		0	9999999	Unsig-ned32	LSWORD - MSWORD SAVE NO MAP

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0257	0x2101	**Rmaskwrn**	Warning mask. If bit=1 the corresponding warning is masked. Bit 0: Not maskable Bit 1: Not maskable Bit 2: I2T Bit 3: Positioning error (Closed Loop. Position out of the DeadBand for time Rposalmtime with requested velocity=0) (disabled by default) Bit 4: Following error (Open Loop. Encoder pulses – normalized motor pulses greater than Rflwwrn) (disabled by default) (Closed Loop. With requested velocity <>0 (Encoder pulses – normalized motor pulses) > Rflwmax for the time Rposwrntime) Bit 5: Not maskable Bit 6: Not maskable Bit 7: Not maskable Bit 8: Undervoltage Bit 9: Current limitated by voltage Bit 10: Saturated regulator Bit 11: Current limit is active Bit 12: Positioning Timeout Bit 13: Free Bit 14: Free Bit 15: Free				Unsig-ned16	WORD SAVE
0258	0x2102	**Rposwrn-time**	Time before positioning warning in Closed Loop.	ms	0	32000	Unsig-ned16	WORD SAVE
0259-260	0x2103	**Rflwdisp**	Display the absolute following error. Write 0 in this register to reset the following alarm.	En-coder pulses			Signed32	LSWORD – MSWORD MAP READ
0261-262	0x2105	**Rflwmem**	Display the maximum saved following error. Write 0 in this register to reset.	En-coder pulses			Signed32	LSWORD – MSWORD NO MAP
0263	0x2107	**Rflwtim**	Filter time before the signal of following error.	ms	0	32000	Unsig-ned16	WORD SAVE NO MAP
0266	0x210A	**Rflwmemp**	Store the maximum positive following error at 16 bit. Value goes from 0 to 65535. Write 0 in this location to reset the memory.	En-coder pulses	0	65535	Unsig-ned16	WORD NO MAP
0267	0x210B	**Rflwmemn**	Store the maximum negative following error at 16 bit. Value goes from 0 to 65535. Write 0 in this location to reset the memory.	En-coder pulses	0	65535	Unsig-ned16	WORD NO MAP
0407-08	0x2197	**Rflwencerr**	Maximum error of counter difference between motor encoder and normalized motor steps to generate an error. This function is similar to the followng error, but it doesn't have a time filter, and it's always active. It's used to detect the encoder absence. If used, it's usually set at a value of 1 or 2 turns of the encoder.	En-coder pulses	0	0x7FFFFFFF	Unsig-ned32	WORD
0409	0x2199	**Rposti-meoutwrn**	Time for positioning timeout warning	ms	0	65535	Unsig-ned16	WORD SAVE

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0559-60	0x222F	**Ralarm32**	Drive alarms Bit 0: Overcurrent HW (not maskable) Bit 1: Overcurrent SW (not maskable) Bit 2: I2T Bit 3: Positioning error (Closed Loop. Position out of the DeadBand for time Rposalmtime with requested velocity=0) (disabled by default) Bit 4: Following error (Open Loop. Encoder pulses – normalized motor pulses greater than Rflwmax) (disabled by default) (Closed Loop. With requested velocity <>0 (Encoder pulses – normalized motor pulses) > Rflwmax for the time Rposalmtime) Bit 5: Overload digital output (not maskable) Bit 6: Overtemperature (not maskable) Bit 7: Overvoltage (not maskable) Bit 8: Undervoltage Bit 9: Motor encoder phasing error (not maskable) Bit 10: Motor phase A disconnected (not maskable) Bit 11: Motor phase B disconnected (not maskable) Bit 12: Positioning Timeout Bit 13: Homing Error Bit 14: Inverted encoder in Smart Mode or Closed Loop Bit 15: Encoder anomaly Bit 16: Reserved Bit 17: Reserved Bit 18: Missing Sync (CANopen or EtherCAT) Bit 19: STO anomaly (Incongruent inputs) Bit 20: VLogic undervoltage alarm Bit 21: STO anomaly (test on STO A failed) Bit 22: STO anomaly (test on STO B failed) Bit 23: Anomaly on output pin “Drive safety state” Note: write 0 into this register to reset the alarms.		0	0xFFFFFFF	Unsig-ned32	WORD MAP READ
0561-62	0x2231	**Rwarning32**	Drive pre-alarm notifications Bit 0: Overcurrent HW (= alarm) (not maskable) Bit 1: Overcurrent SW (not maskable) Bit 2: I2T Bit 3: Positioning error (Closed Loop. Position out of the DeadBand for time Rposalmtime with requested velocity=0) (disabled by default) Bit 4: Following error (Open Loop. Encoder pulses – normalized motor pulses greater than Rflwwrn) (disabled by default) (Closed Loop. With requested velocity <>0 (Encoder pulses – normalized motor pulses) > Rflwmax for the time Rposwrntime) Bit 5: Overload digital output (not maskable) Bit 6: Overtemperature (not maskable) Bit 7: Overvoltage (not maskable) Bit 8: Undervoltage Bit 9: Current limitated by voltage Bit 10: Saturated regulator Bit 11: Current limit is active Bit 12: Positioning Timeout Bit 13: Free Bit 14: Free Bit 15: Free Bit 16: Reserved Bit 17: Reserved Bit 18: Missing Sync (CANopen or EtherCAT) Bit 19: STO anomaly		0	65536	Unsig-ned32	WORD MAP READ
10144	0x4800	**Ralmbuffres**	Alarms history buffer reset				Unsig-ned16	RW SAVE
10145	0x4801	**Ralmbuffptr**	Pointer to the alarms history buffer		0	15	Unsig-ned16	RW SAVE
10146-47	0x4802	**Ralmbuffti-me**	Alarm timestamp pointed by Ralmbuffptr				Unsig-ned32	RO

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
10148-49	0x4803	**Ralmbuffco-de**	Alarm code pointed by Ralmbuffptr				Unsig-ned32	RO

Management of following control in closed loop

When the motor works in closed loop, the motor must be equipped with an encoder, so it is possible to execute a following control between the motor and the encoder.
In closed loop mode, there are two types of alarms generated by the comparison between the encoder position (real) and the position requested by the program (target).
If a movement of the motor is requested (speed different than 0) the following control in enabled.
When the theoric positioning profile ends, and the requested speed is equal to 0, it is necessary to wait the motor enters in the positioning zone indicated in the register Rdeadpos.
When the motor remains in this positioning zone for the time indicated in the register Rsettim, the flag of positioned motor is activated.
At this point, the positioning control enters in function and the following control is disabled.

The positioning control checks that the motor doesn't exit the zone indicated in the registers Rdeadpos.
In the motor exits this zone, a "motor out of position" timer starts.
After the time set in the register Rposwrntime, the bit 3 of the register Rwarning is activated to notify the waring of "motor out of position". If the motor returns to the positioning zone, the warining is automatically reset.
By setting to 1 the bit 3 of the register Rmaskwrn, this notification is disabled.
If the "motor out of position" timer reaches the value set in the register Rposalmtime, the bit 3 of the register Ralarm is activated, the drive enters into alarm state and is disabled. At this stage, an action must be taken to reset the alarm and to restart drive.
By setting to 1 the bit 3 of the register Rmaskalm, this alarm is disabled.

The following control is a continuous comparison between the actual target quota and the real quota of the encoder.
If the difference between these two values exceeds the value set in the register Rflwwrn, the bit 4 of the register Rwarning is set to 1. If the motor returns inside the range, the warining is automatically reset.
By setting to 1 the bit 4 of the register Rmaskwrn, this notification is disabled.
If the difference between the two values exceeds the value set in the register Rflwmax, a filter time is activated for the time set in the register Rflwtim. After this filter time, if the error still exceeds the value of Rflwmax, the bit 4 of the register Ralarm is activated, the drive enters into alarm state and is disabled. At this stage, an action must be taken to reset the alarm and to restart drive.
By setting to 1 the bit 4 of the register Rmaskalm, this alarm is disabled.

The waring is just an anomaly notification, and it doesn't perform any action.

When the drive is disabled, the following error is reset.
To disable the control of the positioning warning, set to 0 the register Rposwrntime.
To disable the control of the positioning error, set to 0 the register Rposalmtime.
To disable the control of the following warning, set to 0 the register Rflwwrn.
To disable the control of the following error, set to 0 the register Rflwmax.

The following registers must be set in order to use the function of following control.
Setting:
Rmotenc : Motor encoder pulses/revolution
Rflwwrn : Following error absolute maximum value in encoder pulses, to activate the warning notification. The value of this register is stored into the drive.
Rflwmax : Following error absolute maximum value in encoder pulses, to activate the error notification (after the filter time). The value of this register is stored into the drive.
Rflwtim : Filter time before the notification of the following error. The warning is immediately displyed , without filter time. The value of this register is stored into the drive.
Rposalmtime : Time before the notification positioning alarm.
Rposwrntime : Time before the notification positioning warning.
Rmaskwrn (b3) : Disable the notification of the positioning warning in the register Rwarning.
Rmaskalm (b3) : Disable the notification of the positioning alarm in the register Ralarm. Avoids the drive to enter the alarm mode, and the consequent deactivation of the motor.
Rmaskwrn (b4) : Disable the notification of the following warning in the register Rwarning.
Rmaskalm (b4) : Disable the notification of the following alarm in the register Ralarm. Avoids the drive to enter the alarm mode, and the consequent deactivation of the motor.

Displaying:
Rflwdisp : Displays the actual following error in encoder pulses.
Rflwmem : Displays the maximum following error in encoder pulses (absolute value). To reset this value, write 0 in the register.
Rflwmemp : Displays the positive maximum following error in encoder pulses (displayed in positive value). To reset this value, write 0 in the register.
Rflwmemn : Displays the negative maximum following error in encoder pulses (displayed in positive value). To reset this value, write 0 in the register.

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0107	0x206B	**Rhslsi (50us)**	High Speed Limit switch input. Bit 0: High Speed FLS limit switch (forward) Bit 1: High Speed BLS limit switch in (backward) Bit 2: High Speed TOP motor encoder Bit 3: High Speed CH.A motor encoder Bit 4: High Speed CH.B motor encoder Bit 5: High Speed TOP external encoder Bit 6: High Speed CH.A external encoder Bit 7: High Speed CH.B external encoder				Unsig-ned16	WORD MAP READ RO
0108	0x206C	**Rhsinp (50us)**	High speed digital input. Bit 0: High speed digital Input 0 Bit 1: High speed digital Input 1 Bit 2: High speed digital Input 2 Bit 3: High speed digital Input 3 Bit 4: High speed digital Input 4 Bit 5: High speed digital Input 5 Bit 6: High speed digital Input 6 Bit 7: High speed digital Input 7 Bit 8: High speed digital Input 8 Bit 9: High speed digital Input 9 Bit 10: High speed digital Input 10 Bit 11: High speed digital Input 11 Bit 12: High speed digital Input 12 Bit 13: High speed digital Input 13 Bit 14: High speed digital Input 14 Bit 15: High speed digital Input 15				Unsig-ned16	WORD MAP READ RO
0109	0x206D	**Rlsi**	Limit switch input. Bit 0: FLS overtravel (forward) Bit 1: BLS overtravel (backward) Bit 2: TOP motor encoder Bit 3: CH.A motor encoder Bit 4: CH.B motor encoder Bit 5: TOP external encoder Bit 6: CH.A external encoder Bit 7: CH.B external encoder				Unsig-ned16	WORD MAP READ RO
0110	0x206E	**Rdeflsi**	Service inputs “active state” definition (0=active high; 1=active low). Bit 0: FLS overtravel (forward) Bit 1: BLS overtravel (backward) Bit 2: TOP motor encoder. This input is ALWAYS active on the rising edge in case it is used with HOME, ESTOP instructions. Else, it works like other inputs. Bit 3: CH.A motor encoder Bit 4: CH.B motor encoder Bit 5: TOP external encoder Bit 6: CH.A external encoder Bit 7: CH.B external encoder				Unsig-ned16	WORD SAVE NO MAP
0111	0x206F	**Rfillsi**	Limit switch input digital filter time	ms	1	16	Unsig-ned16	WORD SAVE NO MAP
0112	0x2070	**Renflsi**	Limit switch input digital filter time enabling				Unsig-ned16	WORD SAVE NO MAP
0113	0x2071	**Rmemlsi**	Limit switch input memory				Unsig-ned16	WORD NO MAP

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0114	0x2072	**Rinp**	Digital input. Bit 0: Input 0 Bit 1: Input 1 Bit 2: Input 2 Bit 3: Input 3 Bit 4: Input 4 Bit 5: Input 5 Bit 6: Input 6 Bit 7: Input 7 Bit 8: Input 8 Bit 9: Input 9 Bit 10: Input 10 Bit 11: Input 11 Bit 12: Input 12 Bit 13: Input 13 Bit 14: Input 14 Bit 15: Input 15				Unsig-ned16	WORD MAP READ RO
0115	0x2073	**Rdefinp**	Digital input “active state” definition (0=active high; 1=active low). Bit 0: Input 0 .... Bit 15: Input 15				Unsig-ned16	WORD SAVE NO MAP
0116	0x2074	**Rfilinp**	Digital input digital filter time	ms	1	16	Unsig-ned16	WORD SAVE NO MAP
0117	0x2075	**Renfinp**	Filter enabling on digital input Bit 0: Input 0 .... Bit 15: Input 15				Unsig-ned16	WORD SAVE NO MAP
0118	0x2076	**Rmeminp**	Digital input memory Bit 0: Input 0 .... Bit 15: Input 15				Unsig-ned16	WORD NO MAP
0119	0x2077	**Rout**	Digital output Bit0: Output0 Bit1: Output1 Bit2: Output2 Bit3: Output3 Bit4: Output4 Bit5: Output5 Bit6: Output6 Bit7: Output7				Unsig-ned16	WORD MAP WRITE
0120	0x2078	**Rdefout**	Digital output “active state” definition (0=active high; 1=active low). Bit0: Output0 .... Bit7: Output7				Unsig-ned16	WORD SAVE NO MAP
0121	0x2079	**Rfuno0**	Digital output 0 function definition register 0) Normal digital output 1) Drive enabled 2) Drive alarm 3) Synchronized motor (Home executed) 4) Motor in movement 5) Task in progress 6) Alarm I^2/T 7) Motor in position 8) Motor in actual movement (for closed loop) 9) Motore in theoric+actual movement (for closed loop) 10) Command for external brake 11) Signal of changed quota while the drive was disabled (only with encoder) 12) Signal of changed quota while the drive was disabled (only with encoder) + motor in position (When the drive is enabled: output=0 if motor not in position or moved while the drive was disabled / =1 if motor in position and not been moved while the drive was disabled). 13) Virtual motor step signal output 14) Motor direction signal output 15) STO output		0	14	Unsig-ned16	WORD SAVE NO MAP

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0122	0x207A	**Rfuno1**	Digital output 1 function definition register (See Rfun0 for details)		0	14	Unsig-ned16	WORD SAVE NO MAP
0123	0x207B	**Rfuno2**	Digital output 2 function definition register (See Rfun0 for details)		0	14	Unsig-ned16	WORD SAVE NO MAP
0124	0x207C	**Rfuno3**	Digital output 3 function definition register (See Rfun0 for details)		0	14	Unsig-ned16	WORD SAVE NO MAP
0125	0x207D	**Rfuno4**	Digital output 4 function definition register (See Rfun0 for details)		0	14	Unsig-ned16	WORD SAVE NO MAP
0126	0x207E	**Rfuno5**	Digital output 5 function definition register (See Rfun0 for details)		0	14	Unsig-ned16	WORD SAVE NO MAP
0127	0x207F	**Rfuno6**	Digital output 6 function definition register (See Rfun0 for details)		0	14	Unsig-ned16	WORD SAVE NO MAP
0128	0x2080	**Rfuno7**	Digital output 7 function definition register (See Rfun0 for details)		0	14	Unsig-ned16	WORD SAVE NO MAP
0129	0x2081	**Rfuni0**	Digital input 0 function definition register 0) Normal digital input 1) Enable/disable drive 2) JOG forward 3) JOG backward 4) GO (Quota set in Rpostarg) 5) GOR 6) HOME 7) Bit 0 number of task to be enabled 8) Bit 1 number of task to be enabled 9) Bit 2 number of task to be enabled 10) Bit 3 number of task to be enabled 11) Bit 4 number of task to be enabled 12) Bit 5 number of task to be enabled 13) Bit 6 number of task to be enabled 14) Start task (enable selected task) 15) Alarms reset 16) Quota Line Up 17) Current reduction 18) ABORT 19) STOP 20) GEAR 21) Direction (reverse the JOG direction) 22) Position recovery (only with encoder)		0	22	Unsig-ned16	WORD SAVE NO MAP
0130	0x2082	**Rfuni1**	Digital input 1 function definition register (See Rfuni0 for details)		0	22	Unsig-ned16	WORD SAVE NO MAP
0131	0x2083	**Rfuni2**	Digital input 2 function definition register (See Rfuni0 for details)		0	22	Unsig-ned16	WORD SAVE NO MAP
0132	0x2084	**Rfuni3**	Digital input 3 function definition register (See Rfuni0 for details)		0	22	Unsig-ned16	WORD SAVE NO MAP
0133	0x2085	**Rfuni4**	Digital input 4 function definition register (See Rfuni0 for details)		0	22	Unsig-ned16	WORD SAVE NO MAP
0134	0x2086	**Rfuni5**	Digital input 5 function definition register (See Rfuni0 for details)		0	22	Unsig-ned16	WORD SAVE NO MAP
0135	0x2087	**Rfuni6**	Digital input 6 function definition register (See Rfuni0 for details)		0	22	Unsig-ned16	WORD SAVE NO MAP
0136	0x2088	**Rfuni7**	Digital input 7 function definition register (See Rfuni0 for details)		0	22	Unsig-ned16	WORD SAVE NO MAP

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0297	0x2129	**Rdiginplev**	Selection of digital inputs trigger level. 0=Trigger at 12V (for inputs at 24V) 1=Trigger at 2.5V (for inputs at 5V) The thresholds of the levels selected with this register are set in the registers 306 Rtrginp5v and 307 Rtrginp24v		0	1	Unsigned16	WORD
0298	0x212A	**Rstpoutmaxfreq**	Maxumim frequency in digital output with step function (Theoric rotation step frequency of the motor) If the real frequency is higher than the maximum frequency, the output stops at the maximum frequency. The accumulated pulses are given in output when the motor is already in stop, or when its speed drops to a theoric frequency lower than the maximum frequency.	Hz	0	10000	Unsigned16	WORD
0306	0x2132	**Rtrginp5v**	Set the voltage threshold for inputs at 5V. The save is performed with the command 145 in Rloadsav.	Volt *100	0	1755	Unsigned16	WORD SAVE NO MAP NO RES
0307	0x2133	**Rtrginp24v**	Set the voltage threshold for inputs at 24V. The save is performed with the command 145 in Rloadsav.	Volt *100	0	1755	Unsigned16	WORD SAVE NO MAP NO RES
0442	0x21BA	**Rfuni8**	Digital input function definition		0	22	Unsigned16	WORD SAVE NO MAP
0443	0x21BB	**Rfuni9**	Digital input function definition		0	22	Unsigned16	WORD SAVE NO MAP
0444	0x21BC	**Rfuni10**	Digital input function definition		0	22	Unsigned16	WORD SAVE NO MAP
0445	0x21BD	**Rfuni11**	Digital input function definition		0	22	Unsigned16	WORD SAVE NO MAP
0446	0x21BE	**Rfuni12**	Digital input function definition		0	22	Unsigned16	WORD SAVE NO MAP
0447	0x21BF	**Rfuni13**	Digital input function definition		0	22	Unsigned16	WORD SAVE NO MAP
0448	0x21C0	**Rfuni14**	Digital input function definition		0	22	Unsigned16	WORD SAVE NO MAP
0449	0x21C1	**Rfuni15**	Digital input function definition		0	22	Unsigned16	WORD SAVE NO MAP
10151	0x4807	**Rstepouttime**	Stepout output activation time Rstepouttime=(register value * 50uS)		0	65535	Unsigned16	RW SAVE
10152-53	0x4808	**Rstepoutcount**	Number of steps to count before activationg the output		1	0xFFFFFFFF	Unsigned32	RW SAVE

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0137	0x2089	**Ranainp**	12 bit analog input register				Signed16	WORD MAP READ RO
0138	0x208A	**Rdefanainp**	Analog input definition register		0	558	Unsig-ned16	WORD SAVE NO MAP
0139	0x208B	**Rmulanainp**	Analog input multiplier register		1	32767	Unsig-ned16	WORD SAVE NO MAP
0140	0x208C	**Rdivanainp**	Analog input divider register		1	32767	Unsig-ned16	WORD SAVE NO MAP
0141	0x208D	**Roffsanainp**	Analog input offset		-32768	32767	Signed16	WORD SAVE NO MAP
0142	0x208E	**Rdeada-nainp**	Analog input dead band	Bit	0	4095	Unsig-ned16	WORD SAVE NO MAP
0143	0x208F	**Ranaout**	10 bit analog output register	Bit	0	1023	Unsig-ned16	WORD MAP WRITE
0144	0x2090	**Rdefanaout**	Parameter definition to be used for Ranaout register		0	558	Unsig-ned16	WORD SAVE NO MAP
0145	0x2091	**Rmulanaout**	Multiplier of the value to be set in the analog output		1	32767	Unsig-ned16	WORD SAVE NO MAP
0146	0x2092	**Rdivanaout**	Division of the value to be set in the analog output		1	32767	Unsig-ned16	WORD SAVE NO MAP
0147	0x2093	**Roffsanaout**	Offset to be added to the value to be set in the analog output		-32768	32767	Signed16	WORD NO MAP SAVE
0313	0x2139	**Rfloatingrol-lerperc**	Only for DMD. Dancer roller percentage (-100/+100)	%	-100	100	Signed16	WORD SAVE
0429	0x21AD	**Ranainpdi-sable**	Disable the analog inputs 0-1-2 with respective bits B0, B1 and B2 high				Unsig-ned16	
0430	0x21AE	**Ranainp1**	Analog input 1 value (only for SMD10.04 and SMD1104)				Signed16	WORD RO MAP READ
0431	0x21AF	**Rdefanainp1**	Analog input 1 function definition (only for SMD10.04 and SMD1104)		0	558	Unsig-ned16	WORD SAVE
0432	0x21B0	**Rmula-nainp1**	Analog input 1 multiplier (only for SMD1004 and SMD1104)		1	32767	Unsig-ned16	WORD SAVE
0433	0x21B1	**Rdivanainp1**	Analog input 1 divider (only for SMD1004 and SMD1104)		1	32767	Unsig-ned16	WORD SAVE
0434	0x21B2	**Roffsa-nainp1**	Analog input 1 offset register (only for SMD10.04 and SMD1104)		-32768	32767	Signed16	WORD SAVE
0435	0x21B3	**Rdeada-nainp1**	Analog input 1 dead band (only for SMD10.04 and SMD1104)	Bit	0	4095	Unsig-ned16	WORD SAVE
0436	0x21B4	**Ranainp2**	Analog input 2 value (only for SMD10.04 and SMD1104)				Signed16	WORD RO MAP READ
0437	0x21B5	**Rdefanainp2**	Analog input 2 function definition(only for SMD10.04 and SMD1104)		0	558	Unsig-ned16	WORD SAVE
0438	0x21B6	**Rmula-nainp2**	Analog input 2 multiplier (only for SMD10.04 and SMD1104)		1	32767	Unsig-ned16	WORD SAVE
0439	0x21B7	**Rdivanainp2**	Analog input 2 divider (only for SMD10.04 and SMD1104)		1	32767	Unsig-ned16	WORD SAVE
0440	0x21B8	**Roffsa-nainp2**	Analog input 2 offset register (only for SMD10.04 and SMD1104)		-32768	32767	Signed16	WORD SAVE

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0441	0x21B9	**Rdeada-nainp2**	Analog input 2 dead band (only for SMD10.04 and SMD1104)	Bit	0	4095	Unsig-ned16	WORD SAVE

ENCODER MANAGEMENT

Indirizzo Modbus / Profibus	Indirizzo CAN EtherCAT Profinet	Nome Regi-stro	Descrizione	U.M.	Valore Minimo	Valore Mas-simo	Tipo	Note
10010	0x471A	**Rencmotfbk**	Motor encoder feedback for loop closure 0= Encoder 1 1= Encoder 2 2= Encoder 3 3= Absolute ncoder		Default 0		Unsign16	WORD RW SAVE
10043	0x473B	**Rencnewpe-riod**	Signal of reading of new encoder pulses period (previously register 165) Bit 0: New period encoder 1 Bit 1: New period encoder 2 Bit 2: New period encoder 3				Unsig-ned16	RW

ENCODER 1

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0151-152	0x2097	**Rmotenc**	Motor encoder quota register. **From firmware rev. 6 replaced by register 10015**	Enc. Pulses			Signed32	LSWORD – MSWORD MAP READ
0163	0x20A3	**Rmotencper**	Period read by the pulses of the motor encoder. **From firmware rev. 6 replaced by register 10040**				Unsig-ned16	WORD
0269	0x210D	**Rmo-tencpuls**	Motor encoder pulses per revolution. **From firmware rev. 6 replaced by register 10025**	Enc. Pulses	1	32767	Unsig-ned16	WORD SAVE NO MAP
10015-16	0x471F	**Renc1pos**	Encoder 1 quota	Enc. Pulses			Signed32	DWORD RW MAP
10025	0x4729	**Renc1pulse**	Encoder 1 pulses/turn	Pulses			Unsign16	RW SAVE
10040	0x4738	**Renc1period**	Period read by encoder 1 pulses				Unsig-ned16	RO
10044	0x473C	**Renc1vel**	Encoder 1 actual velocity				Signed16	RO
10045	0x473D	**Renc1sam-ple**	Encoder 1 sample time	ms			Unsig-ned16	RW SAVE
10046	0x473E	**Renc1vel-mul**	Encoder 1 velocity multiplier				Unsig-ned16	RW SAVE
10047	0x473F	**Renc1veldiv**	Encoder 1 velocity divider				Unsig-ned16	RW SAVE
10048-49	0x4740	**Renc1topcnt**	Counter of pulses received on encoder 1 TOP				Unsig-ned32	DWORD RW
10050	0x4742	**Renc1topvel**	Velocity detected on encoder 1 TOP input				Unsig-ned16	RO
10051	0x4743	**Renc1top-sample**	Encoder 1 TOP input sample time	ms			Unsig-ned16	RW SAVE
10052	0x4744	**Ren-c1topvelmul**	Encoder 1 TOP input velocity multiplier				Unsig-ned16	RW SAVE
10053	0x4745	**Ren-c1topveldiv**	Encoder 1 TOP input velocity divider				Unsig-ned16	RW SAVE
10074	0x475A	**Renc1code**	Encoder 1 model code				Unsig-ned16	RW SAVE

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0051	0x2033	**Rextencvel**	External encoder actual speed. **From firmware rev. 6 replaced by register 10054**				Unsig-ned16	WORD MAP READ RO
0052	0x2034	**Rextencsmp**	External encoder speed reading sample time. **From firmware rev. 6 replaced by register 10055**	ms	0	10000	Unsig-ned16	WORD SAVE MAP WRITE
0053	0x2035	**Rextencvel-mul**	External encoder speed reading multiplier. **From firmware rev. 6 replaced by register 10056**		0	32767	Unsig-ned16	WORD SAVE MAP WRITE
0054	0x2036	**Rextencvel-div**	External encoder speed reading divider. **From firmware rev. 6 replaced by register 10057**		1	32767	Unsig-ned16	WORD SAVE MAP WRITE
0089	0x2059	**Rextenco-topvel**	Speed detected from TOP input of external encoder. Sample time is set in the register Rextencsmp (52). **From firmware rev. 6 replaced by register 10060**				Unsig-ned16	WORD
0090	0x205A	**Rexten-ctopvelmul**	Multiplier of the speed read by external encoder TOP input. **From firmware rev. 6 replaced by register 10062**				Unsig-ned16	WORD
0091	0x205B	**Rexten-ctopveldiv**	Divider of the speed read by external encoder TOP input. **From firmware rev. 6 replaced by register 10063**				Unsig-ned16	WORD
0092-93	0x205C	**Rexten-ctopcnt**	External encoder TOP pulses counter. **From firmware rev. 6 replaced by register 10058**				Unsig-ned32	WORD
0101	0x2065	**Rextencmo-de**	Auxiliary encoder operation mode 0 = Forward quadrature 1 = Back quadrature 2 = Step-dir (Steps multiplied by 1) 3 = Step-dir (Steps multiplied by 2) **From firmware rev. 6 replaced by register 10032**		0	3	Unsig-ned16	WORD SAVE NO MAP
0153-154	0x2099	**Rextenc**	External encoder quota register. **From firmware rev. 6 replaced by register 10017**	Enc. Pulses			Signed32	LSWORD – MSWORD MAP READ
0164	0x20A4	**Rextencper**	Period read by the pulses of the external encoder. **From firmware rev. 6 replaced by register 10041**				Unsig-ned16	WORD
0165	0x20A5	**Rnewencper**	Signal of reading of new encoder pulses period B0= Motor encoder new period B1= External encoder new period. **From firmware rev. 6 replaced by register 10043**				Unsig-ned16	WORD
0268	0x210C	**Rextencpuls**	External encoder pulses per revolution. **From firmware rev. 6 replaced by register 10026**	Enc. Pulses	0	32767	Unsig-ned16	WORD SAVE
10017-18	0x4721	**Renc2pos**	Encoder 2 quota	Imp. Enc.			Signed32	DWORD RW MAP
10026	0x472A	**Renc2pulse**	Encoder 2 pulses/turn	Imp.	Default 512		Unsig-ned16	RW SAVE
10041	0x4739	**Renc2period**	Period read by encoder 2 pulses				Unsig-ned16	RO
10054	0x4746	**Renc2vel**	Encoder 2 actual velocity				Signed16	RO
10055	0x4747	**Renc2sam-ple**	Encoder 2 sample time	ms			Unsig-ned16	RW SAVE
10056	0x4748	**Renc2vel-mul**	Encoder 2 velocity multiplier				Unsig-ned16	RW SAVE
10057	0x4749	**Renc2veldiv**	Encoder 2 velocity divider				Unsig-ned16	RW SAVE
10058-59	0x474A	**Renc2topcnt**	Counter of pulses received on encoder 2 TOP				Unsig-ned32	DWORD RW
10060	0x474C	**Renc2topvel**	Velocity detected on encoder 2 TOP input				Unsig-ned16	RO
10061	0x474D	**Renc2top-sample**	Encoder 2 TOP input sample time	ms			Unsig-ned16	RW SAVE

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
10062	0x474E	**Ren-c2topvelmul**	Encoder 2 TOP input velocity multiplier				Unsig-ned16	RW SAVE
10063	0x474F	**Ren-c2topveldiv**	Encoder 2 TOP input velocity divider				Unsig-ned16	RW SAVE
10075	0x475B	**Renc2code**	Encoder 2 model code				Unsig-ned16	RW SAVE

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0044	0x202C	**Rssiencframelen**	SSI encoder frame length. Default value is 25. **From firmware rev. 6 replaced by register 10028**	Bit	1	32	Unsigned16	WORD SAVE
0094	0x205E	**Rssiencturnbit**	Bits of the frame reserved for encoder turns count. **From firmware rev. 6 replaced by register 10030**	Bit	0	16	Unsigned16	WORD RW SAVE
0095	0x205F	**Rssiencountsbit**	Bits of the frame reserved for encoder position inside the turn count. **From firmware rev. 6 replaced by register 10029**	Bit	0	16	Unsigned16	WORD RW SAVE
0096	0x2060	**Rssiencounts**	SSI encoder position on the turn count register N.B. Preset and complement bits are in the Rhwconfig register: BIT6 = Preset BIT7 = Complement **From firmware rev. 6 replaced by register 10023**		-32768	32767	Unsigned16	WORD MAP READ RO
0097	0x2061	**Rssiencturns**	SSI absolute encoder turns counter register N.B. Preset and complement bits are in the Rhwconfig register: BIT6 = Preset BIT7 = Complement **From firmware rev. 6 replaced by register 10024**		-32768	32767	Unsigned16	WORD MAP READ RO
10021-22	0x4725	**Rencabsquote**	Absolute encoder quota	Enc. Pulses			Signed32	DWORD RW MAP
10023	0x4727	**Rencabspos**	Absolute encoder postion on the turn quota	Enc. Pulses			Unsigned16	WORD RW MAP
10024	0x4728	**Rencabsturns**	Absolute encoder number of turns	Turns			Signed16	WORD RW MAP
10028	0x472C	**Rencabsframebit**	Absolute encoder frame length	Bit	Default 25		Unsigned16	RW SAVE
10029	0x472D	**Rencabsposbit**	Pulses on the turn counter frame length	Bit	Default 12		Unsigned16	RW SAVE
10030	0x472E	**Rencabsturnsbit**	Turns counter frame length	Bit	Default 13		Unsigned16	RW SAVE
10031	0x472F	**Renc1mode**	Encoder mode 0= Quadrature forward 1= Quadrature back 2= Step-dir * 1 3= Step-dir * 2				Unsigned16	RW SAVE
10032	0x4730	**Renc2mode**	Encoder mode 0= Quadrature forward 1= Quadrature back 2= Step-dir * 1 3= Step-dir * 2				Unsigned16	RW SAVE

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
10034	0x4732	**Rencabsmo-de**	**ABSOLUTE ENCODER MODE** **Bit 0:** 0 = Normal / 1= Complement **Bit 1:** 0 = Gray / 1= Binary **Bit 3-2** 00: Data right alignment 01: Data left alignment 10: Centered data 11: Free **Bit 4:** Enable absolute encoder read **Bit 5:** Enable encoder presence inital check **Bit 6:** Enable phasing without motor movement **Bit 7:** Enable Preset hardware **Bit 8:** Complement hardware	Bit			Unsig-ned16	RW SAVE
10035	0x4733	**Rencabsctrl**	Absolute encoder Controlword 1= Reset valid absolute encoder offset in Eeprom 129= Absolute encoder preset 130= Reset absolute encoder position 0 offset in Eeprom				Unsig-ned16	RW
10037	0x4735	**Rencabsbi-trate**	Absolute encoder reading clock frequency (Bit rate = 25MHz / Value of the register)		4	128	Unsig-ned16	RW SAVE
10038	0x4736	**Rencabsfra-mepause**	Pause between two absolute encoder frames readings	uS	0	20000	Unsig-ned16	RW SAVE
10039	0x4737	**Rencabspre-bit**	Number of bits of absolute encoder frame header (to be ignored)				Unsig-ned16	RW SAVE
10077	0x475D	**Rencabsco-de**	Absolute encoder model code				Unsig-ned16	RW SAVE
10078	0x475E	**Rencabspul-se**	Absolute encoder pulses/turn				Unsig-ned16	RW SAVE

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0194	0x20C2	**Rloadsav**	Depending on the value written in this register, the registers block or the variables block will be loaded from or saved into the EEPROM memory The register is reset as the command has been executed. 01 (01h) = Load registers from the Eeprom 02 (02h) = Load variables from the Eeprom 03 (03h) = Load tasks from the Eeprom 04 (04h) = Load current sensors offset from the Eeprom 05 (05h) = Load MAC address from the Eeprom 06 (06h) = Load absolute encoder offset from the Eeprom 07 (07h) = Load current signal amplification from the Eeprom 08 (08h) = Load the trigger levels for digital inputs 09 (09h) = Load the register Rpwmmode 10 (0Ah) = Read the current ANTAIOS communication parameters 11 (0Bh) = Read PROFINET “Device Name” from ANTAIOS 12 (0Ch) = Read Rmodelanatios from expansion card 129 (81h) = Save registers in the Eeprom 130 (82h) = Save variables in the Eeprom 131 (83h) = Save tasks in the Eeprom *132 (84h) = Reset the registers block at the default parameters *133 (85h) = Reset the variables block at 0 *134 (86h) = Reset the tasks *135 (87h) =Initialize the EEPROM **137 (89h) = Save current sensors offset **138 (8Ah) = Aquire and save current sensors offset 139 (8Bh) = Apply Ethernet network parameters 141 (8Dh) = Save absolute encoder offset in the Eeprom 142-143 (8Eh-8Fh) = Restart the drive. The commands must be written in sequence consecutively. 144 (90h) = Save current signal amplification in the Eeprom. 145 (91h) = Save the trigger levels for digital inputs 146 (92h) = Save the register Rpwmmode (pwm current motor generation mode). 147-148 (93h-94h) = Put Antaios chip in Boot Mode. The sequence 147-148-148 must be written to enable the boot mode. The modbus communication is disabled. Switch off to restore. 149 (95h) = Forces IP Address setting on Antaios with PROFINET protocol 150 (96h) = Sends PROFINET Device name to Antaios * Not executed if the user program is in RUN. ** Executed only if the drive is not enabled, also if the user program is in RUN. To save the il MAC address, write the registers Rethmacaddr054, Rethmacaddr032 and Rethmacaddr010 in this order, than run the command (8Ch) in Rloadsav.		0	65535	Unsigned16	WORD NO MAP
0195	0x20C3	**Rmemvar**	By writing in this register, it is possible to load or save single variables in the EEPROM memory. Bit 15 = 1 Load / 0= Save Bit 7-0 =Number of the variable to be loaded or saved (1..128)				Unsigned16	WORD NO MAP

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0196	0x20C4	**Reepsts**	Eeprom status 0 = OK B0 = R/W in progress B1 = Not valid command B2 = Error in reading registers B3 = Error in writing registers B4 = Error in reading variables B5 = Error in writing variables B6 = Error in reading program B7 = Error in writing program B8 = Error in reading password B9 = Error in writing password B10 = Error in reading task B11 = Error in writing task B12 = Error in reading alarms buffer B13 = Error in writing alarms buffer B14 = Error in reading current sensors offset B15 = Error in writing current sensors offset				Unsig-ned16	WORD NO MAP RO
0201	0x20C9	**Rindex**	MIL program variables indexing register		1	128	Holding Register (16bit) Unsig-ned16	WORD NO MAP
0329-30	0x2149	**Reepsts32**	32bit EEPROM status (From SMD5106 on) The low 16 bit are equal to the register Reepsts (196).				Unsig-ned32	LS – MSWORD MAP READ

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0216	0x20D8	**Rstrtmode**	Operation mode at power-on 0 = Load the registers 1 = Load the registers and set the operation mode 2 = Load the registers, set the operation mode and enable the drive 3 = Load the registers, set the operation mode, enable the drive and RUN the program		1	3	Unsig-ned16	WORD SAVE NO MAP
0217	0x20D9	**Rstrtconf**	Configuration at power-on When a register saving command is given, if bit 15 of Rconfig =0, the value of Rconfig is copied into this register.		0	13	Unsig-ned16	WORD SAVE NO MAP
0389-90	0x2185	**Rstrtvel**	Value of Rvel set at the power-on	rev*100 /s	-10000	10000	Signed32	LS-MSWORD SAVE NO MAP
0391-92	0x2187	**Rstrtvss**	Value of Rvss set at the power-on	rev*100 /s	0	10000	Unsig-ned32	LS-MSWORD SAVE NO MAP
0393-94	0x2189	**Rstrtacc**	Value of Racc set at the power-on	rev*10 /s²	1	200000	Unsig-ned32	LS-MSWORD SAVE NO MAP
0395-96	0x218B	**Rstrtdec**	Value of Rdec set at the power-on	rev*10 /s²	1	200000	Unsig-ned32	LS-MSWORD SAVE NO MAP
0397-98	0x218D	**Rstrtpostarg**	Value of Rpostarg set at the power-on	Step	0x80000000	0x7FFFFFFF	Signed32	LS-MSWORD SAVE NO MAP
0399-400	0x218F	**Rstrthvh**	Value of Rhvh set at the power-on	rev*100 /s	1	10000	Signed32	LS-MSWORD SAVE NO MAP
0401-02	0x2191	**Rstrthvl**	Value of Rhvl set at the power-on	rev*100 /s	1	10000	Signed32	LS-MSWORD SAVE NO MAP
0403-04	0x2193	**Rstrthacc**	Value of Rhacc set at the power-on	rev*10 /s²	1	200000	Unsig-ned32	LS-MSWORD SAVE NO MAP

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0405	0x2195	**Rstrthmode**	Value of Rhmode set at the power-on -16 = Homing on FLS + motor encoder TOP, positive direction -15 = Homing on FLS + motor encoder TOP, negative direction -14 = Homing on FLS, positive direction -13 = Homing on FLS, negative direction -12 = Homing with forward mechanical limit + encoder TOP (only SmartMode and Closed Loop) -11 = Homing with backward mechanical limit + encoder TOP (only SmartMode and Closed Loop) -10 = Homing with forward mechanical limit (only SmartMode and Closed Loop) -9 = Homing with backward mechanical limit (only SmartMode and Closed Loop) -8 = Homing with forward mechanical limit + axis measuring (Resets the registers Rlowlim and Rupplim) (only SmartMode and Closed Loop) -7 = Homing with backward mechanical limit + axis measuring (Resets the registers Rlowlim and Rupplim) (only SmartMode and Closed Loop) -6 = Homing only with TOP in positive direction -5 = Homing only with TOP in negative direction -4 = Homing with BLS + TOP rising edge, positive direction -3 = Homing with BLS + TOP rising edge, negative direction -2 = Homing only with BLS in positive direction -1 = Homing only with BLS in negative direction 0 = Homing on place		-16	0	Signed16	WORD SAVE NO MAP

If the drive is powered and the rotary switches are set to 00, the communication parameters of the fieldbuses are set to AEC default values.

In detail:

Modbus RS232/RS485:
Address: 125
BaudRate: 9600
Parity: Even
StopBit 1
Modalita': Intel

CanOpen:
Address: 125
BaudRate: 125Kbit

Profibus:
Address: 125
BaudRate: Auto

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0099	0x2063	**Rcu-ronramptime**	Cuttent ramp time at current on	ms	0	65535	Unsig-ned16	WORD SAVE NO MAP
0209	0x20D1	**Rcuract**	Actual current	mA			Signed16	WORD MAP READ RO
0210	0x20D2	**Rcurnom**	Nominal current supplied to the motor	mA	0	8500	Unsig-ned16	WORD SAVE MAP WRITE
0211	0x20D3	**Rcurred**	Reduced current supplied to the motor	mA	0	8500	Unsig-ned16	WORD SAVE MAP WRITE
0212	0x20D4	**Rcurboost**	Boost current during ramps	mA	0	10000	Unsig-ned16	WORD SAVE MAP WRITE
0213	0x20D5	**Rcurtorque**	Requested current in TORQUE mode	mA	-10000	10000	Signed16	WORD MAP WRITE ANA
0214	0x20D6	**Rtboost**	Maximum boost time	ms	0	5000	Unsig-ned16	WORD SAVE NO MAP
0215	0x20D7	**Rtcred**	Time frame before switching to reduced current	ms	0	10000	Unsig-ned16	WORD SAVE NO MAP
0219	0x20DB	**Rfocmode**	Actual configuration of the drive Actual status of the control 0= Not initialized 1 = Open loop 2 = Closed loop 3 = Smart mode (smart closed loop)		0	3	Unsig-ned16	WORD NO MAP RO
0220	0x20DC	**Rcurmode-act**	Actual current level 0 = No current 1 = Reduced current 2 = Nominal current 3 = Current boost 4 = Automatic current reduction		0	4	Unsig-ned16	WORD RO MAP READ
0221	0x20DD	**Rcurmode**	Current control modes 0 = No current 1 = Reduced current 2 = Nominal current 3 = Current boost		0	3	Unsig-ned16	WORD SAVE NO MAP
0222	0x20DE	**Rconfig**	Drive operation mode configuration 0= Not configured 1= Reserved 2= Open Loop / Speed 3= Open Loop / Position 4= Open Loop / Step-Direction 5= Closed Loop / Torque 6= Closed Loop / Speed (with encoder) 7= Closed Loop / Position 8= Closed Loop / Step-Direction 9= Closed Loop / Speed (with tachometer – Only DMD) 10= Reserved 11= Smart Loop / Speed 12= Smart Loop / Position 13= Smart Loop / Step-Direction 14= Reserved 15= Open loop PWM (Only DMD)		0	15	Unsig-ned16	WORD NO MAP

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0223	0x20DF	**Renmask**	Drive enabling control mask Bit 0: Rconfig is set Bit 1: Bit 2: Bit 3: Bit 4: Bit 15: Control mask error				Unsig-ned16	WORD NO MAP RO
0264-265	0x2108	**Rtrqdisp**	Display the torque				Unsig-ned32	LS – MSWORD MAP READ RO
0270	0x210E	**Rstpres**	Motor step resolution		1	1024	Unsig-ned16	WORD SAVE NO MAP
0279	0x2117	**Rkpiq**	Kp PI current Iq		0	32767	Unsig-ned16	WORD SAVE NO MAP
0280	0x2118	**Rkiiq**	Ki PI current Iq		0	32767	Unsig-ned16	WORD SAVE NO MAP
0281	0x2119	**Rkpid**	Kp PI current Id		0	32767	Unsig-ned16	WORD SAVE NO MAP
0282	0x211A	**Rkiid**	Ki PI current Id		0	32767	Unsig-ned16	WORD SAVE NO MAP
0283	0x211B	**Rkpvel**	Kp PI velocity in FOC Close		0	32767	Unsig-ned16	WORD SAVE NO MAP
0284	0x211C	**Rkivel**	Ki PI velocity in FOC Close		0	32767	Unsig-ned16	WORD SAVE NO MAP
0285	0x211D	**Rkcvel**	Kc PI velocity in FOC Close		0	32767	Unsig-ned16	WORD SAVE NO MAP
0286	0x211E	**Rkppos**	Kp PI position in FOC Close		0	32767	Unsig-ned16	WORD SAVE NO MAP
0287	0x211F	**Rkipos**	Ki PI position in FOC Close		0	32767	Unsig-ned16	WORD SAVE NO MAP
0288	0x2120	**Rkcipos**	Kci PI position in FOC Close		0	32767	Unsig-ned16	WORD SAVE NO MAP
0289	0x2121	**Rkffpos**	Kff PI position in FOC Close		0	32767	Unsig-ned16	WORD SAVE NO MAP
0290	0x2122	**Rkafpos**	Kaf PI position in FOC Close		0	32767	Unsig-ned16	WORD SAVE NO MAP
0291	0x2123	**Rswacfw**	Acc. Forward switch choice 0 = Current 1 = Speed		0	1	Unsig-ned16	WORD SAVE NO MAP
0292	0x2124	**Radjpicur**	Percentage of current PI correction from maximum to minimum current level. This permits to keep the PI reactive on the whole range of current without going in oscillation.		-100	100	Signed16	WORD SAVE NO_MAP
0299	0x212B	**Rstpressd**	Step resolution in step/dir mode The function of this register has been assigned to the register Rstpres from the firmware 3.44 onwards. For firmwares after 3.44, this register has no functions.		1	1024	Unsig-ned16	WORD SAVE NO MAP

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0300	0x212C	**Rkpstpdir**	Kp PI increments management in step/dir mode		0	4096	Unsig-ned16	WORD SAVE NO MAP
0301	0x212D	**Rampfrq**	Amplification of the increment step/dir in FOC close mode from 1 to 10 (1 = 1:1 / 10 = Multiplies the input steps by 10)		0	32767	Unsig-ned16	WORD SAVE NO MAP
0309	0x2135	**Rdefluxen**	Enable FOCOPEN deflux		0	1	Unsig-ned16	WORD SAVE NO MAP
0310	0x2136	**Rdefluxmin**	FOCOPEN minimum deflux current		0	5000	Unsig-ned16	WORD SAVE NO MAP
0311	0x2137	**Rphamode**	Phaase advance FOCOPEN mode		0	9	Unsig-ned16	WORD SAVE NO MAP
0312	0x2138	**Rphagain**	Phase advance FOCOPEN gain		0	8192	Unsig-ned16	WORD SAVE NO MAP
0314	0x213A	**Rveladjo-penmode**	Only for DMD. Speed correction in open loop (armature feedback)		-4096	4096	Signed16	WORD
0351	0x215F	**Radjcura**	Adjustment of signal amplification read by the current sensor of the phase A		-128	127	Signed16	WORD SAVE
0352	0x2160	**Radjcurb**	Adjustment of signal amplification read by the current sensor of the phase B		-128	127	Signed16	WORD SAVE
0368	0x2170	**Rvbusoffs**	Bus voltage reading offset		-32768	32767	Signed16	WORD SAVE
0370	0x2172	**Rla**	Current read from phase A	mA			Signed16	WORD MAP RO
0371	0x2173	**Rlb**	Current read from phase B	mA			Signed16	WORD MAP RO
0372	0x2174	**Rlan**	Filtered current read from phase A	mA			Signed16	WORD MAP RO
0373	0x2175	**Rlbn**	Filtered current read from phase B	mA			Signed16	WORD MAP RO
0374	0x2176	**Rla_offs**	Phase A calculated offset	Bit			Signed16	WORD NO MAP RO
0375	0x2177	**Rlb_offs**	Phase B calculated offset	Bit			Signed16	WORD NO MAP RO
0377	0x2179	**Rlb_offsEl**	Phase B offset manual correction (*) Saved with command 137 in the register Rloadsav. Calculated and saved with command 138 in the regi-ster Rloadsav.	Bit	-32768	32767	Signed16	WORD SAVE (*) NO MAP
0378	0x217A	**Rid**	Direct current	mA			Signed16	WORD NO MAP RO
0379	0x217B	**Riq**	Current in quadrature	mA			Signed16	WORD MAP RO
0388	0x2184	**Ridcorr**	ID correction of current in Smart Mode	mA	-5000	5000	Signed16	WORD RW SAVE
0495	0x21EF	**Rpwmmode**	Motor pwm current generation mode: 0=Centered 1=Left aligned Save the data with the command 146 in Rloadsav, shut down and power-on the drive.		0	1	Unsig-ned16	WORD

MOTOR PARAMETERS

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0218	0x20DA	**Rmottype**	Index of the configured motor. Indicates the index of the configured motor in the motors database.				Unsig-ned16	WORD SAVE NO MAP
0271	0x210F	**Rmotres**	Phase resistance of the motor	ohm *10	1	32767	Unsig-ned16	WORD SAVE NO MAP
0272	0x2110	**Rmotind**	Phase inductance of the motor	mHe *10	1	32767	Unsig-ned16	WORD SAVE NO MAP
0273	0x2111	**Rmotkfm**	F.c.e.m. constant L(mH)*Inom(mA)*100/1000	mHA *100	1	32767	Unsig-ned16	WORD SAVE NO MAP
0274	0x2112	**Rmottens-nom**	Only for DMD. Nominal voltage of the motor.	Volt	1	32767	Unsig-ned16	WORD SAVE
0275	0x2113	**Rmotiph**	Motor nominal phase current	mA	1	32767	Unsig-ned16	WORD SAVE NO MAP
0276	0x2114	**Rmotktq**	Motor torque constant	mNm /A	1	65535	Unsig-ned16	WORD SAVE NO MAP
0277	0x2115	**Rmotvelnom**	Only for DMD. Nominal speed of the motor.	rev/s			Unsig-ned16	WORD SAVE

TIMER

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0155-156	0x209B	**Rtim0**	Register decreased by 1 down to 0 every 1 mS	ms			Unsig-ned32	LSWORD – MSWORD NO MAP
0157-158	0x209D	**Rtim1**	Register decreased by 1 down to 0 every 1 mS	ms			Unsig-ned32	LSWORD – MSWORD NO MAP
0159-160	0x209F	**Rtim2**	Register decreased by 1 down to 0 every 1 mS	ms			Unsig-ned32	LSWORD – MSWORD NO MAP
0161-162	0x20A1	**Rtim3**	Register decreased by 1 down to 0 every 1 mS	ms			Unsig-ned32	LSWORD – MSWORD NO MAP
0536-37	0x2218	**Rtim4**	Register decreased by 1 down to 0 every 1 mS	ms	0		Unsig-ned32	WORD
0538-39	0x221A	**Rtim5**	Register decreased by 1 down to 0 every 1 mS	ms	0		Unsig-ned32	WORD
0540-41	0x221C	**Rtim6**	Register decreased by 1 down to 0 every 1 mS	ms	0		Unsig-ned32	WORD
0542-43	0x221E	**Rtim7**	Register decreased by 1 down to 0 every 1 mS	ms	0		Unsig-ned32	WORD

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Mini-mum value	Maximum value	Type	Note	Note
0148	0x2094	**Rtempact**	Actual temperature of the drive		°C			Signed16	WORD MAP RO
0149	0x2095	**Rtensact**	Actual voltage of the drive CC bus		Volt			Unsig-ned16	WORD MAP RO
0150	0x2096	**Rcurdcact**	Actual current requested by the drive to the DC power supply		mA			Unsig-ned16	WORD NO MAP RO
0190	0x20BE	**Rswrev**	Software revision. **From firmware rev. 6 replaced by register 10005**					Unsig-ned16	WORD NO MAP RO
0191	0x20BF	**Rhwrev**	Hardware revision. **From firmware rev. 6 replaced by register 10006**					Unsig-ned16	WORD NO MAP RO
0192-193	0x20C0	**Rserial**	Serial number of the device. **From firmware rev. 6 replaced by register 10007-8**					Unsig-ned32	LS – MSWORD SAVE NO MAP
0197	0x20C5	**Rrotsw**	Copy of the value read by the rotary switch			0	99	Unsig-ned16	WORD NO MAP RO
0343	0x2157	**Rhwoptions**	Active hardware options B1= STO present					Unsig-ned16	WORD RO

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Mini-mum value	Maximum value	Type	Note	Note
0198	0x20C6	**Rindtype**	Tipo indexer 0: SMD30.06LIM 1: SMD30.06LIC 2: SMD30.06LIP 3: SMD50.06LIM 4: SMD50.06LIC 5: SMD50.06LIP 6: SMD30.06HIM 7: SMD30.06HIC 8: SMD30.06HIP 9: SMD50.06HIM 10: SMD50.06HIC 11: SMD50.06HIP 12: SMD10.04LIM 13: SMD10.04LIC 14: SMD10.04LIP 15: SMD10.04HIM 16: SMD10.04HIC 17: SMD10.04HIP 18: SMD50.06LS 19: SMD50.06HS 20: SMD30.06LS 21: SMD30.06HS 22: SMD10.04LS 23: SMD10.04HS 24: SMD10.04LUM 25: SMD10.04HUM 26: SMD104u 27: SMD30.06LIE 28: SMD50.06LIE 29: SMD30.06HIE 30: SMD50.06HIE 31: SMD10.04LIE 32: SMD10.04HIE 33: SMD11.04LIM 34: SMD11.04LIC 35: SMD11.04LIP 36: SMD11.04LS 37: SMD11.04LUM 38: SMD11.04LIE 39: SMD11.04HIM 40: SMD11.04HIC 41: SMD11.04HIP 42: SMD11.04HS 43: SMD11.04HUM 44: SMD11.04HIE 45: SMD114u 46: SMD51.06LIM 47: SMD51.06LIC 48: SMD51.06LIP 49: SMD51.06HIM 50: SMD51.06HIC 51: SMD51.06HIP 52: SMD51.06LS 53: SMD51.06HS 54: SMD51.06LIE 55: SMD51.06HIE 56: SMD51.06LUM 57: SMD51.06HUM 58: SMD51.06LIT 59: SMD51.06HIT 60: SMD11.04LIT 61: SMD11.04HIT 62: SMD31.06LIM 63: SMD31.06LIC 64: SMD31.06LIP 65: SMD31.06HIM 66: SMD31.06HIC 67: SMD31.06HIP 68: SMD31.06LS 69: SMD31.06HS 70: SMD31.06LIE 71: SMD31.06HIE 72: SMD31.06LUM 73: SMD31.06HUM 74: SMD31.06LIT	75: SMD31.06HIT 76:SMD12.04LIM 77: SMD12.04LIC 78: SMD12.04LIP 79: SMD12.04LS 80: SMD12.04LUM 81: SMD12.04LIE 82: SMD12.04HIM 83: SMD12.04HIC 84: SMD12.04HIP 85: SMD12.04HS 86: SMD12.04HUM 87: SMD12.04HIE 88: SMD12.04LIT 89: SMD12.04HIT 90: SMD22.04LIM 91: SMD22.04LIC 92: SMD22.04LIP 93: SMD22.04HIM 94: SMD22.04HIC 95: SMD22.04HIP 96: SMD22.04LS 97: SMD22.04HS 98: SMD22.04LIE 99: SMD22.04HIE 100: SMD22.04LUM 101: SMD22.04HUM 102: SMD22.04LIT 103: SMD22.04HIT 104: SMD1204HIN 105: SMD1204LIN 106: SMD5106HIN 107: SMD5106LIN 108: SMD2204HIN 109: SMD2204LIN **From firmware rev. 6, replcaed be registers higher than 10000**				Holding Register (16bit) Unsig-ned16 HI-BYTE code 0: SMD 1: DMD 2: BMD b7 LO-BYTE code: 0: I (ex SMD506I) 1: A (ex SMD506S)	WORD NO MAP RO

Indirizzo Modbus / Profibus	Indirizzo CAN EtherCAT Profinet	Nome Registro	Descrizione	U.M.	Valore Minimo	Valore Massimo	Tipo	Note
10000	0x4710	**Rindfamily**	Drive series. (From fw 5.50 the register 198 is read at 0xFFFF) Example: -1204 -2204 -5206				Unsign16	WORD RO
10001	0x4711	**Rindmodel**	**MODEL DEFINITION** **Voltage Bit 2-1-0:** 000= LOW 001= HIGH **Application Bit 5-4-3:** 000= Indexer 001= Step/dir 010= Labeller **Drive model Bit 8-7-6** 000= Undefined 001= SMD 010= DMD 011= BMD **Communication chip bit 11-10-9** 000= DSP 001= Wiznet 010= ANTAIOS 011= VPC3				Unsign16	WORD RO
10002	0x4712	**Rindfielsbus**	Fieldbus Description 0= USB (xUM) 1= Modbus RTU (xxM) 2= Modbus TCP (xxE) 3= Modbus TCP ANTAIOS (xxEA) 4= Profibus (xxP) 5= Profibus ANATIOS (xxPA) 6= EtherCAT (xxT) 7= PROFINET (xxN) 8= Canopen (xxC) 9= Canopen NATAIOS (xxCA) 10= Step-dir				Unsign16	WORD RO
10003	0x4713	**Rindoptions**	**OPTIONS DEFINITION** **Absolute encoder Bit 2-1-0:** 000= None 001= SSI 010= Endat 011= BISS 100= sin/cos 101= free 110= Free 11= free				Unsign16	WORD RO
10004	0x4714	**Rindregset**	Registers mapping 0= Revision 1"				Unsign16	WORD RO
10005	0x4715	**Rindswrev**	Software revision				Unsign16	WORD RO
10006	0x4716	**Rindhwrev**	Hardware revision				Unsign16	WORD RO
10007-8	0x4717	**Rindsn**	Serial number (copy of the register 192-193)				Unsign32	WORD RO

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0042-43	0x202A	**Rtestintmot**	Register for the control of the communication mode (32bit Intel or Motorola). The fixed value of this register is 1234567890 (0x499602D2) By executing a read of this register at 32bit, it is possible to know if the communication is configured in Inter or Motorola mode, depending ont he result.		1234567890 0x499602D2	1234567890 0x499602D2	Holding Register (32bit)	WORD RO
0166	0x20A6	**Rprofists**	Profibus communication status 0 = Disabled 1 = Parameterization 2 = Configuration 3 = Data Exchange FF = Fatal Error				Unsig-ned16	WORD NO MAP RO
0167	0x20A7	**Rprofibaud**	Profibus baudrate 0: Auto baud-rate 1: 9.6KB 2: 19.2KB 3: 31.25KB 4: 45.45KB 5: 93.75KB 6: 187.5KB 7: 500KB 8: 1500KB 9: 3000KB 10: 6000KB 11: 12000KB		0	11	Unsig-ned16	WORD SAVE NO MAP
0168	0x20A8	**Rprofiaddr**	Profibus Address This value is added to the hardware address.		0	127	Unsig-ned16	WORD SAVE NO MAP
0169	0x20A9	**Rcanbaud**	Can Baud Rate 0 = 10Kb 1 = 20Kb 2 = 50Kb 3 = 125Kb 4 = 250Kb 5 = 500Kb 6 = 800Kb 7 = 1Mb		0	8	Unsig-ned16	WORD SAVE NO MAP
0170	0x20AA	**Rcanaddr**	Can Address This value is added to the hardware address.		0	127	Unsig-ned16	WORD SAVE NO MAP
0171-172	0x20AB	**Rcantx**	Number of messages transmitted in Can				Unsig-ned32	LS – MSWORD NO MAP RO
0173-174	0x20AD	**Rcanrx**	Number of messages received in Can				Unsig-ned32	LS – MSWORD NO MAP RO
0175	0x20AF	**Rcanovr**	Number of overrun errors in Can				Unsig-ned16	WORD NO MAP RO
0176	0x20B0	**Rcanerr**	Number of errors in Can				Unsig-ned16	WORD NO MAP RO
0177	0x20B1	**Rcanidx**	Index of the CANopen object to be read or written		0	65535	Unsig-ned16	WORD NO MAP
0178	0x20B2	**Rcansub**	Sub-index of the CANopen object to be read or written		0	255	Unsig-ned16	WORD NO MAP
0179-180	0x20B3	**Rcannew**	New value to be written in CanOpen object				Unsig-ned32	LS – MSWORD NO MAP

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0181	0x20B5	**Rcancmd**	CANopen object Read/Write command 1 = Write value 2 = Read value		0	2	Unsig-ned16	WORD NO MAP
0182-183	0x20B6	**Rcanact**	Read value from CANopen object				Unsig-ned32	LS – MSWORD NO MAP RO
0184	0x20B8	**Rcansts**	CanOpen status Bit 0..3 status LED RED b3 b2 b1 b0 0 0 0 0 Disabled 0 0 1 1 Warning Limit reached 0 1 0 0 Error control event 0 1 0 1 Sync Error 0 1 1 0 Event timer error 0 1 1 1 Bus OFF Bit 4..7 status LED GREEN b7 b6 b5 b4 0 0 0 1 Pre-Operational 0 0 1 0 Stopped 0 1 0 1 Oper				Unsig-ned16	WORD NO MAP RO
0185	0x20B9	**Rserbaud**	Serial port Baud Rate 0 = 1200 1 = 2400 2 = 4800 3 = 9600 4 = 19200 5 = 38400 6 = 57600 7 = 115200		0	7	Unsig-ned16	WORD SAVE NO MAP
0186	0x20BA	**Rserpar**	Serial port parameter (par, stop bit) 0 = NONE, 1 1 = EVEN, 1 2 = ODD, 1 3 = NONE, 2 4 = EVEN, 2 5 = ODD, 2		0	5	Unsig-ned16	WORD SAVE NO MAP
0187	0x20BB	**Rserdly**	Serial Reply delay	ms	0	16	Unsig-ned16	WORD SAVE NO MAP
0188	0x20BC	**Rseraddr**	Serial port address (this value is added to the hardware address)		0	31	Unsig-ned16	WORD SAVE NO MAP
0189	0x20BD	**Rintmot**	Intel/Motorola mode selection for 32bit registers RS232/RS485 serial ports		0	1	Unsig-ned16	WORD SAVE NO MAP
0302	0x212E	**Rfinsenable**	Enable FINS/TCP protocol in drives with Ethernet communication		0	1	Unsig-ned16	WORD SAVE NO MAP
0303	0x212F	**Rfinsnode**	FINS/TCP station number		1	254	Unsig-ned16	WORD SAVE NO MAP
0304	0x2130	**Rmdbport**	TCP port number for Modbus/TCP protocol (Default = 502)		0	65535	Unsig-ned16	WORD SAVE NO MAP
0305	0x2131	**Rethintmot**	Definition of Intel or Motorola mode for registers at 32bit in Modbus/TCP communication or Ethernet based protocols 0= Intel 1=Motorola		0	1	Unsig-ned16	WORD SAVE NO MAP

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0308	0x2134	**Rextmdbti-me**	Modbus RTU inter-message additional time, expres-sed in ms. It's used in case of slow remote commuincation. Default value is 0. If a value <>0 is set, the communication speed slows down.	ms	0	10000	Unsig-ned16	WORD SAVE NO MAP
0353	0x2161	**Rstatuscan**	Controlword-Statusword congruence errors in CanO-pen				Unsig-ned16	WORD NO MAP
0354	0X2162	**Rcanmodeo-foperation**	Copy of the object 0x6060				Unsig-ned16	WORD NO MAP
0355	0x2163	**Rswrevan-taios**	Antaios firmware version **From firmware rev. 6 replaced by register 10140**				Unsig-ned16	WORD NO MAP
0356	0x2164	**Rhwrevan-taios**	Antaios hardware firmware **From firmware rev. 6 replaced by register 10141**				Unsig-ned16	WORD NO MAP
0357	0x2165	**Rethercatid**	Explicit address ID Ethercat				Unsig-ned16	WORD NO MAP
0358	0x2166	**Rstrtmode-sofoperation**	Setting of the object 0x6060 at power-on, for CANo-pen and EtherCAT protocols		-1	8	Unsig-ned16	WORD SAVE
0365	0x216D	**Rds402com-patibility**	Bit used to adapt the DS402 stack to the Master Bit 0: 1= Current off with Controlword xxx7 (OMRON motion blocks) Bit 1: 1= CSP and CSV active with Controlword xxxF (OMRON motion blocks)					
0515	0x2203	**Rethlocipad-dr32**	Byte 3 - Byte 2 Ethernet local IP address		0	65535	Unsig-ned16	WORD SAVE
0516	0x2204	**Rethlocipad-dr10**	Byte 1 - Byte 0 Ethernet local IP address		0	65535	Unsig-ned16	WORD SAVE
0517	0x2205	**Rethsub-net32**	Byte 3 - Byte 2 Subnet ethernet		0	65535	Unsig-ned16	WORD SAVE
0518	0x2206	**Rethsub-net10**	Byte 1 - Byte 0 Subnet ethernet		0	65535	Unsig-ned16	WORD SAVE
0519	0x2207	**Rethgwad-dr32**	Byte 3 - Byte 2 Gateway address ethernet		0	65535	Unsig-ned16	WORD SAVE
0520	0x2208	**Rethgwad-dr10**	Byte 1 – Byte 0 Gateway address ethernet		0	65535	Unsig-ned16	WORD SAVE
0521	0x2209	**Rethmacad-dr054**	Byte 5 - Byte 4 MAC address 0 ethernet		0	65535	Unsig-ned16	WORD EEPROM SAVE
0522	0x220A	**Rethmacad-dr032**	Byte 3 - Byte 2 MAC address 0 ethernet		0	65535	Unsig-ned16	WORD EEPROM SAVE
0523	0x220B	**Rethmacad-dr010**	Byte 1 - Byte 0 MAC address 0 ethernet		0	65535	Unsig-ned16	WORD EEPROM SAVE
0524	0x220C	**Rethmacad-dr154**	Byte 5 - Byte 4 MAC address 1 ethernet (Profinet port 1)		0	65535	Unsig-ned16	WORD
0525	0x220D	**Rethmacad-dr132**	Byte 3 - Byte 2 MAC address 1 ethernet (Profinet port 1)		0	65535	Unsig-ned16	WORD
0526	0x220E	**Rethmacad-dr110**	Byte 1 - Byte 0 MAC address 1 ethernet (Profinet port 1)		0	65535	Unsig-ned16	WORD
0527	0x220F	**Rethmacad-dr254**	Byte 5 - Byte 4 MAC address 2 ethernet (Profinet port 2)		0	65535	Unsig-ned16	WORD
0528	0x2210	**Rethmacad-dr232**	Byte 3 - Byte 2 MAC address 2 ethernet (Profinet port 2)		0	65535	Unsig-ned16	WORD
0529	0x2211	**Rethmacad-dr210**	Byte 1 - Byte 0 MAC address 2 ethernet (Profinet port 2)		0	65535	Unsig-ned16	WORD
0544	0x2220	**Rethprot-sock1**	Protocol for socket 1 0=TCP / 1=UDP		0	1	Unsig-ned16	WORD SAVE
0545	0x2221	**Rethprot-sock2**	Protocol for socket 2 0=TCP / 1=UDP		0	1	Unsig-ned16	WORD SAVE
0546	0x2222	**Rethprot-sock3**	Protocol for socket 3 0=TCP / 1=UDP		0	1	Unsig-ned16	WORD SAVE

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0547	0x2223	**Rethprot-sock4**	Reserved		0	1	Unsig-ned16	WORD SAVE
0548	0x2224	**Rethprot-sock5**	Reserved		0	1	Unsig-ned16	WORD SAVE
0549	0x2225	**Rethprot-sock6**	Reserved		0	1	Unsig-ned16	WORD SAVE
0550	0x2226	**Rethprot-sock7**	Reserved		0	1	Unsig-ned16	WORD SAVE
0551	0x2227	**Rethport-sock1**	Socket port 1		0	65535	Unsig-ned16	WORD SAVE
0552	0x2228	**Rethport-sock2**	Socket port 2		0	65535	Unsig-ned16	WORD SAVE
0553	0x2229	**Rethport-sock3**	Socket port 3		0	65535	Unsig-ned16	WORD SAVE
0554	0x222A	**Rethport-sock4**	Socket port 4 (only for SMD2204)		0	65535	Unsig-ned16	WORD SAVE
0555	0x222B	**Rethport-sock5**	Socket port 5 (only for SMD2204)		0	65535	Unsig-ned16	WORD SAVE
0556	0x222C	**Rethport-sock6**	Socket port 6 (only for SMD2204)		0	65535	Unsig-ned16	WORD SAVE
0557	0x222D	**Rethport-sock7**	Socket port 7 (only for SMD2204)		0	65535	Unsig-ned16	WORD SAVE
0558	0x222E	**Rethtcpti-meout**	TCP socket timeout time in absence of activities. With value=0 the timeout is disabled.	s	0	65535	Unsig-ned16	WORD SAVE
10080	0x4760	**Rprofinetna-meptr**	Pointer to the string “PROFINET device name”				Unsig-ned16	RW
10081	0x4761	**Rprofinetna-mechr**	Pointed character of the string “PROFINET device name”				Unsig-ned16	RW
10082	0x4762	**Rprofinetna-me**	PROFINET string name (not sendable directly to the drive)				STRING	
10140	0x479C	**Rswrevan-taios**	Antaios firmware revision				Unsig-ned16	RW SAVE
10141	0x479D	**Rhwrevan-taios**	Antaios hardware revision				Unsig-ned16	RW SAVE
10142	0x479E	**Rmodelan-taios**	Expansion card model on Antaios FW				Unsig-ned16	RW SAVE

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
8200		**Rfieldbusdatarx1**	Word 1 Data received via fieldbus				Unsigned16	WORD
8201		**Rfieldbusdatarx2**	Word 2 Data received via fieldbus				Unsigned16	WORD
8202		**Rfieldbusdatarx3**	Word 3 Data received via fieldbus				Unsigned16	WORD
8203		**Rfieldbusdatarx4**	Word 4 Data received via fieldbus				Unsigned16	WORD
8204		**Rfieldbusdatarx5**	Word 5 Data received via fieldbus				Unsigned16	WORD
8205		**Rfieldbusdatarx6**	Word 6 Data received via fieldbus				Unsigned16	WORD
8206		**Rfieldbusdatarx7**	Word 7 Data received via fieldbus				Unsigned16	WORD
8207		**Rfieldbusdatarx8**	Word 8 Data received via fieldbus				Unsigned16	WORD
8208		**Rfieldbusdatarx9**	Word 9 Data received via fieldbus				Unsigned16	WORD
8209		**Rfieldbusdatarx10**	Word 10 Data received via fieldbus				Unsigned16	WORD
8210		**Rfieldbusdatarx11**	Word 11 Data received via fieldbus				Unsigned16	WORD
8211		**Rfieldbusdatarx12**	Word 12 Data received via fieldbus				Unsigned16	WORD
8212		**Rfieldbusdatarx13**	Word 13 Data received via fieldbus				Unsigned16	WORD
8213		**Rfieldbusdatarx14**	Word 14 Data received via fieldbus				Unsigned16	WORD
8214		**Rfieldbusdatarx15**	Word 15 Data received via fieldbus				Unsigned16	WORD
8215		**Rfieldbusdatarx16**	Word 16 Data received via fieldbus				Unsigned16	WORD
8216		**Rfieldbusdatarx17**	Word 17 Data received via fieldbus				Unsigned16	WORD
8217		**Rfieldbusdatarx18**	Word 18 Data received via fieldbus				Unsigned16	WORD
8218		**Rfieldbusdatarx19**	Word 19 Data received via fieldbus				Unsigned16	WORD
8219		**Rfieldbusdatarx20**	Word 20 Data received via fieldbus				Unsigned16	WORD
8220		**Rfieldbusdatarx21**	Word 21 Data received via fieldbus				Unsigned16	WORD
8221		**Rfieldbusdatarx22**	Word 22 Data received via fieldbus				Unsigned16	WORD
8222		**Rfieldbusdatarx23**	Word 23 Data received via fieldbus				Unsigned16	WORD
8223		**Rfieldbusdatarx24**	Word 24 Data received via fieldbus				Unsigned16	WORD
8224		**Rfieldbusdatarx25**	Word 25 Data received via fieldbus				Unsigned16	WORD
8225		**Rfieldbusdatarx26**	Word 26 Data received via fieldbus				Unsigned16	WORD
8226		**Rfieldbusdatarx27**	Word 27 Data received via fieldbus				Unsigned16	WORD
8227		**Rfieldbusdatarx28**	Word 28 Data received via fieldbus				Unsigned16	WORD
8228		**Rfieldbusdatarx29**	Word 29 Data received via fieldbus				Unsigned16	WORD

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
8229		**Rfieldbusdatarx30**	Word 30 Data received via fieldbus				Unsigned16	WORD
8230		**Rfieldbusdatarx31**	Word 31 Data received via fieldbus				Unsigned16	WORD
8231		**Rfieldbusdatarx32**	Word 32 Data received via fieldbus				Unsigned16	WORD
8232		**Rfieldbusdatatx1**	Word 1 Data sent to fieldbus				Unsigned16	WORD
8233		**Rfieldbusdatatx2**	Word 2 Data sent to fieldbus				Unsigned16	WORD
8234		**Rfieldbusdatatx3**	Word 3 Data sent to fieldbus				Unsigned16	WORD
8235		**Rfieldbusdatatx4**	Word 4 Data sent to fieldbus				Unsigned16	WORD
8236		**Rfieldbusdatatx5**	Word 5 Data sent to fieldbus				Unsigned16	WORD
8237		**Rfieldbusdatatx6**	Word 6 Data sent to fieldbus				Unsigned16	WORD
8238		**Rfieldbusdatatx7**	Word 7 Data sent to fieldbus				Unsigned16	WORD
8239		**Rfieldbusdatatx8**	Word 8 Data sent to fieldbus				Unsigned16	WORD
8240		**Rfieldbusdatatx9**	Word 9 Data sent to fieldbus				Unsigned16	WORD
8241		**Rfieldbusdatatx10**	Word 10 Data sent to fieldbus				Unsigned16	WORD
8242		**Rfieldbusdatatx11**	Word 11 Data sent to fieldbus				Unsigned16	WORD
8243		**Rfieldbusdatatx12**	Word 12 Data sent to fieldbus				Unsigned16	WORD
8244		**Rfieldbusdatatx13**	Word 13 Data sent to fieldbus				Unsigned16	WORD
8245		**Rfieldbusdatatx14**	Word 14 Data sent to fieldbus				Unsigned16	WORD
8246		**Rfieldbusdatatx15**	Word 15 Data sent to fieldbus				Unsigned16	WORD
8247		**Rfieldbusdatatx16**	Word 16 Data sent to fieldbus				Unsigned16	WORD
8248		**Rfieldbusdatatx17**	Word 17 Data sent to fieldbus				Unsigned16	WORD
8249		**Rfieldbusdatatx18**	Word 18 Data sent to fieldbus				Unsigned16	WORD
8250		**Rfieldbusdatatx19**	Word 19 Data sent to fieldbus				Unsigned16	WORD
8251		**Rfieldbusdatatx20**	Word 20 Data sent to fieldbus				Unsigned16	WORD
8252		**Rfieldbusdatatx21**	Word 21 Data sent to fieldbus				Unsigned16	WORD
8253		**Rfieldbusdatatx22**	Word 22 Data sent to fieldbus				Unsigned16	WORD
8254		**Rfieldbusdatatx23**	Word 23 Data sent to fieldbus				Unsigned16	WORD
8255		**Rfieldbusdatatx24**	Word 24 Data sent to fieldbus				Unsigned16	WORD
8256		**Rfieldbusdatatx25**	Word 25 Data sent to fieldbus				Unsigned16	WORD
8257		**Rfieldbusdatatx26**	Word 26 Data sent to fieldbus				Unsigned16	WORD
8258		**Rfieldbusdatatx27**	Word 27 Data sent to fieldbus				Unsigned16	WORD

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
8259		**Rfieldbusdatx28**	Word 28 Data sent to fieldbus				Unsigned16	WORD
8260		**Rfieldbusdatx29**	Word 29 Data sent to fieldbus				Unsigned16	WORD
8261		**Rfieldbusdatx30**	Word 30 Data sent to fieldbus				Unsigned16	WORD
8262		**Rfieldbusdatx31**	Word 31 Data sent to fieldbus				Unsigned16	WORD
8263		**Rfieldbusdatx32**	Word 32 Data sent to fieldbus				Unsigned16	WORD

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0413	0x219D	**Rmodulcmd**	Operation mode with module cyclic quota B0 : Forward module B1 : Backward module B2 : Minimum distance module B3 : Encoder 1 module B4 : Encoder 2 module B5 : Encoder 3 module B6 : Absolute encoder module		0	65535	Unsig-ned16	WORD SAVE NO_MAP
0414-415	0x219E	**Rmodulpos**	Module quota value for positioner	Step	1	0x7FFFFFFF	Unsig-ned32	H- LWORD SAVE NO_MAP
0416-417	0x21A0	**Rmodulmo-tenc**	Module quota value for motor encoder	Enc. Pulses	1	0x7FFFFFFF	Unsig-ned32	H-LWORD SAVE NO_MAP
0418-419	0x21A2	**Rmodu-lextenc**	Module quota value for external encoder	Enc. Pulses	1	0x7FFFFFFF	Unsig-ned32	H-LWORD SAVE NO_MAP

Management of module quota function (cyclic or Rollover)

With the register Rmodulcmd it is possible to enable the management of the quota in cyclic mode (or rollover).
The quota of the positioner is closed in a loop between a minimum value of 0 and the maximum limit set in the registrer Rmodulpos. Through the bits from 3 to 6 of the module Rmodulcmd it is possible to do the same for the motor encoder 1-2-3 or the absolute encoder.

The quota in module is used in the management of rotary tables, where there is an accurate angular position of the table and the positions cyclically repeat themselves.
With this mode it is possible to indicate an absolute quota inside the module range, and that quota is reached, even if the table must execute a “rollover” of the quota, both forward or backward.

By using the mode “always forward direction”(BIT0=1), each quota is reached by letting the motor (or the table) rotate always in forward direction. As an example, if we are at quota 500 and we give a GO at quota 400, the new position is reached by arriving at the maximum quota of the module, then by resetting it at 0 when the 0 point is reached, and then by advancing until the quota reaches out 400.

By using the mode “always backward direction” (BIT1=1), the sequence is the same as above, but with the rotation direction always set “backward”. So, if we are at quota 500 and we give a go at 600, the new position is reached by reaching quota 0, then by executing the rollover on the maximum quota of the module, and then by proceeding with the back direction until quota 600 is reached.
The mode “minumim distance” (BIT2=1), before starting the movement, executes a check of which is the direction that brigs to the achievement of the desired quota by following the shortest path.

Once the control has decided the sense of rotation of the motor (or the table), the operation mode is restored to mode “always forward” or “always backward”.
Regarding encoders, the range of the quotas is always between 0 and a maximum value indicated in the preset encoder module register.
The quotas of he motor encoder and of the external encoder are recorded in the usual registers Rmotenc and Rextenc.

The registers used in this operation mode are:
Rmodulcmd: Module opration mode selection register.
Rmodulpos: Module quota for the positioner. It’s the maximum reachable quota from the positioner before executing the rollover of the quota.
Rmodulmotenc: Module quota for the encoder 1. It’s the maximum reachable quota from the register of the encoder 1 before executing the rollover of the quota.
Rmodulextenc: Module quota for the encoder 2. It’s the maximum reachable quota from the register of the encoder 2 before executing the rollover of the quota.

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0420	0x21A4	**Rlineupcmd**	Quotas realignment command B0 : positioner quota B1 : encoder 1 quota B2 : encoder 2 quota B3 : encoder 3 quota B4 : absolute encoder quota				Unsig-ned16	WORD MAP WRITE
0421-422	0x21A5	**Rlineuppos**	Positioner realign quota value	Step			Signed 32	H-LWORD SAVE NO_MAP
0423-424	0x21A7	**Rlineupmo-tenc**	Encoder 1 realign quota value	Enc. Pulses			Signed32	H-LWORD SAVE NO_MAP
0425-426	0x21A9	**Rlineu-pextenc**	Encoder 2 realign quota value	Enc. Pulses			Signed 32	H-LWORD SAVE NO_MAP
0427	0x21AB	**Rlineupdef**	Enable quotas realignament from an external com-mand (input)When the external digital command arri-ves, these bit are copied in the register Rlineupcmd. B0 : Rialign positioner quota B1 : Rialign motor encoder quota B2 : Rialign external encoder quota				Unsig-ned16	WORD SAVE MAP WRITE

Quota realignment management

It is possible to realign the actual requested quota, the motor encoder quota and the external encoder quota to a prefi-xed values by acting on the register Rlineupcmd.

Rlineupcmd is managed at bit. The bits have the followin meanings:
B0: Realign positioner quota to register Rposactreq.
B1: Realign motor encoder quota to register Rmotenc.
B2: Realign external encoder quota to register Rextenc.

Once the realignment is completed, the register Rlineupcmd is reset to 0, and it waits for another realignment com-mand.

The realignment bits must be transferred all together, in order to avoid the possible loss of realignment requests.

The registers which contain the realignment quota must be already set before giving the realignment.

Essentially, the realignment function copies the lineup registers on the work registers:

Rlineuppos → Rposactreq
Rlineupmotenc → Rmotenc
Rlineupextenc → Rextenc

During the realignment, the interrupts of the DSP are blocked, so in case of realignment of more than one quota, these will be executed in the same moment.

It is advised not to use the realignment quota during the deceleration ramp.

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0331	0x214B	**Rtouchpro-befunc**	Controlword of the Touch Probe function. B0: 0=Switch off touch probe 1 1=Enable touch probe 1 B1: 0=Trigger first event 1=Continuous B2: 0=Trigger with t.p.1 input 1=Trigger with TOP encoder B3: Reserved B4: 0=Switch off sampling at t.p.1 1=Enable sampling at t.p.1 B5: Not supported B7-6: 00=Touch Rposact 01=Touc+E323h Motor Encoder 10=Touch External Encoder 11=Touch Absolute Encoder B8: 0=Switch off touch probe2 1=Enable touch probe 2 B9: 0=Trigger first event 1=Continuous B10: 0=Trigger with t.p.2 input 1=Trigger with TOP encoder B11: Reserved B12: 0=Switch off sampling at t.p.2 1=Enable sampling at t.p.2 B13: Not supported B15-14: 00=Touch Rposact 01=Touch Motor Encoder 10=Touch External Encoder 11=Touch Absolute Encoder **From firmware rev. 6 replaced by register 10084**		0	65535	Unsig-ned16	WORD SAVE MAP WRITE
0332	0x214C	**Rtouchpro-bestatus**	Statusword of the Touch Probe function. B0: 0=Touch Probe 1 is switched off 1=Touch Probe 1 is enabled B1: 0=T.P. 1 no value stored 1=Touch Probe 1 value stored B2: Not supported B3: Reserved B4: Reserved B5: Reserved B6: Reserved B7: Shall toggle with every update of T.P.1 value stored B8: 0=Touch Probe 2 is switched off 1=Touch Probe 2 is enabled B9: 0=T.P. 2 no value stored 1=Touch Probe 2 value stored B10: Not supported B11: Reserved B12: Reserved B13: Reserved B14: Reserved B15: Shall toggle with every update of T.P.2 value stored **From firmware rev. 6 replaced by register 10085**				Unsig-ned16	WORD RO MAP READ
0333-34	0x214D	**Rtouchpro-bepos1pos**	Touch Probe 1 position value at positive edge of t.p.1 touch signal **From firmware rev. 6 replaced by register 10086-87**				Signed32	WORD RO MAP READ
0335-36	0x214F	**Rtouchpro-bepos1neg**	Touch Probe 1 position value at negative edge of t.p.1 touch signal **From firmware rev. 6 replaced by register 10088-89**				Signed32	WORD RO MAP READ
0337-38	0x2151	**Rtouchpro-bepos2pos**	Touch Probe 2 position value at positive edge of t.p.2 touch signal **From firmware rev. 6 replaced by register 10097-98**				Signed32	WORD RO MAP READ
0339-40	0x2153	**Rtouchpro-bepos2neg**	Touch Probe 2 position value at negative edge of t.p.2 touch signal **From firmware rev. 6 replaced by register 10099-100**				Signed32	WORD RO MAP READ

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0341	0x2155	**Rtouchpro-be1inpdef**	Setting of the digital input to be used for the touch probe 1 function **From firmware rev. 6 replaced by register 10090**		0	255	Unsig-ned16	WORD RO MAP WRITE
0342	0x2156	**Rtouchpro-be2inpdef**	Setting of the digital input to be used for the touch probe 2 function **From firmware rev. 6 replaced by register 10101**		0	255	Unsig-ned16	WORD RO MAP WRITE
10084	0x4764	**Rtouchpro-befunc**	Control word of the Touch Probe function		0	65535	Unsig-ned16	RW SAVE
10085	0x4765	**Rtouchpro-bestatus**	Status word of the Touch Probe function		0	65535	Unsig-ned16	RW SAVE
10086-87	0x4766	**Rtouchpro-bepos1pos**	Touch Probe 1 position value at positive edge of t.p.1 touch signal				Signed32	RO
10088-89	0x4768	**Rtouchpro-bepos1neg**	Touch Probe 1 position value at negative edge of t.p.1 touch signal				Signed32	RO
10090	0x476A	**Rtouchpro-be1inp**	Touch Probe 1 source				Unsig-ned16	RW SAVE
10091-92	0x476B	**Rtouchpro-be1time-stamppos**	Touch probe time stamp 1 positive value				Unsig-ned32	RO
10093-94	0x476D	**Rtouchpro-be1time-stampneg**	Touch probe time stamp 1 negative value				Unsig-ned32	RO
10095	0x476F	**Rtouchpro-be1posed-gecnt**	Touch probe positive edge counter				Unsig-ned16	RO
10096	0x4770	**Rtouchpro-be1neged-gecnt**	Touch probe negative edge counter				Unsig-ned16	RO
10097-98	0x4771	**Rtouchpro-bepos2pos**	Touch Probe position 2 positive value				Signed32	RO
10099-100	0x4773	**Rtouchpro-bepos2neg**	Touch Probe position 2 negative value				Signed32	RO
10101	0x4775	**Rtouchpro-be2inp**	Touch Probe 2 source				Unsig-ned16	RW SAVE
10102-3	0x4776	**Rtouchpro-be2time-stamppos**	Touch probe time stamp 2 positive value				Unsig-ned32	RO
10104-5	0x4778	**Rtouchpro-be2time-stampneg**	Touch probe time stamp 2 negative value				Unsig-ned32	RO
10106	0x477A	**Rtouchpro-be2posed-gecnt**	Touch probe 2 positive edge counter				Unsig-ned16	RO
10107	0x477B	**Rtouchpro-be2neged-gecnt**	Touch probe 2 negative edge counter				Unsig-ned16	RO
10108	0x477C	**Rtouchpro-be1source**	Touch Probe 1 input definition 0 = Rposact 1 = Encoder 1 2 = Encoder 2 3 = Encoder 3 4 = Absolute encoder				Unsig-ned16	RW SAVE
10109	0x477D	**Rtouchpro-be2source**	Touch Probe 2 encoder input definition 0 = Rposact 1 = Encoder 1 2 = Encoder 2 3 = Encoder 3 4 = Absolute encoder				Unsig-ned16	RW SAVE

Touch probe function

The touch probe function records an axis position at the point in time of an input digital signal.
Since the position is usually not recorded directly in the PLC, but via an external hardware latch, it is highly accurate and independent of cycle time. The touch probe function controls this mechanism and determines the externally recorded position.

Rtouchprobefunc (CanOpen/EtherCAT standard Object 60B8h) – Touch probe function

Modbus register	CanOpen Index	CanOpen SubIndex	Parameter Name	Data type	Access type	Default value	PDO mapping
“0331”, or “10084” (from firmware rev. 6)	60B8h	0	**Rtouchprobefunc** Touch probe function	Unsigned 16	RW	0	Yes

This object indicates the configured function of the touch probe.

Notes: Bit 0 to 7: for touch probe 1
Bit 8 to 15: for touch probe 2
Bit2/10 cannot be changed after 60B8h Bit4/12 was set to 1.

Bit No.	Value	Definition
0	0	Switch off touch probe 1
	1	Enable touch probe 1
1	0	Trigger first event
	1	continuous
2	0	Trigger with touch probe 1 input
	1	Trigger with zero signal of position encoder
3	-	Reserved
4	0	Switch off sampling at touch probe 1
	1	Enable sampling at touch probe 1
5	-	not supported
6, 7	0	User-defined (not used)
8	0	Switch off touch probe 2
	1	Enable touch probe 2
9	0	Trigger first event
	1	continuous
10	0	Trigger with touch probe 2 input
	1	Trigger with zero signal of position encoder
11	0	Reserved
12	0	Switch off sampling at touch probe 2
	1	Enable sampling at touch probe 2
13	0	not supported
14, 15	0	User-defined (not used)

Rtouchprobestatus (CanOpen/EtherCAT standard Object 60B9h) - Touch probe status

Modbus register	CanOpen Index	CanOpen SubIndex	Parameter Name	Data type	Access type	Default value	PDO mapping
“0332”, or “10085” (from firmware rev. 6)	60B9h	0	**Rtouchprobestatus** Touch probe Status	Unsigned 16	RO	0	Yes

This object provides the status of the touch probe.

Value range: Unsigned16

Bit No.	Value	Definition
0	0	Touch probe 1 is switched off
	1	Touch probe 1 is enabled
1	0	Touch probe 1 no value stored
	1	Touch probe 1 value stored
2	0	not supported
3 to 6	0	Reserved
7	0,1	Shall toggle with every update of Touch probe 1 value stored *1
8	0	Touch probe 2 is switched off
	1	Touch probe 2 is enabled
9	0	Touch probe 2 no value stored
	1	Touch probe 2 value stored
10	0	not supported
11 to 14	0	Reserved
15	0,1	Shall toggle with every update of Touch probe 2 value stored *1

Notes: Bit 0 to 7: for touch probe 1
Bit 8 to 15: for touch probe 2

*1) If the continuous latch is enabled (object 60B8 bit 1 = 1, or bit 9 = 1), bit 7 or bit 15 of object 60B9h is toggled with every stored update of the touch probe value.

Rtouchprobepos1pos (CanOpen/EtherCAT standard Object 60BAh) - Touch probe position 1 positive value

Modbus register	CanOpen Index	CanOpen SubIndex	Parameter Name	Data type	Access type	Default value	PDO mapping
“0333-34” “10086-87” (from firmware rev. 6)	60BAh	0	**Rtouchprobepos1pos** Touch probe position 1 positive value	Integer 32	RO	0	Yes

This object provides the position value of the touch probe 1.
The value shall be given in user-defined position units.

Value range: Integer32

Units: Pos units

Rtouchprobepos1neg (CanOpen/EtherCAT standard Object 60BBh) - Touch probe position 1 negative value

Modbus register	CanOpen Index	CanOpen SubIndex	Parameter Name	Data type	Access type	Default value	PDO mapping
“0335-36” “10088-89” (from firmware rev. 6)	60BBh	0	**Rtouchprobepos1neg** Touch probe position 1 negative value	Integer 32	RO	0	Yes

This object provides the position value of the touch probe 1.
The value shall be given in user-defined position units.

Value range: Integer32

Units: Pos units

Rtouchprobepos2pos (CanOpen/EtherCAT standard Object 60BCh) - Touch probe position 2 positive value

Modbus register	CanOpen Index	CanOpen SubIndex	Parameter Name	Data type	Access type	Default value	PDO mapping
“0337-38” “10097-98” (from firmware rev. 6)	60BCh	0	**Rtouchprobepos2pos** Touch probe position 2 positive value	Integer 32	RO	0	Yes

This object provides the position value of the touch probe 2.
The value shall be given in user-defined position units.

Value range: Integer32

Units: Pos units

Rtouchprobepos2neg (CanOpen/EtherCAT standard Object 60BDh) - Touch probe position 2 negative value

Modbus register	CanOpen Index	CanOpen SubIndex	Parameter Name	Data type	Access type	Default value	PDO mapping
“0339-40” “10099-100” (from firmware rev. 6)	60BDh	0	**Rtouchprobepos2neg** Touch probe position 2 negative value	Integer 32	RO	0	Yes

This object provides the position value of the touch probe 2.
The value shall be given in user-defined position units.

Value range: Integer32

Units: Pos units

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0345	0x2159	**Rcaptselinp**	Selection of the digital input to be used for “capture” function.		0	15	Unsig-ned16	WORD SAVE
0346-47	0x215A	**Rcaptvel-max**	Maximum value reachable by the timer in “capture” function. (Time base 20us)				Unsig-ned32	WORD SAVE
0348-49	0x215C	**Rcaptval**	Value of the period registerd by the “capture” function between two edges of the digital input. (Time base 20us)				Unsig-ned32	WORD
0350	0x215E	**Rcaptcnt**	Number of pulses received on the digital input asso-ciated to the “capture” function. It ranges from 0 to 65535. Once the upper limit is reached, it restarts from 0.		0	65535	Unsig-ned16	WORD

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0380	0x217C	**Rcammstpul-se**	Master encoder pulses per revolution (for cams Type1 and Type2)	Enc. Pulses	1	32767	Unsig-ned16	WORD SAVE NO MAP
0381	0x217D	**Rcammst-maxfrq**	Master encoder maximum pulses/sec (for cam Type2)	Enc. Pulses /s	1	32767	Unsig-ned16	WORD SAVE NO MAP
0382	0x217E	**Rcamstrtpha-se**	Number of the task with the first phase to be executed (for cams Type1 and Type2)		0	63	Unsig-ned16	WORD SAVE MAP READ
0383	0x217F	**Rcammstpo-sact**	Master encoder actual position(for cams Type1 and Type2)	Enc. Pulses			Unsig-ned16	WORD MAP READ RO
0384	0x2180	**Rcamphase-act**	Actual phase (task) in progress (for cams Type1 and Type2)				Unsig-ned16	WORD NO MAP RO
0385	0x2181	**Rcamflgwr**	Cam writing flag Bit 0: Cam reset Bit 1: Only one revolution of the cam. then stops at the maximum value. Bit 2: Bit 3: Bit 4: Bit 5: Bit 6: Bit 7: Bit 8: Bit 9: Bit 10: Bit 11: Bit 12: Bit 13: Bit 14: Bit 15: (for cams Type1 and Type2)				Unsig-ned16	WORD NO MAP
0386	0x2182	**Rcamflgrd**	Cam reading flag Bit 0: Cam reset Bit 1: Cam in movement Bit 2: Camma in acceleration Bit 3: Camma at constant speed Bit 4: Camma in deceleration Bit 5: Bit 6: Bit 7: Bit 8: Bit 9: Bit 10: Bit 11: Bit 12: Bit 13: Bit 14: Bit 15: (for cams Type1 and Type2)				Unsig-ned16	WORD MAP READ RO
0387	0x2183	**Rcamkp**	Kp PI increments management in cam mode (for cams Type1, Type2, STSP and Winding)		0	32767	Unsig-ned16	WORD SAVE NO MAP

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0456	0x21C8	**Rcamst-spmstmm-pulse**	Millimeters/pulse master shift in start/stop cam (mm*1000/pulse)	mm/ Imp.	0	65535	Unsig-ned16	WORD SAVE
0457	0x21C9	**Rcamst-spslvmm-step**	Millimeters/pulse slave shift in start/stop cam (mm*1000/pulse)	mm/ Imp.	0	65535	Unsig-ned16	WORD SAVE
0458	0x21CA	**Rcamst-spcor**	Master/slave centesimal correction in start/stop cam (-100/+100)	%	-1000	1000	Signed16	WORD SAVE
0459	0x21CB	**Rcamst-spaccspace**	Slave acceleration space on master space in start/ stop cam	mm*10	0	65535	Unsig-ned16	WORD SAVE
0460	0x21CC	**Rcamstspc-trl**	Start/stop cam control B0: 0=Stop/1=Run B1: 1=Disable start delay B2: 1=Disable stop delay B3: 1=Simulated encoder B4: B5: B6: B7:				Unsig-ned16	WORD MAP WRITE
0461	0x21CD	**Rcamst-spstatus**	Start/stop cam status B1: Cam enabled B2: Cam in RUN B3: Cam in acceleration B4: Cam at constant acceleration B5: Cam in decelaration B5: B6: B7:				Unsig-ned16	WORD RO MAP READ
0462-63	0x21CE	**Rcamst-splowquo-teslv**	Minimum slave quota in start/stop cam	Step	0	0x7FFFFFFF	Unsig-ned32	H-LWORD SAVE
0464-65	0x21D0	**Rcamstsphi-ghquoteslv**	Maximum slave quota in start/stop cam	Step	0	0x7FFFFFFF	Unsig-ned32	H-LWORD SAVE
0466-67	0x21D2	**Rcamst-splowquo-temst**	Minimum master quota in start/stop cam	Enc. Pulses	0	0x7FFFFFFF	Unsig-ned32	H-LWORD SAVE
0468-69	0x21D4	**Rcamstsphi-ghquotemst**	Maximum master quota in start/stop cam	Enc. Pulses	0	0x7FFFFFFF	Unsig-ned32	H-LWORD SAVE
0470	0x21D6	**Rcammode-select**	Cam mode selection 0=Type1 cam (Blk) 1=Type2 cam (Bert)		0	1	Unsig-ned16	WORD SAVE
0496	0x21F0	**Rcamst-spsimvel**	Speed in mm/s of the simulated encoder in start/stop cam function	mm/s			Unsig-ned16	WORD SAVE
0497	0x21F1	**Rcamst-spdecspace**	Slave deceleration space on master space in start/ stop cam	mm*10			Unsig-ned16	WORD SAVE
0498	0x21F2	**Rcamst-spstartdly**	Mater space before the slave start in start/stop cam mode	mm*10			Unsig-ned16	WORD SAVE
0499	0x21F3	**Rcamst-spstopdly**	Slave space from the stop signal to the start of the deceleration ramp	mm*10			Unsig-ned16	WORD SAVE

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0471	0x21D7	**Rwindspoolimpturn**	Spool encoder pulses per revolution in winding mode	Enc. Pulses	1	65535	Unsigned16	WORD SAVE
0472	0x21D8	**Rwindyarnguidemmstep**	Wire guide shift in mm for stepper mode in winding mode	mm* 1000	0	65535	Unsigned16	WORD SAVE
0473	0x21D9	**Rwindaccspoolturn**	Wire guide acceleration turns to reach the requested speed in winding mode	1/10 turn	0	65535	Unsigned16	WORD SAVE
0474	0x21DA	**Rwindyarnshift**	Wire guide shift for spool revolution in winding mode	mm* 100	0	65535	Unsigned16	WORD SAVE
0475	0x21DB	**Rwindtyingshift**	Wire guide shift for spool revolution during the binding phase in winding mode	mm* 100	0	65535	Unsigned16	WORD SAVE
0476	0x21DC	**Rwindctrl**	Control word winding mode B0: 1=Enable winding cam B1: 1=Enable stop sequence B2: 1=Enable binding sequence B3: 1=Disable encoder reading B4: 1=Winding cam error reset B5: Start direction of the deposit				Unsigned16	WORD MAP WRITE
0477	0x21DD	**Rwindstatus**	Status word winding mode B0: 1=Winding cam enabled B1: 1=Stop sequence request enabled B2: 1=Binding sequence request enabled B3: 1=Encoder disabled B4: Cam stepper motor direction flag B5: 1=End of cam sequence B6: 1=Cam error B7: 1=Binding sequence executed B8: B9: B10: B11: B12: B13: Cam in acceleration B14: Cam in constant speed B15: Cam in deceleration				Unsigned16	WORD RO MAP READ
0478-79	0X21DE	**Rwindquoteleft**	Quota toward wire guide machine in winding mode	mm*10	0	0x7FFFFFFF	Unsigned32	H-LWORD
0480-81	0x21E0	**Rwindquoteright**	Quota toward wire guide user in winding mode	mm*10	0	0x7FFFFFFF	Unsigned32	H-LWORD
0482-83	0x21E2	**Rwindquotestop**	Wire guide stop quota in winding mode	mm*10	0	0x7FFFFFFF	Unsigned32	H-LWORD
0484-85	0x21E4	**Rwindquotetyingleft**	Quota toward wire guide machine for binding in winding mode	mm*10	0	0x7FFFFFFF	Unsigned32	H-LWORD
0486-87	0x21E6	**Rwindquotetyingright**	Quota toward wire guide user for binding in winding mode	mm*10	0	0x7FFFFFFF	Unsigned32	H-LWORD
0488-89	0x21E8	**Rwindyarnguidemmturnmotor**	Wire guide shift in mm per stepper motor revolution	mm* 100	0	0x7FFFFFFF	Unsigned32	H-LWORD
0490	0x21EA	**Rwindrevcnt**	Reversals counter in winding mode				Unsigned16	WORD MAP READ
0491-92	0X21EB	**Rwindrevpause**	Pulses of master encoder with stepper motor in pause during the reversal of the direction of motion	Enc. Pulses	0	0x7FFFFFFF	Unsigned32	H-LWORD
10012	0x471C	**Renccamfbk**	Encoder feedback for cam/winding functions 0= Encoder 1 1= Encoder 2 2= Encoder 3 3= Absolute encoder		Default 1		Unsign16	"WORD RW SAVE"

FUNCTIONAL TASK WITH DIGITAL OUTPUTS FUNCTION

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0530	0x2212	**Rtskfmstfw1**	Experimental function for movement from task table + outputs management				Unsig-ned16	WORD
0531	0x2213	**Rtskfmstfw2**	Experimental function for movement from task table + outputs management				Unsig-ned16	WORD
0532	0x2214	**Rtskfmst-macvelact**	Experimental function for movement from task table + outputs management				Unsig-ned16	WORD
0533	0x2215	**Rtskfmst-macvelmax**	Experimental function for movement from task table + outputs management				Unsig-ned16	WORD
0534	0x2216	**Rtskfmst-spcphdis**	Experimental function for movement from task table + outputs management				Unsig-ned16	WORD
0535	0x2217	**Rtskfmstphinp**	Experimental function for movement from task table + outputs management				Unsig-ned16	WORD

EXTERNAL BRAKE

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0098	0x2062	**Rbrakedlyo-pen**	Opening delay of external brake, commanded by digital output	ms	0	65535	Unsig-ned16	WORD SAVE
0344	0x2158	**Rbra-kedlyclose**	Closing delay of external brake, commanded by digital output	ms	0	65535	Unsig-ned16	WORD SAVE

PID FUNCTION

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0500	0x21F4	**Rusrpidkp**	User PID proportional gain				Unsig-ned16	WORD SAVE
0501	0x21F5	**Rusrpidki**	User PID integral gain				Unsig-ned16	WORD SAVE
0502	0x21F6	**Rusrpidkd**	User PID derivative gain				Unsig-ned16	WORD SAVE
0503	0x21F7	**Rusrpidkt**	User PID integration time in mS	ms			Unsig-ned16	WORD SAVE
0504	0x21F8	**Rusrpidflag**	User PID control flag B0: 0=Stop PID / 1=Start PID B1: 1=Reset PID B2: B3: B4: B5:				Unsig-ned16	WORD
0505-06	0x21F9	**Rusrpidma-xout**	User PID maximum output value				Signed32	WORD SAVE
0507-08	0x21FB	**Rusrpidmi-nout**	User PID minimum output value				Signed32	WORD SAVE
0509-10	0x21FD	**Rusr-pidsetpoint**	User PID Setpoint				Signed32	WORD SAVE MAP WRITE
0511-12	0x21FF	**Rusrpidpro-cessvalue**	User PID process value				Signed32	WORD MAP READ
0513-14	0x2201	**Rusrpidout**	User PID output value				Signed32	WORD MAP READ

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0315	0x213B	**Rfgapplyto**	Data to which the internal function generator signal must be applied: 0: None 1: Current Phase A 2: Current Phase B 3: Speed 4: Position 5: Profile		0	5	Unsig-ned16	WORD
0316-17	0x213C	**Rfgvalmin**	Function generator minimum value	Bit	0x80000000	0x7FFFFFFF	Signed32	LS – MSWORD NO MAP RO
0318-19	0x213E	**Rfgvalmax**	Function generator maximum value	Bit	0x80000000	0x7FFFFFFF	Signed32	LS – MSWORD NO MAP RO
0320	0x2140	**Rfgfreq**	Frequency of the generated signal	Hz	1	10000	Unsig-ned16	WORD NO MAP
0321	0x2141	**Rfgperiod**	Period of the generated signal	ms	1	1000	Unsig-ned16	WORD NO MAP
0322	0x2142	**Rfgcmd**	Function generator command 0 = FG stop 1 = FG start		0	1	Unsig-ned16	WORD SAVE NO MAP
0323	0x2143	**Rfgmode**	Type of signal generated 0 = sinusoidal 1= square wave 2 = pulse		0	2	Unsig-ned16	WORD
0324-25	0x2144	**Rfgout32**	Functions generator output				Signed32	LS – MSWORD NO MAP RO
0326	0x2146	**Rfgout16**	Functions generator output				Signed16	WORD

Indirizzo Modbus	Nome Registro	Descrizione	U.M.	Valore Minimo	Valore Massimo	Tipo	
7000		Oscilloscope data 0 channel 1				Unsigned16	WORD
7001-7498		Oscilloscope data 1-498 channel 1				Unsigned16	WORD
7499		Oscilloscope data 499 channel 1				Unsigned16	WORD
7500		Oscilloscope data 0 channel 2				Unsigned16	WORD
7501-7998		Oscilloscope data 1-498 channel 2				Unsigned16	WORD
7999		Oscilloscope data 499 channel 2				Unsigned16	WORD
8000		Channel 1 value divider Nuber of right Shifts				Unsigned16	WORD
8001		Channel 2 value divider Nuber of right Shifts				Unsigned16	WORD
8002		Trigger value divider Nuber of right Shifts				Unsigned16	WORD
8003		Ch1 data Modbus address				Unsigned16	WORD
8004		Ch2 data Modbus address				Unsigned16	WORD
8005		Trigger data Modbus address				Unsigned16	WORD
8006		Value for trigger				Unsigned16	WORD
8007		Trigger type Bit 0: 0=Single / 1=Continuous Bit 1: 1=Arm trigger in single mode Bit 2: 1=Immediate start Bit 3: 1=Start if trigger > Bit 4: 1=Start if trigger < Bit 5: 1=Acquisition stop Bit 6: Bit 7:				Unsigned16	WORD
8008		Pretrigger points				Unsigned16	WORD
8009		Times base Number of 250uS temporal quantum				Unsigned16	WORD
8010		Oscilloscope commands B0: Enable oscilloscope B1: B2: B3: B4: B5: B6: B7:				Unsigned16	WORD
8011		Oscilloscope Status B0: Oscilloscope enabled B1: Copy of the bit “Arm trigger in single mode” B2: Trigger armed B3: Acquisition in progress B4: Data buffer ready B5: B6: B7:				Unsigned16	WORD

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
0204	0x20CC	**Rstsprg**	MIL program status register Bit 0: RUN (Program in RUN if 1) Bit 1: PRG_OK (Program not valid if 0) Bit 2: Bit 3: Bit 4: Bit 5: Bit 6: Bit 7: Bit 8:				Unsig-ned16	WORD NO MAP RO
0205	0x20CD	**Rprgrunmo-de** **Run mode User Pro-grammare** **0 = Stop** **1 = Single step** **2 = Run**	Run mode User Program 0 = Stop 1 = Single step 2 = Run		0	2	Unsig-ned16	WORD NO MAP
0206	0x20CE	**Rprgstope-vent**	Stop event status register (value reading) 0: No stop request 1: Stop from manual command 2: Not valid program 3: Program Counter over the end of the program 4: Program Counter over the end of the program memory 5: Not valid instruction 6: Interrupt event manager overflow stack pointer 7: END Instruction 8: Excessive number of subroutines 9: Excessive number of returns from subroutine(corrupted stack) 10: Excessive number of interrupt ONH events (ONH+ONL<10) 11: Excessive number of interrupt ONL events (ONH+ONL<10) 12: Excessive number of returns from interrupt events (corrupted stack)				Unsig-ned16	WORD RO
0207	0x20CF	**Rprgcnt**	Program counter User program N.B. it can be written only if user program is in stop.		0	4998	Unsig-ned16	WORD NO MAP
0208	0x20D0	**Rprgccflag**	Condition code flag B0: Carry B1: Overflow B2: Zero B3: Negative B4: Interrupt (1 ONH and ONL instruction mask)				Unsig-ned16	WORD NO MAP
2000		**Rprgusrpsw**	Password entered for the enabling of the access to the user program memory location. It is not stored in the eeprom. If not equal to the value in the eeprom, the user program memory location is not accessible neither in read nor in write.				Unsig-ned16	WORD
2001		**Rprgcmd**	Coomands 0x01: Save program in EEPROM 0x02: Load program from EEPROM 0x81: Delete password (delete the whole program) 0x82: Set new password 0x83: Delete program in RAM				Unsig-ned16	WORD
2002		**Rprgcmdsts**	Command status Last command output status				Unsig-ned16	WORD
2003		**Rprgnew-psw**	New Password				Unsig-ned16	WORD
2004		**Rprgpswact**	Password active 0: Program unlocked 1: Program protected by password				Unsig-ned16	WORD

Modbus Profibus address	CAN EtherCAT Profinet address	Register name	Description	U.M.	Minimum value	Maximum value	Type	Note
2005		**Rprgbpena-ble**	Enable the software breakpoint management The value is expressed in byte				Unsig-ned16	WORD
2006		**Rprgbpnum**	Number of the breakpoint to be managed in read/write with Rprgbpval 0= Number of instructions to be executed in single step mode 1= Break point number 1 address 2= Break point number 2 address 3= Break point number 3 address 4= Break point number 4 address				Unsig-ned16	WORD
2007		**Rprgbpval**	Value read or to be written in the pointed braskpoint.				Unsig-ned16	WORD
2008		**Rprglen**	Length of the user program				Unsig-ned16	WORD
2009		**Rprgprgcks**	User program checksum				Unsig-ned16	WORD
2010		**Rprgusrprg**	User program location 0				Unsig-ned16	WORD
2011-5008			User program locations nnn				Unsig-ned16	WORD
6999			User program location 4998				Unsig-ned16	WORD

VARIABLES READ/WRITE

With this object, it is possible to read and write the value of the internal variables of the drives.

Index of the object

Index	0x23E8				
Symbol	read_write_variable	**Length (byte)**	4	**Min value**	
Object Code	Record	**Elements**	7	**Max value**	
Data Type		**Access**		**Default value**	
		PDO mapping			

Sub-index 0: number of sub-indexes present in the object

Index	0x23E8:00				
Symbol	number_of_entries	**Length (byte)**	1	**Min value**	
Object Code	Variable	**Elements**		**Max value**	
Data Type	Unsigned8	**Access**	Read	**Default value**	0x02
		PDO mapping	No		

Read/write via SDO

To access the variables, it's necessary to set the number of the variable in the sub-index 01, and read or write the value in the sub-index 02.

Index	0x23E8:01				
Symbol	variable_number	**Length (byte)**	2	**Min value**	0x0001
Object Code	Variable	**Elements**		**Max value**	0x0080
Data Type	Unsigned16	**Access**	Read/Write	**Default value**	0x0001
		PDO mapping	No		

Index	0x23E8:02				
Symbol	variable_value	**Length (byte)**	4	**Min value**	0x80000000
Object Code	Variable	**Elements**		**Max value**	0x7FFFFFFF
Data Type	Integer32	**Access**	Read/Write	**Default value**	0x00000000
		PDO mapping	No		

Write via PDO

To write a variable via PDO, it's necessary to set the number of the variable in the sub-index 03, and write the value in the sub-index 04.

Index	0x23E8:03				
Symbol	variable_number	**Length (byte)**	2	**Min value**	0x0001
Object Code	Variable	**Elements**		**Max value**	0x0080
Data Type	Unsigned16	**Access**	Read/Write	**Default value**	0x0001
		PDO mapping	Write		

Index	0x23E8:04				
Symbol	variable_value	**Length (byte)**	4	**Min value**	0x80000000
Object Code	Variable	**Elements**		**Max value**	0x7FFFFFFF
Data Type	Integer32	**Access**	Read/Write	**Default value**	0x00000000
		PDO mapping	Write		

Read via PDO

To read a variable via PDO, it's necessary to set the number of the variable in the sub-index 06, and read the value in the sub-index 06.

Index	0x23E8:05				
Symbol	variable_number	**Length (byte)**	2	**Min value**	0x0001
Object Code	Variable	**Elements**		**Max value**	0x0080
Data Type	Unsigned16	**Access**	Read/Write	**Default value**	0x0001
		PDO mapping	Write		

Index	0x23E8:06				
Symbol	variable_value	**Length (byte)**	4	**Min value**	0x80000000
Object Code	Variable	**Elements**		**Max value**	0x7FFFFFFF
Data Type	Integer32	**Access**	Read/Write	**Default value**	0x00000000
		PDO mapping	Read		

VARIABLES READ/WRITE IN MULTIAXIS DRIVES

Variables read/write procedure in multiaxis drives is the same as for single axis drives.
For the axis 1, refer to the previous chapter. For the axis 2 and 3, the sub-indexes are shifted of 10. So, for the axis 1 the sub-indexes to be used are from 01 to 06, for the axis 2 from 11 to 16, for the axis 3 from 21 to 26.

Examples:

Read/write of a varible via SDO

Axis 1: set the number of the variable in the sub-index 01 and read/write the value in the sub-index 02.
Axis 2: set the number of the variable in the sub-index 11 and read/write the value in the sub-index 12.
Axis 3: set the number of the variable in the sub-index 21 and read/write the value in the sub-index 22.

Write of a variable via PDO

Axis 1: set the number of the variable in the sub-index 03 and write the value in the sub-index 04.
Axis 2: set the number of the variable in the sub-index 13 and write the value in the sub-index 14.
Axis 3: set the number of the variable in the sub-index 23 and write the value in the sub-index 24.

Read of a variable via PDO

Axis 1: set the number of the variable in the sub-index 05 and read the value in the sub-index 06.
Axis 2: set the number of the variable in the sub-index 15 and read the value in the sub-index 16.
Axis 3: set the number of the variable in the sub-index 25 and read the value in the sub-index 26.

SIGNALATIONS

FLD LED STATUS

According to the CIA-DR303v1.3 specifications, information about the status of the bus can be read using a bicolor LED.

Color	LED_stat	Status	Description
RED	OFF	No error	Correct operation
	Single Flash	Warning limit reached	One of the CAN error counters reached or exceeded the warning level (too many frame errors)
	Double Flash	Error control event	A guard event (Node guarding/Life guarding) or a heartbeat occurred
	Triple Flash	Sync error	The SYNC message has not been received in the programmed temporal range (Communication cycle period time out 0x1006).
	Quadruple Flash	Event-timer error	No PDO message has been received before the expiry of the event-timer.
	ON	Bus off	The CAN controller is disabled (bus off)
GRN	OFF	No communic.	No active communication
	Blinking	PRE-OPERATIONAL	The drive is in PRE-OPERATIONAL status
	Single Flash	STOPPED	The drive is in STOPPED status
	ON	OPERATIONAL	The drive is in OPERATIONAL status

ERROR SIGNALATIONS

In case an alarm occurred, the drive disables the power output and the motor is stopped in non-controlled way.
The presence of errors is indicated by the object 0x1001 (Error register) and by the STS led.
The type of error that occurred is stored in a 2 locations buffer (0x1003).
In case the number of errors is greater than 2, the last occurred error will overwrite the eighth position of the buffer.

The procedure to clear the errors is the following:
1. eliminate the cause of the error;
2. write 0 in the object 0x1003:00;
3. restore the operation enable status of the drive.

CONNECTIONS

CABLE CHARACTERISTICS

The following table summarizes the main characteristics for CANopen cables class A, according to IEC61158/EN 50170 standard.

Bus length [m]	Cable		Termination resistance	Baudrate [kbit/s]
	Resistance/m [mΩ/m]	Cross section [mm²]		
0 .. 40	70	0,25 .. 0,34	124	1000 a 40m
40 .. 300	< 60	0,34 .. 0,6	150 .. 300	>500 a 100m
300 .. 600	< 40	0,34 .. 0,6	150 .. 300	>100 a 500m
600 .. 1000	< 26	0,75 .. 0,8	150 .. 300	>50 a 1000m

It is recommended to use cables with an impedance of 120Ω, and a line delay less than 5 ns/m. The following table shows the maximum length in relation to the cross section of the cable and to the number of nodes.

Cross section [mm²]	Maximum Legth [m] (1)			Maximum Length [m] (2)		
	n = 32	n = 64	n = 100	n = 32	n = 64	n = 100
0,25	200	170	150	230	200	170
0,5	360	310	270	420	360	320
0,75	550	470	410	640	550	480

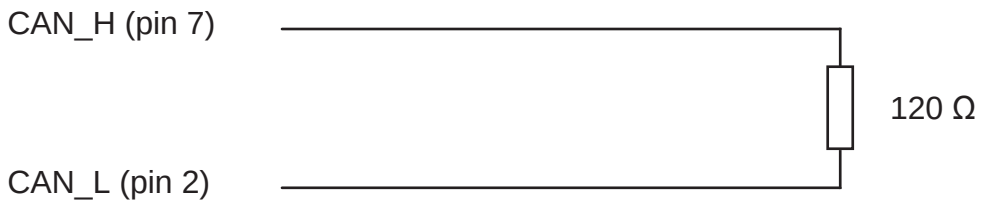
(1) Safety margin 0,2 (2) Safety margin 0,1.

STUBS

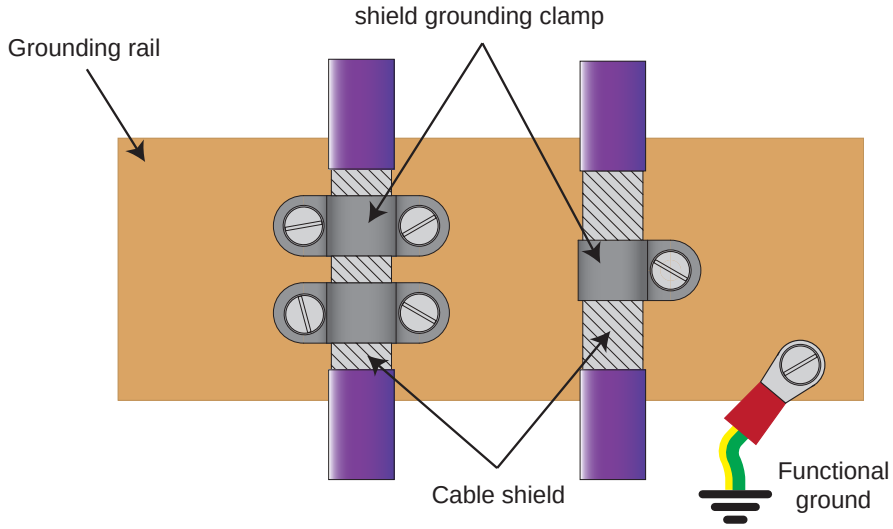
It is unrecommended to use passive stubs in communication networks.
In case of connection of diagnostic or programming devices, it is necessary to use active stubs (terminators, repetitors, active cables).

TERMINATORS

In order to reduce reflection effects and line noise, both of the sides of the CANopen network must be terminated as reported below, according to IEC61158/EN 50170 standard for 9-pin SUB-D connectors).



One of the main issues in a communication network, are EMI disturbances caused by cables or devices contained in the same power panel.
In order to reduce the influence of external agents it is necessary to follow some precautions when choosing cables and wiring methods.
The connection must be carried out with an A-class cable, according to CiA specifications, with a double shielding system (braid shield and spiral shield).
At the installation of each device, the shield of the input cable must be connected to the mass bar and to the shield of the output cable, with the shortest possible path.
The mass bar must not be fixed to painted surfaces, in order to ensure a contact to ground with the best possible conductivity.

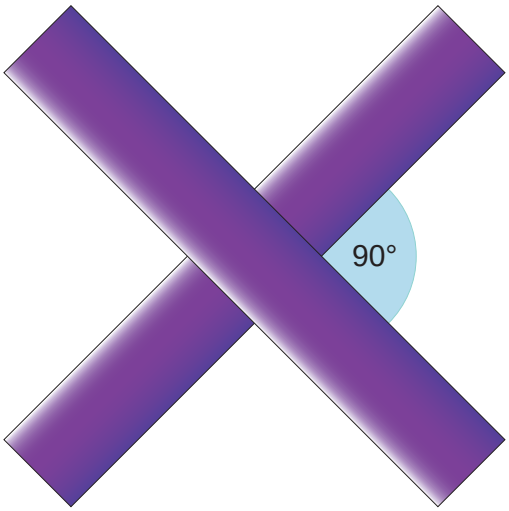


Besides a good shield, it is of fundamental importance keeping the minimum distances between CAN cables and other cables.
According to EN50174-2 standard, the minimum distance between the cables must be:

Minimum distance between CAN cables and other cables	Distance (in mm)		
	Without cable trays or with non metallic cable trays	With aluminium cable trays	With steel cable trays
Signal			
• Communication networks similar to CAN • Digital signals of low power (sensors, PLC, PC) • Shielded analog inputs or outputs	0	0	0
Power supply			
• Not shielded	200	100	50
• Shielded	0	0	0

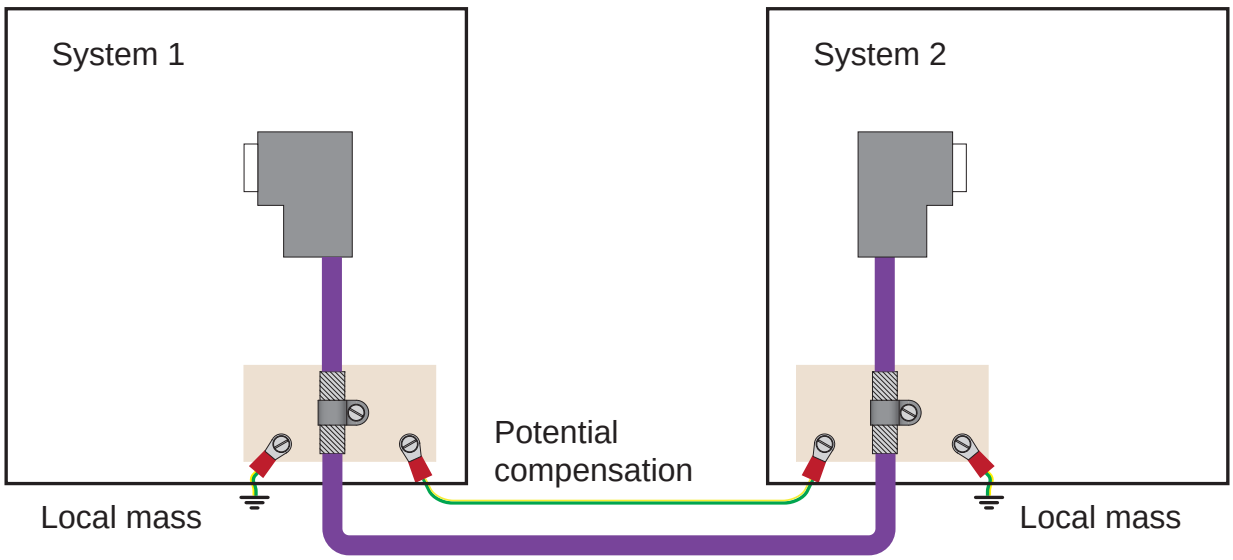
In case it is not possible to keep the minimum distances required, it is necessary to use two separate metallic rails, each one containing cables of the same category.
In this case, the two rails can be tiled.

In case it is necessary to cross the cables, it is advisable that the angle of intersection is a right angle, in order to reduce capacitive coupling of the disturbances.

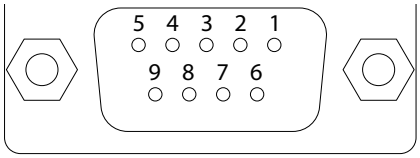


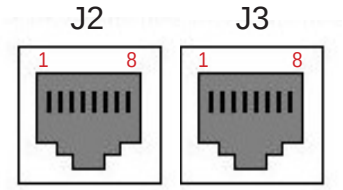
The potential differences between the masses of the nodes of a same network can cause disturbances or damage the connected devices.
In order to avoid these issues, it is necessary to:

- connect each node to the same mass potential of the system, with the shortest possible path and the lowest possible resistance.
- use a potential compensation cable in order to ensure equipotentiality.
- achieve a good grounding connection of the system.



CONNECTOR PINS

CANopen (9 ways D-SUB male connectors)			
	Pin	Name	Description
	1	-	Reserved
	2	CAN_L	CAN dominant line
	3	CAN_GND	Communication signals common ground
	4	-	Reserved
	5	CAN_SHLD	Shield
	6	-	Reserved
	7	CAN_H	CAN recessive line
	8	-	Reserved
	9	CAN_V+	5V output@100mA

CANopen (RJ45 connector)			
	Pin	Name	Description
	1	CAN_H	CAN recessive line
	2	CAN_L	CAN dominant line
	3	CAN_GND	Communication signals common ground
	4	-	Reserved
	5	-	Reserved
	6	CAN_SHLD	Shield
	7	CAN_GND	Communication signals common ground
	8	CAN_V+	5V output@100mA