CANOPEN-ETHERCAT

User guide for the CANopen communication protocol and the EtherCAT Fieldbus, available for NTT, TOMCAT EVO e DGFOX EVO drive.

USER GUIDE

Version data	Changes Description
01/04/2021 Rev: 4.0	• First release, firmware: 4.28, Caliper: 4.35

PRELIMINARY

PRELIMINARY

PRELIMINARY

Thanks for choosing this H.D.T. product.

www.hdtlovato.com

Read carefully this manual before using this product.

For continuous improvement, H.D.T. reserves the right to change features and specifications to manual and product without notice for the customer.

Details of images, contained here in, may be different from real product.

Doc.	Canopen - Ethercat
Rev. N°	4.0
Date	01/04/2021

Summary

PRELIMINARY

PRELIMINARY

PRELIMINARY

PRELIMINARY

PRELIMINARY

Ch. 1 Introduction

The purpose of this document is to describe the structure and parameters related to the CANopen protocol and etherCAT fieldbus, developed for the HDT's drives Tomcat, DGFOX and NTT. Detailed explanations about specific aspect of the CANopen protocol can be downloaded from the website

www.modbus.org

1.01 Configuration

The CANopen is a communication protocol based on a serial bus transmission system, where every node is assigned with a unique address, therefore in every CANopen network is necessary to set the nodes addresses and a communication speed equal for all the connected units.

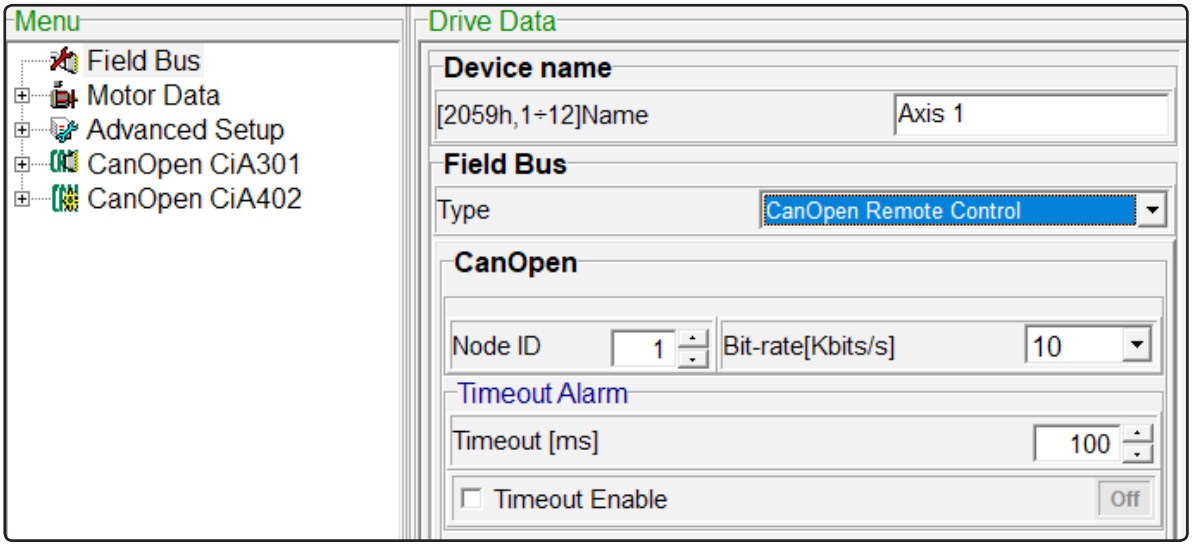
To enable the CANopen protocol is necessary to connect to the Drive with the software Caliper, which you can download from the HDT website (www.hdtlovato.com) after registering and logging in.

To select the CANopen protocol you have to follow this steps:

- Select "**Field Bus**" from the tree menu;
- Set the parameter "**Type**" on "**CanOpen Remote Control**";
- Set the "**Node-ID**" and "**Bit-rate**" parameters.
- If you want to activate the internal control of the communication timeout you have to enable the parameter "**Timeout Enable**" and set the value of the parameter "**Timeout**", which represents the maximum waiting time allowed between two telegrams. If no telegram is received within this time the alarm "**Error life guard**" will be activated and the function set on parameter "**Abort connection option code**" (index 6007h) will be performed.

PRELIMINARY

Bit rate	Cable length
1 Mbit/s	25 m
800 Kbit/s	50 m
500 Kbit/s	100 m
250 Kbit/s	250 m
125 Kbit/s	500 m
100 Kbit/s	700 m
50 Kbit/s	1000 m
20 Kbit/s	2500 m
10 Kbit/s	5000 m



Ch. 2 DS-301 protocol

The CanOpen is mainly based on the use of the variables contained in the ***Object Dictionary***; the access to these parameters is performed with various types of telegrams,described below.

2.01 SDO

SDO (***Service Data Objects***) are telegrams used for data exchange between master and slave (or client/server), where the master sends a request to the slave, and the slave answers with a confirmation containing any data required by the master. This data exchange is not performed in real time. ***SDO*** allow you to read and write all the parameters of the ***Object Dictionary***, regardless of their length or type. They're usually used for:

- Configuring the Drive system
- Reading error codes.

PRELIMINARY

2.02 PDO

PDO (***Process Data Objects***) are broadcast telegrams that do not require a confirmation response. They are used to periodically exchange process data in real time,like the speed reference and the measured speed of the motor. Two devices can use the PDO only to exchange "mapped" objects; in addition, before they start communicating it's necessary to define for both of them which parameters they will exchange. Depending on the transmission direction the PDOs can be of two different types:

- ***TPDO*** (***Trasmit-PDO***): Data telegram sent from the slave to the master
- ***RPDO*** (***Receive-PDO***): Data telegram sent from the master to the slave

2.03 SYNC

The ***SYNC*** is a telegram usually transmitted by the master, with the aim to provide a regular ***clock*** to synchronize the events (like the ***PDO*** transmission) within the network.

2.04 EMERGENCY OBJECT

The ***EMERGENCY OBJECT*** is a telegram which could be sent from any device to signal an emergency situation, the alarm code could be one foreseen by the standard, or it could represent a specific alarm code of the device.

2.04.1 Emergency object structure

The emergency object is a type of telegram composed by 8 bytes, which are described below:

Byte index	Description
0...1	***Emergency Error Code***, see the error code table below
2	***Error register*** (object on Index 1001h)
3...7	***Manufacturer specific Error Field:*** 4 of these 5 Bytes are used to report all the current alarms of the Drive; every single bit of the bytes 3,4,5, and 6 identify an alarm, when the alarm is present the corresponding bit is set to 1.

Bit	Alarm	Manufacturer Specific Error Field - Alarm bit description
0	FA_01	Error parameters storage
1	FA_02	Error offset currents
2	FA_03	Power overcurrent
3	FA_04	Over-voltage DC Bus
4	FA_05	No voltage
5	FA_06	phase missing
6	FA_07	Error position sensor of the motor
7	FA_08	Secure disable
8	FA_09	Over temperature motor
9	FA_10	Braking resistor (only for Tomcat and NTT)
10	FA_11	24 [V] absence
11	FA_12	Under voltage DC Bus
12	FA_13	Position error
13	FA_14	Error home position
14	FA_15	Warning I2t inverter
15	FA_16	Warning I2t motor
16	FA_17	Warning over speed
17	FA_18	Secure disable error 1
18	FA_19	(reserved)
19	FA_20	Fieldbus communication error
20	FA_21	Fault memory
21	FA_22	Motor phases Error
22	FA_23	Alarm Secure Disable
23	FA_24	Heatsink overtemperature
24	FA_25	Error sequence sensors hall motor
25	FA_26	Speed error
26	FA_27	Alarm I2t inverter
27	FA_28	Alarm I2t motor
28	FA_29	Alarm Simmetry DC Bus
29	FA_30	Alarm Overpressure
30	FA_31	Alarm Underpressure
31	-	reserved

PRELIMINARY

2.04.2 Automatic recovery from a DC BUS over-voltage

Depending on the parameter "***Alarm mode***" (index 200Fh) value, the Drive could respond in two different ways when an over-voltage occurs:

» ***Bit 0*** set to 0 (***stored***): with this mode the Drive remains in the "***Fault***" state even after the voltage returned below the maximum threshold allowed. To return on the "***Switch On Disable***" state you have to command a reset.

» ***Bit 0*** set to 1 (***auto reset***): when the voltage returns below the maximum threshold allowed the overvoltage alarm is automatically cancelled and the Drive returns on the "***Switch On Disable***" state.

2.04.3 Automatic recovery from a DC BUS under-voltage

Depending on the parameter "***Alarm mode***" (index 200Fh) value, the Drive could respond in two different ways when an over-voltage occurs:

» ***Bit 1*** set to 0 (***stored***): with this mode the Drive remains in the "***Fault***" state even after the voltage returned above the minimum threshold allowed. To return on the "***Switch On Disable***" state you have to command a reset.

» ***Bit 1*** set to 1 (***auto reset***): when the voltage returns above the minimum threshold allowed the undervoltage alarm is automatically cancelled and the Drive returns on the "***Switch On Disable***" state.

2.04.4 Warning I2t

When a "***Warning-I2t inverter***" or a "***Warning-I2t motor***" occur, an emergency telegram with the corresponding alarm code is sent, but the Drive doesn't go in "***Fault***" state. In this situation if the Drive is on "***Operation Enabled***" state the maximum torque current (Iq) is limited to avoid that the estimated temperature surpasses its maximum value allowed.

The Drive response in the "***Warning-I2t inverter***" state depends on the value set on the bit 2 of the parameter ***Alarm mode*** (index 200Fh).

PRELIMINARY

2.05 NETWORK MANAGEMENT OBJECTS

The ***NETWORK MANAGEMENT OBJECTS*** (***NMT***) are telegrams used for Network Management type services within Master-Slave structures; in practice the ***NMT*** structure handles the communication using 4 states (Initializing, Pre-operational, Operational, Stop), by which it enables and disables the transmission of certain types of telegrams and initialize the device.

Using the ***NMT*** telegrams the master set the slave state by sending a package made by 2 data: the first contains the ***command specifier*** (CS), the second contains the ***Node-ID***.

	INITIALISING	PRE-OPERATIONAL	OPERATIONAL	STOPPED
PDO			X	
SDO		X	X	
Synchronisation Object		X	X	
Time Stamp Object		X	X	
Emergency Object			X	
Boot-up Object	X			
Network Management Objects		X	X	X

Value	Comand Specifier (CS)	Description
001	Start Remote Node	Switch to the Operational state
002	Stop Remote Node	Switch to the Stopped state
128	Enter Pre-Operational-State	Switch to the Pre-Operational state
129	Reset Node	Node recovery
130	Reset Comunication	comunication recovery

2.06 Node Guarding protocol

The "***node guarding***" is a protocol used to verify the communication between the master and the slaves of the network. In this case the master has to send to the slave a request composed by the COB-ID= (1792+ Node-ID), which has to be done within the time set on the parameter "***Node life time***", otherwise the slave takes note of the missing communication and reacts accordingly.

After receiving the master request, the slave has to answer with a confirmation data package within the time set on "***Node life time***", otherwise the master takes note of the missing communication and reacts sending the error notification to the master application.

The "***node guarding***" protocol is activated if the parameters "***Guard time***" (index 100Ch) and "***Life time factor***" (index 100Dh) differs from 0 when the master send the first request "***node guarding***".

The parameter "***Node life time***" is obtained multiplying the "***Guard time***" parameter with the "***Life time factor***", and is expressed in [ms]. The Drive is prepared to handle the request telegrams coming from the master, in addition when a missing communication is reported the reaction set on parameter "***Abort connection option code***" (index 6007h) is started. The slave answer is composed by a byte where the bit 7 is used as a "***toggle***", while the other bits report the slave NMT state.

PRELIMINARY

2.07 Parameters of the DS-301 Communication Protocol

The CanOpen communication is handled by the drive using the parameters included between the indexes 1000h and 1FFFh of the ***Object Dictionary***, this range includes the communication profile setting common to all the devices connected to the network (standard DS-301). Communication data ca be viewed on PC with the Caliper software.

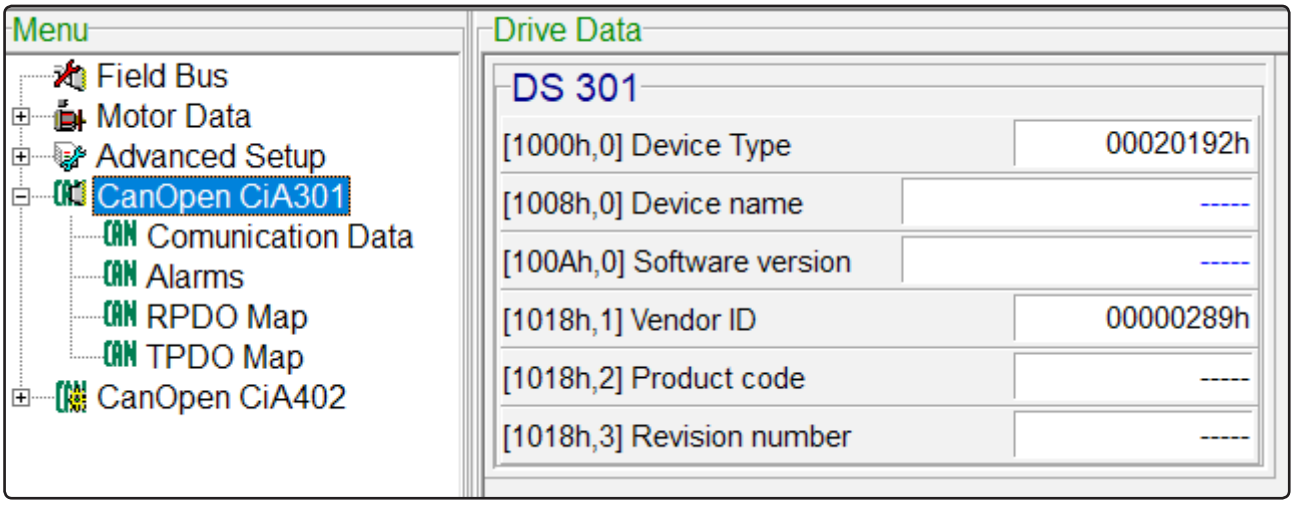
Code	Variable types
UINT8	UNSIGNED8
INT8	INTEGER8
UINT16	UNSIGNED16
INT16	INTEGER16
UINT32	UNSIGNED32
INT32	INTEGER32
STR	STRING

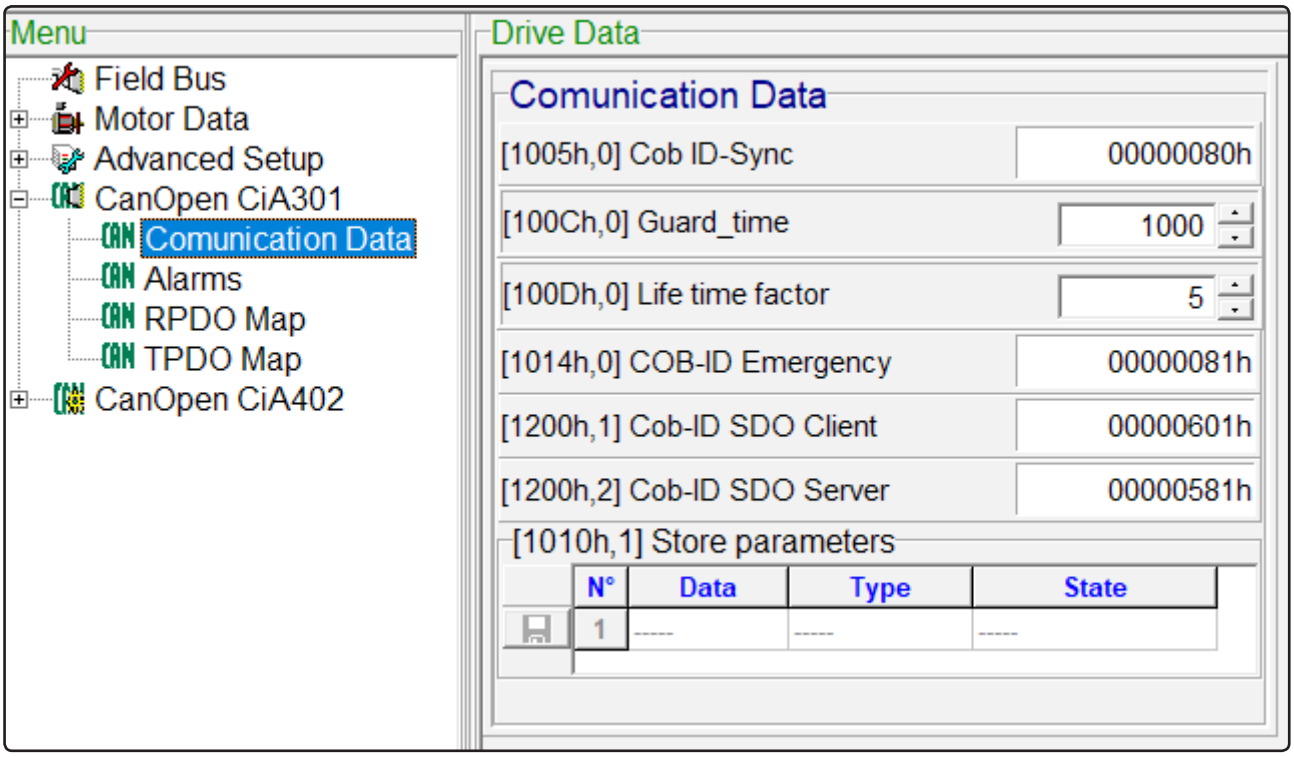
Attributes	Description
rw	readable and writable parameter (read, write)
w	writable parameter (write)
ro	readable parameter (read only)

2.08 Communication parameters

COMMUNICATION PARAMETERS							
Index (HEX)	**Sub - index**	**Object Dictionary parameter name**	**Type**	**Range values**	**Default**	**Map**	**Attribu-tes**
1000h		Device Type	UINT32	0...FFFFFFFFh	00020192h	No	ro
1001h		Error Register	UINT8	0..FFh	0	No	ro
1003h		Pre-defined Error Field	ARRAY				
	0	Number of error	UINT8	0..8	0	No	rw
	1..8	Standard error field	UINT32	0...FFFFFFFFh	0	No	ro
1008h		Manufacturer device name	STR	-		No	const
100Ah		Manufacturer software version	STR	-			
100Ch		Guard time	UINT16	0..FFFFh	100 [ms]	No	rw

COMMUNICATION PARAMETERS							
Index (HEX)	**Sub - index**	**Object Dictionary parameter name**	**Type**	**Range values**	**Default**	**Map**	**Attribu-tes**
100Dh		Life time factor	UINT8	0..FFh	2	No	rw
1010h		Store Parameters	ARRAY				
	0	Massimo numero sottoindici	UINT8	4	4	No	ro
	1..4	Modo salvataggio	UINT32	--	--	No	rw
1014h		COB-ID Emergency Object	UINT32	81h..FFh	80h+Node-ID	No	ro
1018h		Identity Object	RECORD				
	0	number of entries	UINT8	0..1	1	No	ro
	1	Vendor ID	UINT32	0...FFFFFFFFh	00000289h	No	ro
	2	Product Code	UINT32	0...FFFFFFFFh	00C0458h	No	ro
	3	Revision number	UINT32	0...FFFFFFFFh	-	No	ro





2.08.1 Communication parameters description

Index	Name	Description
1000h	Device type	• ***first field*** (higher 16 bits):it represents the device type, for the drive it's equal to 0002h (servo drive). • ***Second field*** (lower 16 bits): it represents the DS-402 device profile and it's equal to 0192h
1001h	Error register	***Error register***. As prescribed by the DS-301 CanOpen standard, the drive handles only the bit 0, which represents the occurrence of a generic error. For further information on the alarms you can refer to the following parameters: • ***Pre-defined Error Field*** (index 1003h) • ***Error Code*** (index 603Fh) • ***Emergency Object***
1003h	Pre-defined error Field	This parameter is composed by 9 fields: • ***sub index 0***: variable of one byte, it represents the number of memorized alarms, the maximum readable value is 8 and it corresponds with the maximum number of alarms which could be stored. To reset the stored alarms, you have to reset the sub index 0, as specified by DS-301 • ***sub index da 1 a 8***: these are 32 bits variables, used to store eventual alarms using numerical codes of 32 bits. To check the meaning of these alarm codes, see the "***Error codes***" table.
1008h	Manufacturer device name	Identifying string of the device
100Ah	Manufacturer software version	String containing the version of the Drive's software
100Ch	Guard time	This parameter is used with the "***life time factor***" to calculate the "***node life time***"
100Dh	Life time factor	This parameter is used with the "***guard time***" to calculate the "***node life time***"
1010h	Store parameters	Vector parameter used to save the Drive data in permanent memory on command
1014h	COB-ID Emergency Object	This variable stores the ***COB-ID*** for the emergency object, ***COB-ID***=80h+***Node-ID***
1018h	Identity Object	Record of parameters which contains general information of the device. The supported parameters are reported below:

Sub index	Name	Description
0	***number of entries***	Indica il numero di elementi che compongono la struttura
1	***Vendor ID***	Il valore numerico di questo campo corrisponde a 00000289h
2	***Product Code***	
3	***Revision number***	Numero revisione hardware

2.08.1.a Object 1010h - Store Parameters

Index	*Sub-Index*	*Parameter name*	*Type*	*Range*	*Default*	*Unit*	*Map*	*Attributes*
1010h		***Store parameters***	ARRAY					
	0	number of entries	UINT8	3	3		No	ro
	1	Save all parameters	UINT32	-	-		No	rw
	2	Save the DS301 parameters	UINT32	-	-		No	rw
	3	Save the DS402 parameters	UINT32	-	-		No	rw

PRELIMINARY

Using the "***store parameters***" you can store the Drive's parameters, depending on the type of the parameters that you want to save you have to write on one of the sub-index 1,2,3 or 4 the 32-bit data 65766173h (command of saving), this number converted to ASCII format corresponds to the string "***save***".

e	*v*	*a*	*s*
65h	76h	61h	73h

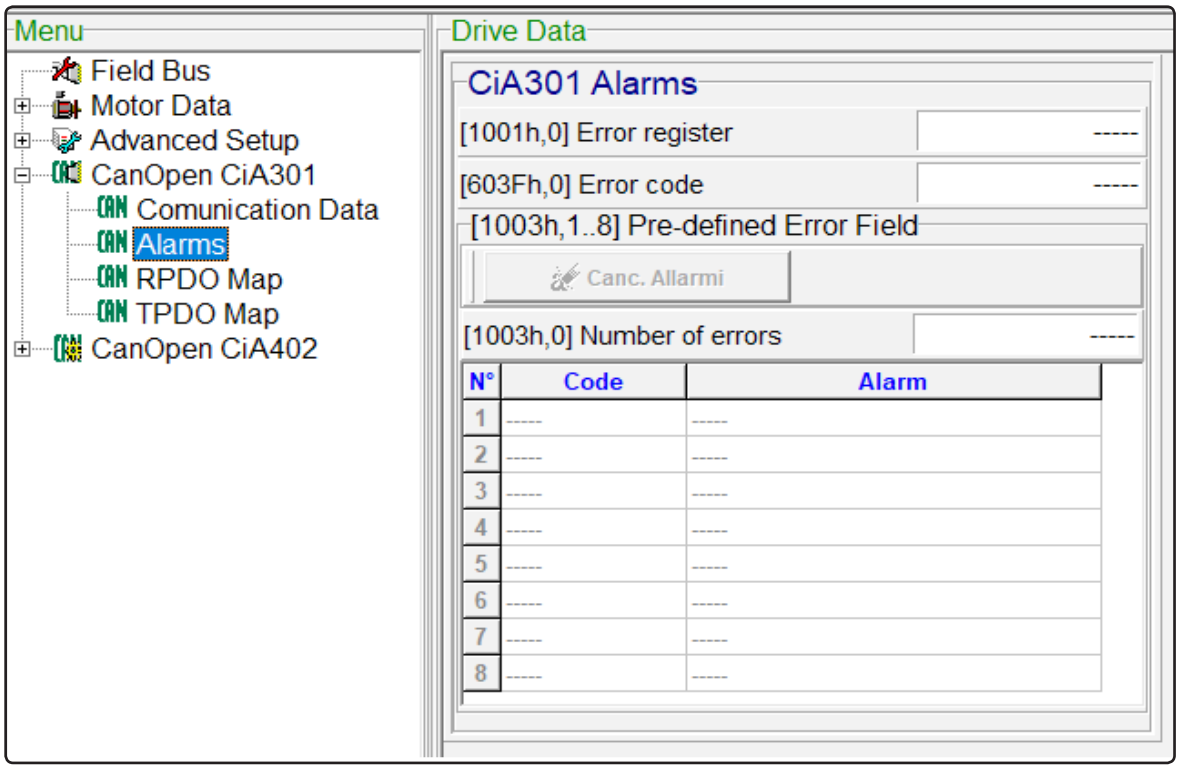
2.08.1.b Object 1005h - COB-ID sync

2.08.2 Error codes

The error codes are stored in a 16 bit variable and are reported in the first 2 bytes of the ***emergency*** telegrams.
Bold codes are drive specific errors (alarms).

PRELIMINARY

ERROR CODES (Alarms)		
Error code	**DS-301 description**	**Detailed description**
7600h	Data storage	Parameter memorization error
FF01h	***Manufacturer Alarm***	***Current offset error***
2340h	Short circuit	Power circuit overcurrent
3210h	DC link over-voltage	DC BUS overvoltage
FF0Bh	***Manufacturer Alarm***	***Hall sensors error***
FF08h	***Manufacturer Alarm***	***Secure Disable***
7120h	Allarme Motore	Motor overtemperature
7110h	Brake Chopper	Braking resistor (solo drive TomCat)
3220h	DC link Under Voltage	DC BUS undervoltage
8611h	Following Error	Position error
FF02h	***Manufacturer Alarm***	***Home Position error***
FF03h	***Manufacturer Alarm***	***Warning I2t Inverter***
FF04h	***Manufacturer Alarm***	***Warning I2t Motor***
FF05h	***Manufacturer Alarm***	***Warning overspeed***
FF09h	***Manufacturer Alarm***	***Secure Disable Error 1***
8130h	Life Guard Error	Life Guard Error
FF07h	***Manufacturer Alarm***	***Memory Fault***
FF0Ah	***Manufacturer Alarm***	***Motor phases error***
FF0Dh	***Manufacturer Alarm***	***Secure Disable Error 2***
4310h	Excess temperature drive	Radiator overtemperature
FF0Ch	***Manufacturer Alarm***	***Hall sensors sequency Error***
8400h	Velocity speed controller	Speed Error



2.09 Parameters SDO

PARAMETERS SDO (Server)							
Index	**Sub index**	**Object Dictionary parameter name**	**Type**	**Range values**	**Default**	**Map**	**Attributes**
1200h		Server SDO Parameter	RECORD				
	0	number of entries	UINT8	2	2	No	ro
	1	COB-ID Client -> Server (rx)	UINT32	601h..67Fh	600h+Node-ID	No	ro
	2	COB-ID Server -> Client (rx)	UINT32	581h..5FFh	580h+Node-ID	No	ro

2.09.1 SDO parameters description

The index 1200h contains a record reporting the ***COB-ID*** of the server receiving the ***SDO*** telegrams and the ***COB-ID*** of the server transmitting the ***SDO*** telegrams.

2.10 RPDO parameters

RPDO PARAMETERS (Reception)							
Index	**Sub index**	**Object Dictionary parameter name**	**Type**	**Range Values**	**Default**	**Map**	**Attri-butes**
1400h		***Receive PDO parameter 1***	RECORD				
	0	largest sub-index supported	UINT8	2	2	No	ro
	1	COB-ID used by PDO	UINT32	--	200h + Node-ID	No	rw
	2	Transmission type	UINT8	0..FFh	0	No	rw
1401h		***Receive PDO parameter 2***	RECORD				
	0	largest sub-index supported	UINT8	2	2	No	ro
	1	COB-ID used by PDO	UINT32	--	300h + Node-ID	No	rw
	2	Transmission type	UINT8	0..FFh	0	No	rw

PRELIMINARY

RPDO PARAMETERS (Reception)							
Index	**Sub index**	**Object Dictionary parameter name**	**Type**	**Range Values**	**Default**	**Map**	**Attri-butes**
1402h		***Receive PDO parameter3***	RECORD				
	0	largest sub-index supported	UINT8	2	2	No	ro
	1	COB-ID used by PDO	UINT32	--	400h + Node-ID	No	rw
	2	Transmission type	UINT8	0..FFh	0	No	rw
1403h		***Receive PDO parameter4***	RECORD				
	0	largest sub-index supported	UINT8	0..2	2	No	ro
	1	COB-ID used by PDO	UINT32	--	500h + Node-ID	No	rw
	2	Transmission type	UINT8	0..FFh	0	No	rw
1600h		***Receive PDO mapping 1***					
	0	number of mapped PDO	UINT8	0..8	--	No	rw
	1..8	PDO mapping	UINT32	--	--	No	rw
1601h		***Receive PDO mapping 2***					
	0	number of mapped PDO	UINT8	0..8	--	No	rw
	1..8	PDO mapping	UINT32	--	--	No	rw
1602h		***Receive PDO mapping 3***					
	0	number of mapped PDO	UINT8	0..8	--	No	rw
	1..8	PDO mapping	UINT32	--	--	No	rw
1603h		***Receive PDO mapping 4***					
	0	number of mapped PDO	UINT8	0..8	--	No	rw
	1..8	PDO mapping	UINT32	--	--	No	rw

PRELIMINARY

2.10.1 RPDO parameters description

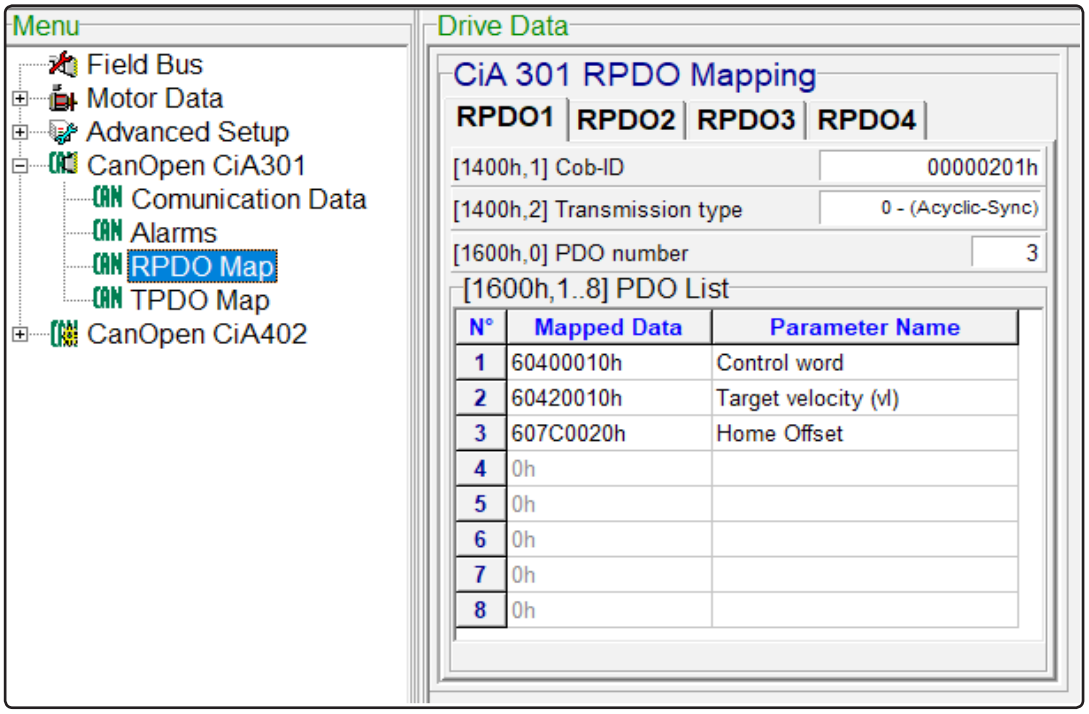
Index	Parameter name	Description
1400h.. 1403h	Receive PDO parameter	These records contain the reception COB-ID of the RPDO telegrams and the parameters used to set the reception type

The sub-Index 1 of these parameters indicates the ***COB-ID*** of the receiving ***PDO***, while the sub-index 2 indicates the type of the settable ***RPDO*** transmission, based on the latter value the following types of communication are possible:

Transmission type	Description
0	***Asynchronous cyclic communication***: the data are sent to the master only when they change, when the "***sync***" is received these values are updated on the Drive parameters
1..240	***Synchronous cyclic communication***: the value set on "***Transmission type***" represents the number of "***sync***" that have to be received before the master sends the data, the received data is updated on the drive's parameter when the "***sync***" is received.
241..251	reserved values
252..253	These options are available only for the TPDO because they're handled with RTR telegrams (Remote Transmission Request).
254	***Asynchronous communication (manufacturer specific)***: this option is reserved for drive specific communications, with this option the Drive "behaves" as with option 255.
255	***Asynchronous communication (device profile specific)***: the data received using the RPDO are immediately updated on the Drive's parameters, without waiting for the "***sync***".

2.10.2 RPDO mapping parameters description

Index	Parameter name	Description
1600h..1603h	Receive PDO mapping	Parameters record used to store the received PDO content, dynamic mapping is allowed for the Drive



PRELIMINARY

2.11 TPDO parameters

TPDO parameters (Trasmission)							
Index	**Sub index**	**Object Dictionary parameter name**	**Type**	**Range Values**	**Default**	**Map**	**Attribu-tes**
1800h		**Transmit PDO parameter 1**	RECORD				
	0	largest sub-index supported	UINT8	2	2	No	ro
	1	COB-ID used by PDO	UINT32	--	180h+ Node-ID	No	rw
	2	Transmission type	UINT8	0..FFh	0	No	rw
1801h		**Transmit PDO parameter 2**	RECORD				
	0	largest sub-index supported	UINT8	2	2	No	ro
	1	COB-ID used by PDO	UINT32	--	280h + Node-ID	No	rw
	2	Transmission type	UINT8	0..FFh	0	No	rw
1802h		**Transmit PDO parameter3**	RECORD				
	0	largest sub-index supported	UINT8	2	2	No	ro
	1	COB-ID used by PDO	UINT32	--	380h + Node-ID	No	rw
	2	Transmission type	UINT8	0..FFh	0	No	rw
1803h		**Transmit PDO parameter4**	RECORD				
	0	largest sub-index supported	UINT8	2	2	No	ro
	1	COB-ID used by PDO	UINT32	--	480h + Node-ID	No	rw
	2	Transmission type	UINT8	0..FFh	0	No	rw
1A00h		**Transmit PDO mapping 1**					
	0	number of mapped PDO	UINT8	0..8	--	No	rw
	1..8	PDO mapping	UINT32	--	--	No	rw

TPDO parameters (Trasmission)							
Index	**Sub index**	**Object Dictionary parameter name**	**Type**	**Range Values**	**Default**	**Map**	**Attribu-tes**
1A01h		**Transmit PDO mapping 2**					
	0	number of mapped PDO	UINT8	0..8	--	No	rw
	1..8	PDO mapping	UINT32	--	--	No	rw
1A02h		**Transmit PDO mapping 3**					
	0	number of mapped PDO	UINT8	0..8	--	No	rw
	1..8	PDO mapping	UINT32	--	--	No	rw
1A03h		**Transmit PDO mapping 4**					
	0	number of mapped PDO	UINT8	0..8	--	No	rw
	1..8	PDO mapping	UINT32	--	--	No	Rw

PRELIMINARY

2.11.1 TPDO parameters description

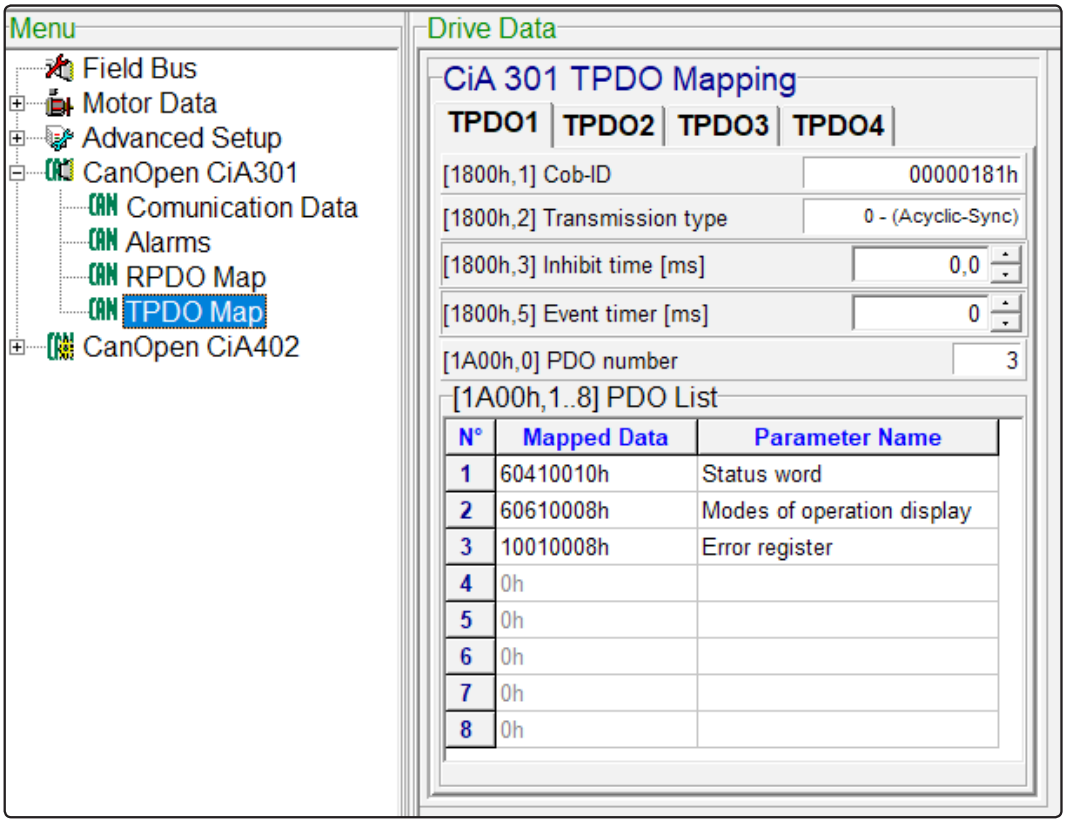
Index	Parameter name	Description
1800h..1803h	Transmit PDO mapping	These records contain the transmission ***COB-ID*** of the ***PDO*** telegrams and the parameters used to set this type of transmission

Sub-Index 1 indicates the ***COB-ID*** of the transmission ***PDO***, while sub-index 2 indicates the transmission type among the ones supported by the ***TPDO***, depending on the latter's value you can have this type of communication:

Transmission type	Description
0	***Asynchronous cyclic communication***: the mapped data are sent when the ***sync*** is received but only if they've changed.
1..240	***Synchronous cyclic communication***: with this option the data are cyclically sent in accordance with the "***sync***", the value set on "***Transmission type***" represents the number of "***sync***" that have to be received before the data is sent.
241..251	Reserved values
252	***Synchronous RTR Communication***: the data is sent when the "***sync***" is received, after receiving an RTR telegram (*Remote Transmission Request*) with the same ***COB-ID*** of the ***TPDO*** to send.
253	***Asynchronous RTR Communication***: the data is sent after receiving an RTR telegram (*Remote Transmission Request*) with the same ***COB-ID*** of the ***TPDO*** to send.
254	***Asynchronous communication (manufacturer specific)***: this option is reserved for drive specific communications, with this option the Drive "behaves" as with option 255.
255	***Asynchronous communication (device profile specific)***: communication handled according with the asynchronous events, in the Drive the mapped data is sent when an RTR telegram is received, like in option 253.

2.11.2 TPDO mapping parameters description

Index	Parameter name	Description
1A00h..1A03h	Transmit PDO mapping	Record of parameters used to set the content of the transmitting ***PDO***, in the drive the ***dynamic mapping*** is allowed

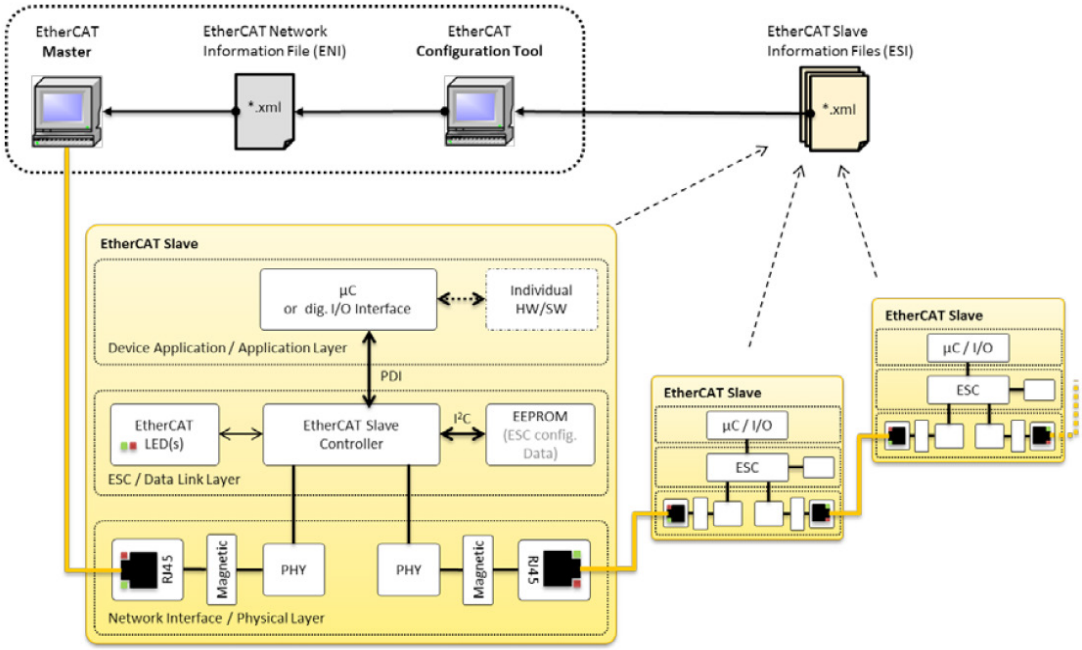


Ch. 3 EtherCAT

All the HDT Drives (NTT, DGFOX and TOMCAT) support the CAN bus and the EtherCAT (Ethernet Control Automation Technology) Fieldbus, and both solutions allows you to use the CANopen protocol. EtherCAT allows you to use the CANopen on Ethernet, which could become a valid alternative in order to overcome certain limitations of the CAN bus (es. the network length or the number of nodes,ecc). The SDO and PDO protocols and the device profiles of the CANopen described in this manual are used in EtherCAT without any changes.

3.01 Introduction

The figure below shows the EtherCAT network with focus on the slave architecture. Basically, the slave contains three main components : the ***Physical layer***, the ***Data link layer*** and the ***Application layer***.



The ***Physical layer*** contains all the standard ethernet components necessary to process fieldbus signals. It acts as an interface between the ***etherCAT slave controller*** (***ESC***) of a Slave and the rest of the network: it forwards network data to the ***ESC*** and applies signals from the ***ESC*** to the network.

The ***Data link layer*** is formed by two elements: the ***EtherCAT slave controller*** (***ESC***) and the ***EEPROM***.

The ***ESC*** is a chip for EtherCAT communication,which handles the EtherCAT protocol in real-time by processing frames on the fly. The data read by the ***ESC*** are then sent to the Slave microcontroller via the ***Process Data Interface*** (***PDI***). The ***PDI*** exchanges data in both direction, so the microcontroller can use it to send to send data to ***ESC***, which will convert it and send it to the Network Master.

The ***EEPROM***, also called ***Slave Information Interface*** (***SII***), contains all the Slave's information that must be stored permanently, regarding its identity and functionality. It contains also hardware configuration information for the ***ESC*** which is loaded to the ***ESC***'s registers during power-up. This preliminary configuration of the ***ESC*** registers is necessary in order to make the ***PDI*** work correctly. The EEPROM can be written by the configuration tool (via EtherCAT) based on the ESI file.

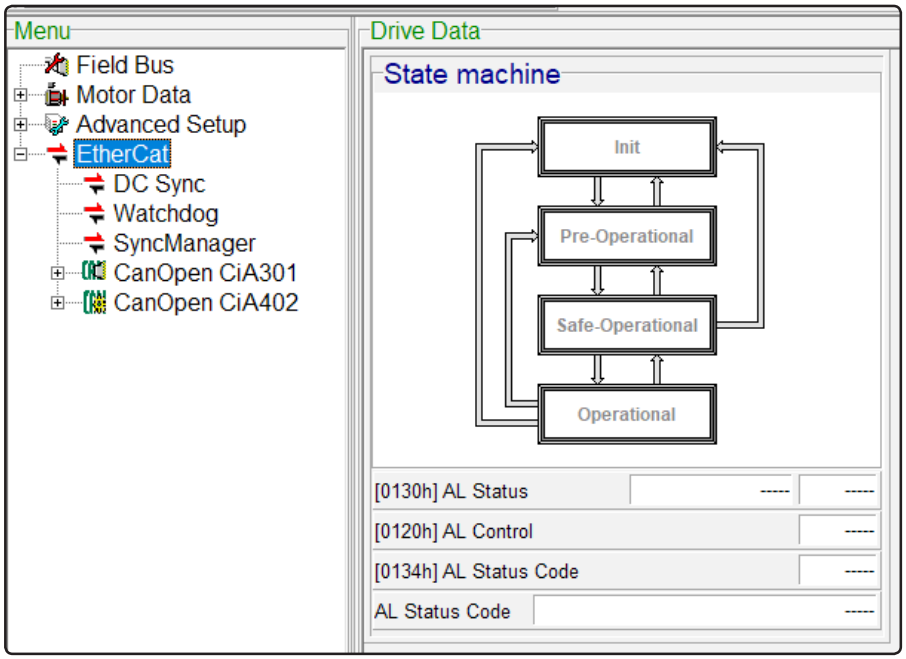
The ***Application layer*** includes all the specific software (i.e. communication software, specific applications) of the Slave, implemented in the microcontroller. The microcontroller has to handles also the following :

- the ***EtherCAT State Machine*** (***ESM***) of the slave device
- Process data exchange with the slave application (e.g. application and configuration parameters, object dictionary (***SDO***))
- Mailbox based protocol (***CoE***- ***CANopen over etherCAT***) for acyclic data exchange

3.02 EtherCAT State Machine

All the Slaves (Drives) connected to an EtherCAT network have to run the state machine described below to indicate their current state. Depending on its state, a Slave may or may not perform certain functions.

This state machine and its registers can be seen in the menu "***EtherCat***" in which the following screen appears:



This state machine does not replace the one described on paragraph *"4.01 State Machine" pag. 38*, they run together. it only refers to the state of the communication between the Drive and the ethercat network.

The state machine is controlled and monitored via 3 registers: the ***AL control*** register [0120h], the ***AL Status*** register [0130h] and the ***AL Status Code*** register [0134h].The Master can order a Slave to change its state by writing a request on its ***AL control*** register. If the current conditions allow to move to the requested state, the Slave acknowledges the new state and writes a confirmation answer on the ***AL Status*** register; if not, the Slave sets the error flag in the ***AL Status*** register and writes an error code to the ***AL Status Code*** register.

AL Control [0x0120h]				
Bit	***Description***	***ECAT***	***PDI***	***Default***
3:0	Initiate State Transition of the Device State Machine: 1: Request ***Init*** State 2: Request ***Pre-Operational*** State 4: Request ***Safe-Operational*** State 8: Request ***Operational*** State	Rw	Ro	1
4	Error Index Acknowledgment: 0: "***Error Index***" in ***AL Status*** register = 0 1: "***Error Index***" in ***AL Status*** register = 1	Rw	Ro	0
15:5	Reserved, write 0	Rw	Ro	0

AL Status Code [0x0134]				
Bit	***Description***	***ECAT***	***PDI***	***Default***
15:0	AL Status Code	Ro	Rw	0

AL Status [0x0130]				
Bit	**Description**	**ECAT**	**PDI**	**Default**
3:0	Actual State of the Device State Machine: 1: ***Init*** State 2: ***Pre-Operational*** State 4: ***Safe-Operational*** State 8: ***Operational*** State	Ro	Rw	1
4	Error Index: 0: Device is in State as requested or Flag cleared by command 1: Device has not entered requested State or changed State as result of a local action	Ro	Rw	0
15:5	Reserved, write 0	Ro	Rw	0

The states of this state machine and the transition phases are described in the following table:

Status	Description
Init	After switch-on the EtherCAT slave is in the ***Initial state***. No mailbox or process data communication is possible.
Init to Pre-Op	During the transition between ***Init*** and ***Pre-Op*** the Master configures the ***data link layer*** addresses and initiates the mailbox communication. The ***DC clocks*** synchronization is initialized, and a request to change into ***Pre-op*** status is set on the ***AL control register***. The slave checks whether the mailbox was initialized correctly.
Pre-Op	In ***Pre-Op*** state mailbox communication is possible, but not process data communication.
Pre-Op to Safe-Op	The EtherCAT master initializes the ***SyncManager*** channels for process data, the FMMU channels and the PDO mapping of the ***SyncManager*** assignment. After that the master sends a request for the ***Safe-Operational*** state. The Slave checks whether the ***SyncManager*** channels for process data communication and the distributed clocks settings are correct. Before it acknowledges the change of state, the EtherCAT slave copies current input data into the associated DP-RAM areas of the ***ESC***.
Safe-Op	In ***Safe-Op*** state mailbox and process data communication is possible, although the slave keeps its outputs in a safe state (i.e. it's not executed), while the input data are updated cyclically
Safe-Op to Op	The Master transmits effective output data, and it asks the Slave to change into ***OP*** status.
Op	In the ***Op*** state the slave copies the output data of the Master to its outputs. Process data and mailbox communication is possible (both in input and in output).

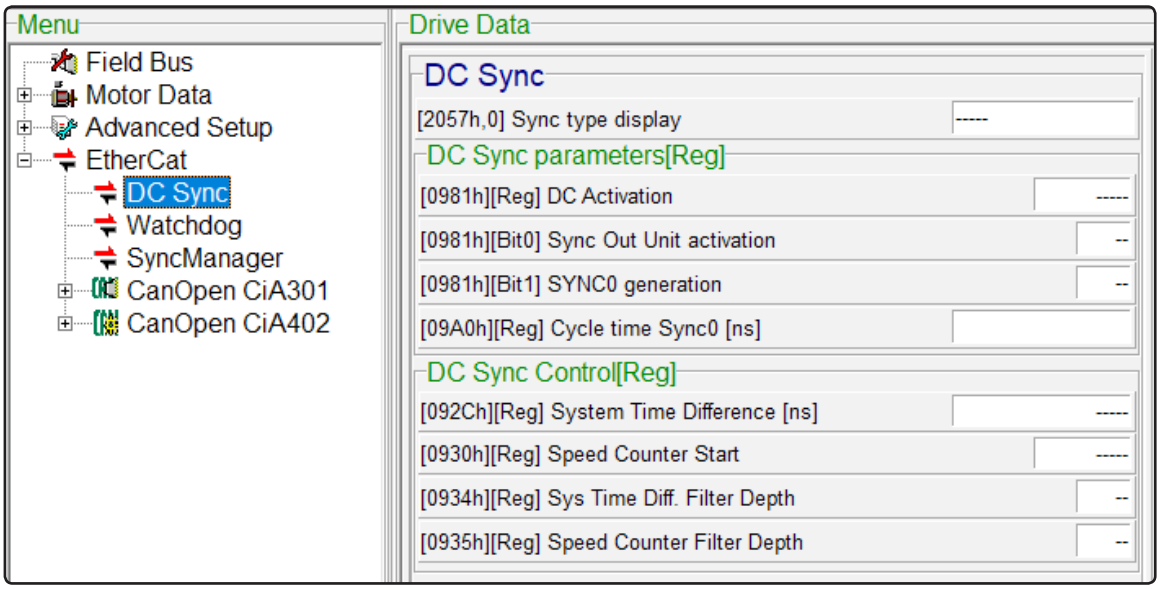
NOTE: Not all state changes are possible, e.g., the transition from ***Init*** to ***Operational*** requires the following sequence: ***Init*** -> ***Pre-Operational*** -> ***Safe-Operational*** -> ***Operational***.

3.03 DC Synchronization

In most real-world applications there are multiple servo drives connected to the same etherCAT network that must work together, for example by making some axes move in sync to perform a certain task. But in order to achieve high synchronization precision all the Drives clocks must share the same time (under a certain margin). The method of ***Distributed Clocks*** (***DC***) is used for this very reason, i.e. to provide highly precise time synchronization between slaves. This method is also necessary for:

- Generation of synchronous output signals (SyncSignals)
- Precise time stamping of input events (LatchSignals)
- Generation of synchronous interrupts
- Synchronous Digital Output updates
- Synchronous Digital Input sampling

The registers used by the ***DC clocks*** can be seen in the menu "***EtherCat-DC Sync***" in which the following screen appears:



PRELIMINARY

3.03.1 Functioning of the DC clocks

To synchronize an EtherCAT network one of the connected device will be used as a ***Reference Clock***, typically the ***ESC*** of the Slave closer to the Master. The ***Reference Clock*** provides the ***System Time***. The clocks of the other Drives have to report the same value of the ***System Time*** within a certain period. These clocks initially differs from the ***System time*** by a certain error composed by two terms, the ***offset*** and the ***drift***:

- the ***offset*** is a constant error due to the propagation delay of the network and the initial differences of the local times resulting from different times at which the ***ESCs*** have been powered up.
- the error ***drift*** term between a local clock and the reference clock occurs when they run at different speed, which is the most common case since they are not typically connected to the same clock source (e.g. a quartz). if the drift is not compensated ,the error will increase over time.

To eliminate these errors and synchronize all the clocks the following steps are performed:

1. Propagation Delay Measurement
2. ***Offset*** compensation to ***Reference Clock*** (***System Time***)
3. ***Drift*** compensation to ***Reference Clock***

The network propagation delay is due to the cables that compose it and to the time required by every Drive to elaborate a frame and forward it to the next Drive. In order to measure this delay the Master sends a broadcast frame to all the network, with the order to write the moment when it's received. All the Slaves follow this order when the frame arrives to one of their ports, and if they have more ports they write the time of arrival for each of them. These time stamps are then read by the Master, which will use them to calculate how much time is necessary to a frame to come and go from each port of the Slaves. Using this information and knowing the topology of the Network the Master derives the propagation delay of the network. It's necessary to compensate this delay to make all the Drives actually work in sync.

To compensate this delay, the initial ***offset*** and the ***Drift*** for every Clock is theoretically only necessary a closed loop control called ***Time Control Loop*** (see paragraph *"3.03.2 Time Control Loop" pag. 27*), present on every ***ESC***. This loop however is slow, so if the offset is too high it would take too long to eliminate it. For this very reason the Master use another correction, rough but faster, to eliminate the initial offset. the Master calculates the difference between the ***System Time*** and the Local Time for every Drive, then saves these values on the register ***System Time Offset*** of each of them. The Drives will add this difference to their local time to get their "local copy" of the ***System Time***:

Local copy of System Time = Local time + time Offset

After the first rough compensation of the ***offset***, the ***Time Control Loop*** makes a more precise correction to compensate the network propagation delay and the ***Drift*** effect. The master distributes the ***System Time*** of the ***Reference Clock*** to all Slave clocks periodically. The ***Time control loop*** of each slave takes the lower 32 bit of the ***System Time*** received from the ***Reference Clock*** and compares it to its local copy of the ***System Time***. For this difference, the propagation delay has to be taken into account:

$$\Delta t = (Local\ time + time\ Offset - Propagation\ delay) - Received\ System\ Time$$

If Δt is positive, the local time is running faster than the ***System Time***, and has to be slowed down. If Δt is negative, the local time is running slower than the ***System Time***, and has to be sped up. Now the local copy of the ***System Time***, adjusted with this control, can be used to generate interrupts and be passed to the microcontroller.

3.03.2 Time Control Loop

The time control loop has some configuration and status registers .The default settings of these registers are sufficient for proper operation of the drift compensation.

System time difference [0x92Ch:0x92Fh]				
Bit	***Description***	***ECAT***	***PDI***	***Default***
30:0	Mean difference between local copy of ***System Time*** and received ***System Time*** values	Ro	Ro	0
31	0) Local copy of ***System Time*** greater than or equal received ***System Time*** 1) Local copy of ***System Time*** smaller than Received ***System Time***	Ro	Ro	0

The ***System Time Difference*** register (0x092C:0x092F) contains the mean value of the difference between local copy of the ***System Time*** and the received value of ***System Time*** (Δt). This value converges to zero when both times are identical. The clocks synchronization is established when the value of this register drops below a certain threshold.

Speed Counter Start [0x930h:0x931h]				
Bit	***Description***	***ECAT***	***PDI***	***Default***
14:0	Bandwidth for adjustment of local copy of ***System Time*** (larger values -> smaller bandwidth and smoother adjustment) A write access resets ***System Time Difference*** (0x092C:0x092F) and ***Speed Counter Diff*** (0x0932:0x0933). Minimum value: 0x0080	Rw	Rw	0x1000
15	Reserved	Ro	Ro	0

The ***Speed Counter Start*** register (0x0930:0x0931) represents the bandwidth of the ***Time Control Loop***.

System Time Difference Filter Depth [0x934h]				
Bit	**Description**	**ECAT**	**PDI**	**Default**
3:0	Filter depth for averaging the received ***System Time*** deviation .A write access resets ***System Time Difference*** (0x092C:0x092F)	Rw	Rw	4
7:4	Reserved	Ro	Ro	0

The ***System Time Difference Filter Depth*** register (0x0934) sets filter depths for mean value calculation of the received ***System Time***'s deviation.

Speed Counter Filter Depth [0x935h]				
Bit	**Description**	**ECAT**	**PDI**	**Default**
3:0	Filter depth for averaging the clock period deviation. A write access resets the internal ***speed counter filter***.	Rw	Rw	12
7:4	Reserved	Ro	Ro	0

The ***Speed Counter Filter Depth*** register (0x0935) set filter depths for mean value calculation of the calculated clock's period deviation.

PRELIMINARY

3.03.3 SyncSignals

The ***DC Cyclic Unit / Sync Unit*** supports the generation of two ***SyncSignals*** (Synchronized signals), SYNC0 and SYNC1. The ***SyncSignals*** can both be used internally and externally of the ***ESC***. ***SyncSignals*** can be generated at a specific ***System Time***. Four operation modes are supported: ***cyclic generation***, ***single shot***, ***cyclic acknowledge***, and ***single shot acknowledge*** mode. The acknowledged modes are typically used for interrupt generation. The interrupts have to be acknowledged by the microcontroller. The cycle time of the SYNC0 signal is configured in the ***SYNC0 Cycle Time register*** (0x09A0:0x09A3), the start time is set in the ***Start Time Cyclic Operation register*** (0x0990:0x0997). After the ***Sync Unit*** is activated and the output of the SYNC0/1 signals is enabled (***DC Activation register*** 0x0981), the ***Sync Unit*** waits until the start time is reached and generates the first SYNC0 pulse. Some ***ESC***s support additional activation options like auto-activation when the Start Time is written, or 64 bit extension if only 32 bit of the ***Start Time*** is written. Other options are to detect invalid Start Times and provide debug output of ***SyncSignals***. Internally, the ***SyncSignals*** are generated with an update rate of 100 [MHz] (10 [ns] update cycle). The jitter of the internal ***SyncSignal*** generation in comparison to the ***System Time*** is 12 [ns].

Cyclic Generation

In ***Cyclic Generation*** mode, the ***Sync unit*** generates isochronous ***SyncSignals*** after the Start Time. The generation ends if the Cyclic Unit is deactivated or SYNC0/1 generation is deactivated. The Cycle times are determined by the ***SYNC0/1 Cycle Time registers***. The Pulse Length of the SYNC signals has to be greater than 0. If the Pulse Length is greater than the Cycle Time, the SyncSignal will always be activated after the Start Time.

Single Shot Mode

In ***Single Shot*** mode (***SYNC0 Cycle Time*** set to 0), only one ***SyncSignal*** pulse is generated after the Start Time is reached. Another pulse can only be generated by deactivating the Cyclic Unit (i.e putting bit 0 of 0x0981 to 0), reprogramming the Start Time, and reactivating the Cyclic Unit.

Cyclic Acknowledge Mode

The ***Cyclic Acknowledge*** mode is typically used for generation of isochronous interrupts. The acknowledged modes are selected by setting the ***Pulse Length of SYNC Signals*** to 0 (0x0982:0x0983). Each SyncSignal pulse remains active until it is acknowledged by a microcontroller by reading the appropriate ***SYNC0/SYNC1 Status register*** (0x098E, 0x098F). The first pulse is generated after the Start Time is reached, following pulses are generated when the next regular SYNC0/1 event would occur.

Single Shot Acknowledge Mode

In ***Single Shot Acknowledge*** mode (both ***Pulse Length of SYNC Signals*** and ***SYNC0 Cycle Time*** are 0), only one pulse is generated when the Start Time is reached. The pulse remains active until it is acknowledged by reading the appropriate SYNC0/1 Status registers. Another pulse can only be generated by deactivating the Cyclic Unit (i.e putting bit 0 of 0x0981 to 0), reprogramming the Start Time, and reactivating the Cyclic Unit.

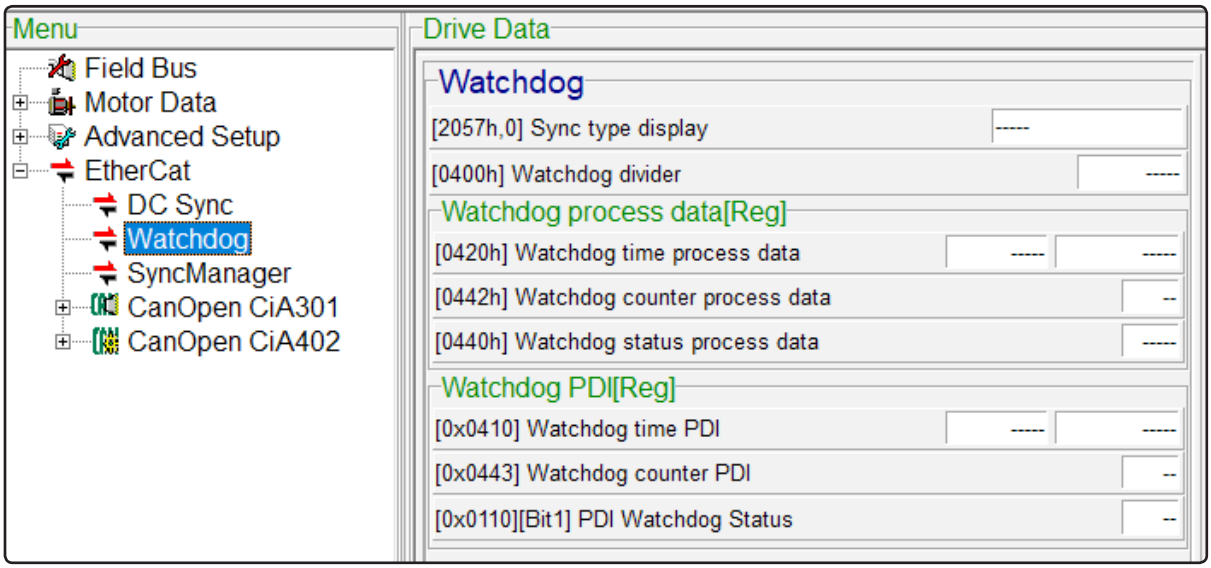
DC Activation register [0x981h]				
Bit	***Description***	***ECAT***	***PDI***	***Default***
0	Sync Out Unit activation: 0) Deactivated 1) Activated Write 1 after Start Time was written	Rw	Rw	0
1	Sync0 Generation: 0) Deactivated 1) Sync0 pulse is generated	Rw	Rw	0

Sync0 Cycle time [0x9A0h:0x09A3h]				
Bit	***Description***	***ECAT***	***PDI***	***Default***
31:0	Time between two consecutive Sync0 pulses in [ns]. 0) Single shot mode, generate only one Sync0 pulse	Rw	Rw	0

3.04 Watchdog

The ESCs support up to two internal watchdogs, a ***Process Data watchdog*** and a ***PDI watchdog***. The ***Process Data watchdog*** is reset after each successful EtherCAT process data communication with the terminal. If no EtherCAT process data communication takes place with the terminal for longer than the threshold time set on this watchdog, e.g. in the event of a line interruption, the watchdog is triggered and the outputs are set to FALSE. The watchdog is only reset after a successful EtherCAT process data access.

The ***PDI watchdog*** instead monitors the communication between the ESC and the microcontroller. If no PDI communication with the ESC takes place for longer than the PDI watchdog threshold time, this watchdog is triggered. These watchdogs and their registers can be seen in the menu "***EtherCat-Watchdog***" in which the following screen appears:



The timeout for both watchdogs can be configured individually, but they share a single ***Watchdog Divider*** (register 0x0400:0x0401): this register represents the basic watchdog increment. The watchdog timeout is calculated from the ***Watchdog Divider*** settings multiplied with the ***Watchdog Time PDI*** (register 0x0410:0x0411) or ***Watchdog Time Process Data*** (register 0x0420:0x0421).

Watchdog Divider [0x0400:0x0401]				
Bit	***Description***	***ECAT***	***PDI***	***Default***
15:0	Number of 25 MHz tics (minus 2) that represents the basic watchdog increment. (Default value is 100µs = 2498)	Rw	Ro	0x09C2

Watchdog Time PDI [0x0410:0x0411]				
Bit	***Description***	***ECAT***	***PDI***	***Default***
15:0	Number of basic watchdog increments (Default value with Watchdog divider 100µs means 100ms Watchdog)	Rw	Ro	0x03E8

Watchdog Time Process Data [0x0420:0x0421]				
Bit	***Description***	***ECAT***	***PDI***	***Default***
15:0	Number of basic watchdog increments (Default value with Watchdog divider 100µs means 100ms Watchdog)	Rw	Ro	0x03E8

PRELIMINARY

The ***Watchdog divider***'s time is a multiple of 40 [ns]. The Watchdog timeout jitter depends on the ***Watchdog Divider*** settings. I.e. selecting smaller ***Watchdog Divider*** settings results in smaller jitter.

The following equations are used for a quick estimation of the watchdog timeout (they are not exact in terms of nanoseconds):

*tWD_Div = (WD_DIV + 2) * 40ns*

*tWD_PDI = between (tWD_Div * WD_PDI) and (tWD_Div * WD_PDI + tWD_Div)*

*tWD_PD = between (tWD_Div * WD_PD) and (tWD_Div * WD_PD + tWD_Div)*

where:

- *WD_DIV* is the ***Watchdog Divider*** (register 0x0400:0x0401)
- *WD_PDI* is the ***Watchdog Time PDI*** (register 0x0410:0x0411)
- *WD_PD* is the ***Watchdog Time Process Data*** (register 0x0420:0x0421)
- *tWD_PDI* is the timeout of the ***PDI watchdog***
- *tWD_PD* is the timeout of the ***Process Data watchdog***

The ***Process Data watchdog*** is rewound (triggered) by a write access to a ***SyncManager*** buffer area, if the ***SyncManager*** is configured to generate a watchdog trigger signal (***SyncManager Control register*** 0x0804.6 for ***SyncManager 0***, etc.). The watchdog trigger signal is generated after the buffer was completely and successfully written (similar to the Interrupt Write of a ***SyncManager***).

The ***Process Data watchdog*** can be disabled by setting the ***Process Data Watchdog Time*** to 0.

A timeout of the ***Process Data watchdog*** has these consequences:

- ***Watchdog Status register for Process Data*** (0x0440.0) reflects the watchdog status.
- The Digital I/O PDI takes back digital output data, either by not driving the signals anymore or by driving them low (ESC and configuration dependent).
- The Watchdog Counter Process Data (0x0442) is incremented.

Watchdog Status Process Data [0x0440:0x0441]				
Bit	**Description**	**ECAT**	**PDI**	**Default**
0	Watchdog Status of Process Data (triggered by SyncManagers) 0: Watchdog Process Data expired 1: Watchdog Process Data is active or disabled	Ro	Ro	0
15:1	Reserved	Ro	Ro	0

Watchdog Counter Process Data [0x0442]				
Bit	**Description**	**ECAT**	**PDI**	**Default**
7:0	Watchdog Counter Process Data (counting is stopped when 0xFF is reached). Counts if Process Data Watchdog expires. Cleared if one of the Watchdog counters 0x0442:0x0443 is written.	Rw	Ro	0

The ***PDI watchdog*** is rewound (triggered) by any correct read or write access by the PDI. The ***PDI watchdog*** can be disabled by setting the ***PDI Watchdog Time*** to 0. A timeout of the ***PDI watchdog*** has these consequences:

- ***ESC DL Status register*** (0x0110.1) reflects the watchdog status. This can be mapped to the ECAT Interrupt to inform the master.
- The ***Watchdog Counter PDI*** (0x0443) is incremented.

NOTE: The Digital I/O PDI only triggers the PDI watchdog upon input events.

Watchdog Counter PDI [0x0443]				
Bit	**Description**	**ECAT**	**PDI**	**Default**
7:0	Watchdog PDI counter (counting is stopped when 0xFF is reached). Counts if PDI Watchdog expires. Cleared if one of the Watchdog counters 0x0442:0x0443 is written.	Rw	Ro	0

ESC DL Status [0x0110:0x0111]				
Bit	**Description**	**ECAT**	**PDI**	**Default**
0	PDI operational/EEPROM loaded correctly: 0: EEPROM not loaded, PDI not operational (no access to Process Data RAM) 1: EEPROM loaded correctly, PDI operational (access to Process Data RAM)	Ro	Ro	0
1	PDI Watchdog Status: 0: Watchdog expired 1: Watchdog reloaded	Ro	Ro	0

PRELIMINARY

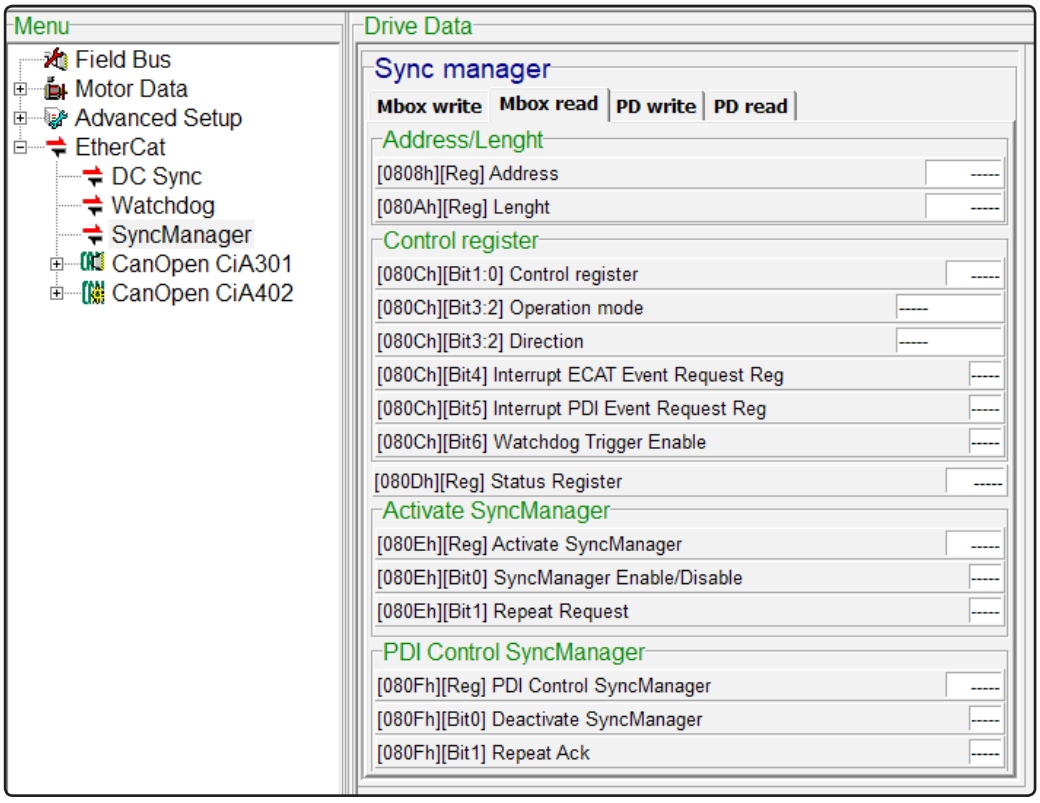
3.05 SyncManager

The EtherCAT master and the microcontroller both access the ESC memory to exchange process data and parameters. However, using the memory for communication like this has some drawbacks:

- Data consistency is not guaranteed. Semaphores have to be implemented in software for exchanging data in a coordinated way.
- Data security is not guaranteed. Security mechanisms have to be implemented in software.
- Both EtherCAT master and microcontroller have to poll the memory in order to get to know when the access of the other side has finished.

To address these problems the ESCs use mechanisms called ***SyncManagers***, which enable consistent and secure data exchange between the EtherCAT master and the microcontroller, and generate interrupts to inform both sides of any change. SyncManagers use a buffer located in the memory area for exchanging data; access to this buffer is controlled by their hardware. SyncManagers are configured by the EtherCAT master.

The ***SyncManager*** configuration registers can be seen in the menu "***EtherCat - SyncManager***" in which the following screen appears:



Our Drives use 4 ***SyncManagers***, with 4 different functions. Since all ***SyncManagers*** use configuration registers with the same function, to avoid repetition we described them only once and used simple formulas to report their addresses for every ***Syncmanager***.

The first ***SyncManager*** is used for ***Mailbox write***, and its configuration registers start at address 0x800. We consider its addresses as the starting points (i.e y=0 in the following formulas). The second ***SyncManager*** is used for ***Mailbox Read***, and the adresses of its configuration registers are shifted of 8 (i.e y=1). The third is used for ***Process Data write*** and the adresses of its configuration registers are shifted of another 8 (i.e y=2). The fourth is used for ***Process data read*** and the adresses of its configuration registers are shifted of another 8 (i.e y=3).

physical Start Address SyncManager y [0x0800+y*8:0x0801+y*8]				
Bit	***Description***	***ECAT***	***PDI***	***Default***
15:0	Specifies first byte that will be handled by SyncManager	Rw	Ro	0

Length SyncManager y [0x0802+y*8:0x0803+y*8]				
Bit	**Description**	**ECAT**	**PDI**	**Default**
15:0	Number of bytes assigned to SyncManager (shall be greater 1, otherwise SyncManager is not activated. If set to 1, only Watchdog Trigger is generated if configured)	Rw	Ro	0

Control Register SyncManager y [0x0804+y*8]				
Bit	**Description**	**ECAT**	**PDI**	**Default**
1:0	***Operation Mode:*** 00: Buffered (3 buffer mode) 01: Reserved 10: Mailbox (Single buffer mode) 11: Reserved	Rw	Ro	00
3:2	***Direction:*** 00: Read: ECAT read access, PDI write access. 01: Write: ECAT write access, PDI read access. 10: Reserved 11: Reserved	Rw	Ro	00
4	***Interrupt in ECAT Event Request Register:*** 0: Disabled 1: Enabled	Rw	Ro	0
5	***Interrupt in PDI Event Request Register:*** 0: Disabled 1: Enabled	Rw	Ro	0
6	***Watchdog Trigger Enable:*** 0: Disabled 1: Enabled	Rw	Ro	0
7	Reserved, write 0	Rw	Ro	0

Status Register SyncManager y [0x0805+y*8]				
Bit	**Description**	**ECAT**	**PDI**	**Default**
0	***Interrupt Write***: 1: Interrupt after buffer was completely and successfully written 0: Interrupt cleared after first byte of buffer was read	Ro	Ro	0
1	***Interrupt Read:*** 1: Interrupt after buffer was completely and successful read 0: Interrupt cleared after first byte of buffer was written	Ro	Ro	0
2	Reserved, write 0	Ro	Ro	0
3	***in Mailbox mode-> mailbox status:*** 0: Mailbox empty 1: Mailbox full ***in Buffered mode-> reserved***	Ro	Ro	0

Status Register SyncManager y [0x0805+y*8]				
Bit	**Description**	**ECAT**	**PDI**	**Default**
5:4	***in Buffered mode->buffer status (last written buffer):*** 00: 1. buffer 01: 2. buffer 10: 3. buffer 11: (no buffer written) ***in Mailbox mode-> reserved***	Ro	Ro	11
7:6	Reserved, write 0	Ro	Ro	0

Activate SyncManager y [0x0806+y*8]				
Bit	**Description**	**ECAT**	**PDI**	**Default**
0	***SyncManager Enable/Disable:*** 0: Disable: Access to Memory without SyncManager control 1: Enable: SyncManager is active andcontrols Memory area set in configuration	Rw	Ro	0
1	***Repeat Request:*** A toggle of Repeat Request means that a mailbox retry is needed (primarily used in conjunction with ECAT Read Mailbox)	Rw	Ro	0
5:2	Reserved, write 0	Ro	Ro	0
7:6	Not used			

PDI Control SyncManager y [0x0807+y*8]				
Bit	**Description**	**ECAT**	**PDI**	**Default**
0	***Deactivate SyncManager:*** ***Read***: 0: Normal operation, SyncManager activated. 1: SyncManager deactivated and reset SyncManager locks access to Memory area. ***Write***: 0: Activate SyncManager 1: Request SyncManager deactivation *NOTE: Writing 1 is delayed until the end of a frame which is currently processed.*	Ro	Rw	0
1	***Repeat Ack:*** If this is set to the same value as set by Repeat Request, the PDI acknowledges the execution of a previous set Repeat request.	Ro	Rw	0
7:2	Reserved, write 0	Ro	Ro	0

PRELIMINARY

3.05.1 Buffered mode and Mailbox mode

The ***SyncManagers*** can operate in two modes: the ***Buffered Mode*** and the ***Mailbox Mode*** .

The ***Buffered Mode*** allows writing and reading data simultaneously without interference. If the buffer is written faster than it is read out, old data will be dropped. The buffered mode is also know as 3-buffermode. Physically, 3 buffers of identical size are used for buffered mode. The start address and size of the first buffer are configured on registers ***physical Start Address SyncManager y [0x0800+y*8:0x0801+y*8]*** and ***Length SyncManager y [0x0802+y*8:0x0803+y*8]***. The addresses of these buffers have to be used by the master and the microcontroller for reading/writing the data. Depending on the ***SyncManager*** state, accesses to the first buffer's (0) address range are redirected to one of the 3 buffers. The memory used for buffers 1 and 2 can not be used and should be taken into account for configuring other ***SyncManagers***. One of the three buffers is allocated to the producer (for writing), one buffer to the consumer (for reading), and the third buffer keeps the last consistently written data of the producer. The ***Status register of the SyncManager [0x0805+y*8]*** reflects the current state. The last written buffer is indicated by the Bits 4 and 5 (informative only, access redirection is performed by the ESC), as well as the interrupt states. If the SyncManager buffer was not written before, the last written buffer is indicated to be 3 (start/empty).

The ***Mailbox Mode*** only allows alternating reading and writing. This assures all data from the producer reaches the consumer. The mailbox mode uses just one buffer of the configured size. At first, after initialization/activation, the buffer is writeable. Once it is written completely, write access is blocked, and the buffer can be read out by the other side. After it was completely read out, it can be written again. The time it takes to read or write the mailbox does not matter. Bit 3 of the ***Status Register SyncManager y [0x0805+y*8]*** shows if the Mailbox is full or empty

The SyncManager mode is set on Bits 0 and 1 of the ***Control Register SyncManager y [0x0804+y*8] .***

3.05.2 *Interrupt and Watchdog Trigger Generation*

Interrupts can be generated when a buffer was completely and successfully written or read. You can also generate a watchdog trigger to rewind (trigger) the ***Process Data watchdog*** used for Digital I/O. Interrupt and watchdog trigger generation is configured using bits 4,5,6 of the ***Control Register SyncManager y [0x0804+y*8].*** The ***Status SyncManager register y [0x0805+y*8]*** reflects the current buffer state.

3.05.3 *Repeating Mailbox Communication*

A lost datagram with mailbox data is handled by the application layer. The ***Repeat Request*** bit of the ***Activate SyncManager y [0x0806+y*8]*** and the ***Repeat Acknowledge*** bit of the ***PDI Control SyncManager y [0x0807+y*8]*** are used in mailbox mode for retransmissions of buffers from a slave to the master. If a mailbox read frame gets lost/broken on the way back to the master, the master can toggle the ***Repeat Request*** bit. The slave polls this bit or receives an interrupt (***SyncManager activation register changed [0x0220.4]***) and writes the last buffer again to the ***SyncManager***. Then the PDI toggles the ***Repeat Acknowledge*** bit in the ***PDI Control register***. The master will read out this bit and read the buffer content. Communication resumes afterwards.

3.05.4 *SyncManager Deactivation by the PDI*

A ***SyncManager*** can be deactivated by the PDI to inform the master of local problems (typically used in buffered mode only). The master can detect ***SyncManager*** deactivation by checking the ***Working Counter***, which is not incremented if a deactivated ***SyncManager*** buffer is accessed. If a ***SyncManager*** is deactivated by the PDI (using bit 0 of ***PDI Control SyncManager y [0x0807+y*8]***), the state of the ***SyncManager*** is reset, interrupts are cleared and the ***SyncManager*** has to be written first after reactivation. The entire ***SyncManager*** buffer area is read/write protected while the ***SyncManager*** is deactivated by the PDI.

3.05.5 Sync Manager PDO

Sync manager type [0x 1C00]				
sub-index	***Description***	***type***	***Attributes***	***Map***
0	Number of used Sync Manager Channels	UINT8	Ro	No
1	Sync-Manager Type Channel 1: Mailbox Write	UINT8	Ro	No
2	Sync-Manager Type Channel 2: Mailbox Read	UINT8	Ro	No
3	Sync-Manager Type Channel 3: Process Data Write (Outputs)	UINT8	Ro	No
4	Sync-Manager Type Channel 4: Process Data Read (Inputs)	UINT8	Ro	No

RPDO assign [0x1C12]				
sub-index	***Description***	***type***	***Attributes***	***Map***
0	Number of Assigned RPDO (max. 32)	UINT8	Rw	No
1	Contains the index of the corresponding RPDO mapping object	UINT16	Rw	No
2		UINT16	Rw	No
3		UINT16	Rw	No
4		UINT16	Rw	No

TPDO assign [0x1C13]				
sub-index	***Description***	***type***	***Attributes***	***Map***
0	Number of Assigned TPDO (max. 32)	UINT8	Rw	No
1	Contains the index of the corresponding TPDO mapping object	UINT16	Rw	No
2		UINT16	Rw	No
3		UINT16	Rw	No
4		UINT16	Rw	No

3.05.6 Sync Manager synchronization

SM output parameter [0x1C32]				
sub-index	***Name***	***type***	***Attributes***	***Map***
0	SyncManager output parameter	UINT8	Ro	
1	Synchronization type	UINT16	Rw	
2	Cycle time	UINT32	Rw	
4	Synch. types supported	UINT16	Ro	
5	Minimum cycle time	UINT32	Ro	
6	Calc. and copy time	UINT32	Ro	
9	Delay time	UINT32	Ro	

The value of the ***Synchronization type [0x1C32.1]*** indicates the types of synchronization actually used by the Slave:

» 0 = "***Free run***"

» 1="***Synchronous with SyncManager 2 event mode***"

» 2="***DC mode-synchronous with SYNC0 event***"

» 3="***DC mode-synchronous with SYNC1 event***"

The value of the ***Cycle time [0x1C32.2]*** indicates time between two events of the microcontroller local timer, expressed in [ns]

» ***Free run***: cycle time of the local timer
» ***Synchronization with SM2 Event***: Master cycle time
» ***DC Mode***: Sync0/Sync1 Cycle time

The bits of ***Synch. types supported [0x1c32.4]*** indicates what synchronization types are supported

» ***Bit 0 =1*** : Free run supported
» ***Bit 1=1:*** Synchronization with SM2 Event supported
» ***Bit 3,2 =01***: DC mode supported

the default value is 7, which means that the Drives support all the types above

The value of the ***Minimum cycle time [0x1C32.5]*** is the minimum time the slave needs to update process data once and still servicing acyclic tasks (for example copying process data from/to the SyncManager, Processing process data, mailbox services and local functions)

The value of the ***Calc and copy time [0x1C32.6]*** is an optional information which represents the time needed to the slave for processing data (input or output) and copying them from the ***ESC*** to the microcontroller

The value of the ***Delay time [0x1C32.9]*** represents a delay inserted after the occurrence of the DC sync0 event.If you're sending data, when this time interval is over the output is considered valid; if you're waiting for input data, when this time interval is over the Latch Trigger occurs.

SM input parameter [0x1C33]				
sub-index	***Name***	***type***	***Attributes***	***Map***
0	SyncManager input parameter	UINT8	Ro	
1	Synchronization type	UINT16	Rw	
2	Cycle time	UINT32	Rw	
4	Synch. types supported	UINT16	Ro	
5	Minimum cycle time	UINT32	Ro	
6	Calc. and copy time	UINT32	Ro	
9	Delay time	UINT32	Ro	

PRELIMINARY

Ch. 4 DS-402 protocol parameters

The ***DS-402*** protocol defines a group of common variables used to configure and control Drives for motor control, in particular for the following control types:

- Profile Position mode
- Homing Position
- Profile Velocity Mode
- Velocity Mode
- Interpolated Position mode
- Profile Torque Mode

H.D.T Drives also support other control types in addition to the ones reported on the ***DS-402*** protocol:

- Gearbox Mode
- Press Mode

4.01 State Machine

The status of the Drive is controlled using a Finite State Machine, which handles the enabling and disabling phases of the Drive and the fault state caused by an alarm.

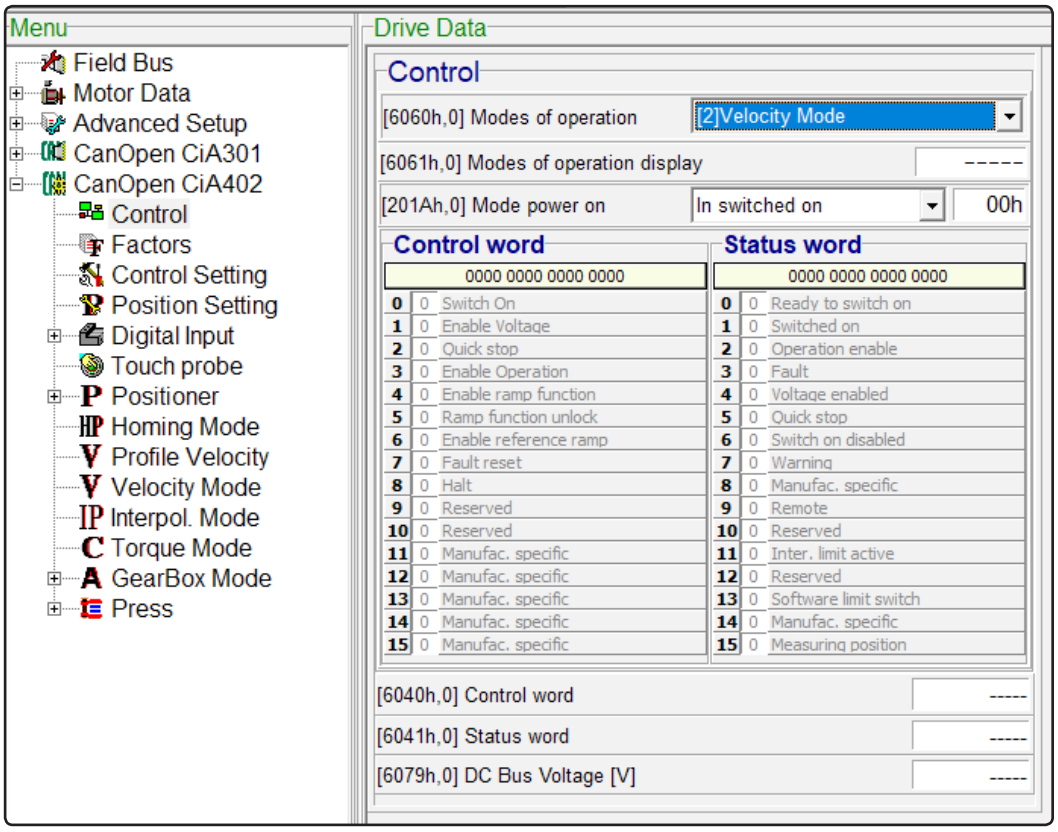
State	Description
NOT READY TO SWITCH ON	• The Drive is not ready yet, because it's loading and initializing its parameters
SWITCH ON DISABLE	• All parameters are all initialized, now their values can also be changed. • The power is disabled
READY TO SWITCH ON	• The Drive can be powered by the main supply • The parameters values can be changed • The power is disabled
SWITCH ON	• The Drive is powered by the main supply • The parameters values can be changed • The power is enableded • The Drive is ready to follow the control reference
OPERATION ENABLED	• The Drive has no alarms • The parameters values can be changed • The power is enabled • The Drive is following the control reference • The motor is powered and controlled by the Drive
QUICK STOP ACTIVE	• The Master has sent a stop command to the Drive, which is executed by the predisposed emergency function • The parameters values can be changed • The power is enabled • The motor is powered and controlled by the Drive
FAULT REACTION ACTIVE	• An alarm occurred which caused the execution of the predisposed emergency function • The parameters values can be changed
FAULT	• The Drive has an alarm, the predisposed ***Fault Reaction*** function initiated by the alarm is terminated • The parameters values can be changed • The power is disabled

PRELIMINARY

All the possible conditions that imply a variation of the Drive's state are reported on the ***Statusword***, a read-only 16 bits variable, which is located on index [6041h] of the ***Object Dictionary***. So by reading the ***Statusword*** it is possible to check at any time what the state of the State Machine is.

The commands used to change the Drive state are handled by the master which sets them on the ***Controlword***, a 16-bit variable, which is located on index [6040h] of the ***Object Dictionary***.

These variables can be seen in the menu "***EtherCat - SyncManager***" in which the following screen appears:



PRELIMINARY

4.02 PDO settings with default mapping

As described in the chapter of the communication parameters, 4 RPDO and 4 TPDO are available in the Drive, which are already configured with default values. The set values can be changed, since the dynamic remapping of the PDO is allowed.

4.02.1 RPDO 1 [index 1600h] default mapping

Receive PDO 1			
Index	**Sub-Index**	**Nome**	**Valore Default**
1600h	0	Number od mapped objects	3
	1	Controlword	6040 0010h
	2	Target velocity	6042 0010h
	3	Home offset	607C 0020h
	4...8	-	0

4.02.2 RPDO 2 [index 1601h] default mapping

Receive PDO 2			
Index	**Sub-Index**	**Nome**	**Valore Default**
1601h	0	Number of mapped objects	2
	1	Target position	607A 0020h
	2	Profile velocity	6081 0020h
	3..8	--	0

4.02.3 RPDO 3 [index 1602h] default mapping

Receive PDO 3			
Index	**Sub-Index**	**Nome**	**Valore Default**
1602h	0	Number of mapped objects	2
	1	Profile acceleration	6083 0020h
	2	Profile deceleration	6084 0020h
	3..8	--	0

4.02.4 RPDO 4 [index 1603h] default mapping

Receive PDO 4			
Index	**Sub-Index**	**Nome**	**Valore Default**
1603h	0	Number of mapped objects	1
	1	Interpolation data record (sub-index 1 Posizione)	60C1 0120h
	2..8	--	0

4.02.5 TPDO 1 [index 1A00h] default mapping

Transmit PDO 1			
Index	**Sub-Index**	**Nome**	**Valore Default**
1A00h	0	Number of mapped objects	3
	1	Statusword	6041 0010h
	2	Modes of operation display	6061 0008h
	3	Error Register	1001 0008h
	4..8	--	0

4.02.6 TPDO 2 [index 1A01h] default mapping

Transmit PDO 2			
Index	**Sub-Index**	**Nome**	**Valore Default**
1A01h	0	Number of mapped objects	2
	1	Position demand value	6062 0020h
	2	Position actual value	6064 0020h
	3..8	--	0

PRELIMINARY

4.02.7 TPDO 3 [index 1A02h] default mapping

Transmit PDO 3			
Index	**Sub-Index**	**Nome**	**Valore Default**
1A02h	0	Number of mapped objects	2
	1	vl velocity demand	6043 0010h
	2	vl control effort	6044 0010h
	3..8	--	0

4.02.8 TPDO 4 [index 1A03h] default mapping

Transmit PDO 4			
Index	**Sub-Index**	**Nome**	**Valore Default**
1A03h	0	Number of mapped objects	1
	1	Digital inputs	60FD 0020h
	2..8	--	0

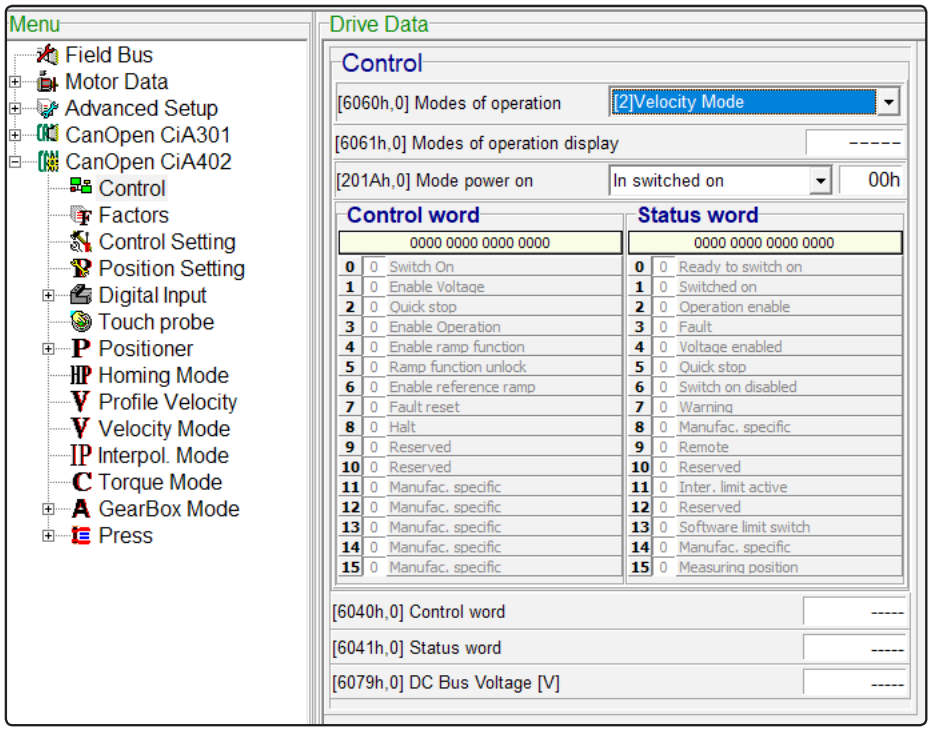
PRELIMINARY

4.03 DS-402 Control parameters

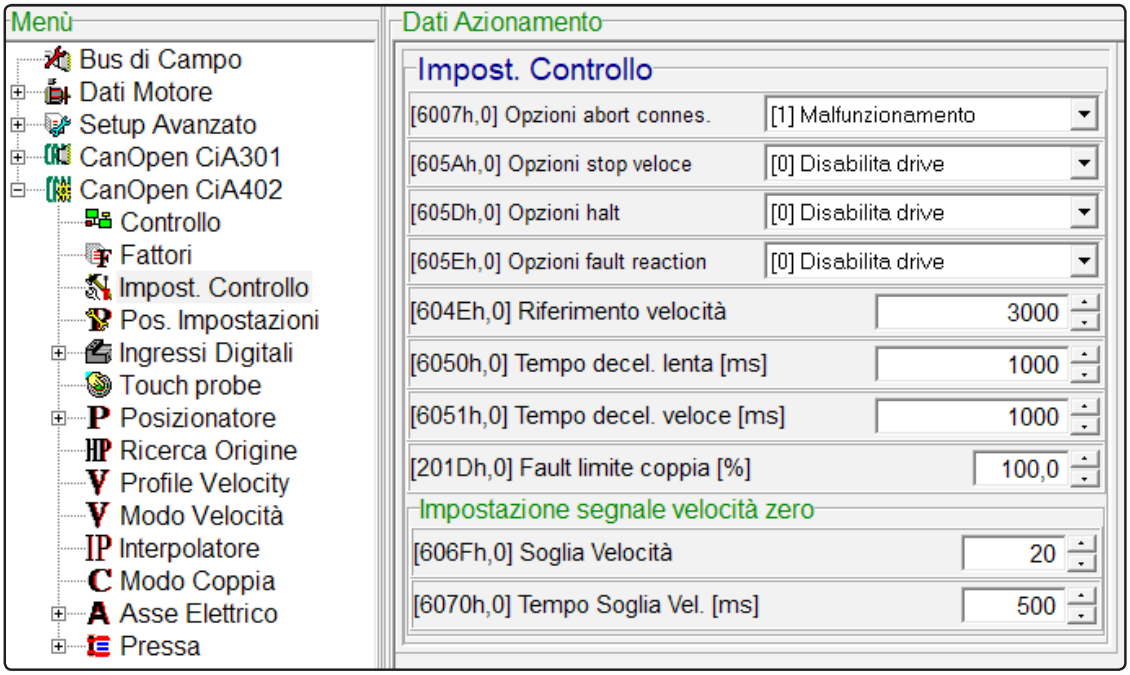
The ***DS-402*** protocol, in addition to the ***Controlword*** and the ***Statusword***, defines a set of parameters used to set the control type and the Drive's reaction after a ***fault*** event.

Index	*Name*	*Type*	*Map*	*Attributes*
6040h	Controlword	UINT16	Yes	rw
6041h	Statusword	UINT16	Yes	ro
6007h	Abort conn. option code	INT16	No	rw
603Fh	Error code	UINT16	Yes	ro
604Eh	Velocity reference	UINT32	No	rw
6050h	Slow down time	UINT32	Yes	rw
6051h	Quick down time	UINT32	Yes	rw
605Ah	Quick stop option code	INT16	No	rw
605Dh	Halt option code	INT16	No	rw
605Eh	Fault reaction options	INT16	No	rw
6060h	Modes of operation	INT8	Yes	rw
6061h	Modes of operation display	INT8	Yes	ro

These variables can be seen in the menu "***(EtherCat)-CanOpen CiA402-Control***" in which the following screen appears:



and in the menu "***(EtherCat)-CanOpen CiA402- Control Setting***":



PRELIMINARY

4.03.1 Object 6040h - Controlword

Index	Name	Type	Range	Default	Unit	Map	Attributes
6040h	**Controlword**	UINT16	0..65535	0		Yes	rw

bit	*Controlword- bits description:*
0	Switch on
1	Enable voltage
2	Quick stop
3	Enable operation
6..4	Operation mode specific
7	Fault reset
8	halt
10..9	reserved
15..11	manufacturer specific

PRELIMINARY

4.03.1.a Controlword bits 0..3 and 7

Bits 0..3 and 7 are used as comands to switch the state of the ***state machine*** according to the following table:

Comands	*Controlword bits*					*Transictions*
	Fault reset	*Enable operation*	*Quick stop*	*Enable voltage*	*Switch on*	
Shutdown	0	x	1	1	0	2,6,8
Switch on	0	0	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

The bits marked with an "x" are irrelevant

4.03.1.b Controlword bits 4,5,6 and 8

These bits are related to the operating mode:

Operating mode	Bit			
	4	**5**	**6**	**8**
Press	Reserved	Set mode press/vel	Enable setpoint	Halt
Gearbox	Enable axis	Reset encoder	Reset position	Halt
Velocity	Enable ramp function	Ramp function unlock	Enable ref. ramp	Halt
Position	New setpoint	Change set. immediately	Absolute/relative	Halt
Profile velocity	Reserved	Reserved	Reserved	Halt
Profile torque	Reserved	Reserved	Reserved	Halt
Homing	Homing start	Reserved	Reserved	Halt
Interpolator	Enable ip. mode	Reserved	Reserved	Halt
Cyclic Sync. position	Reserved	Reserved	Reserved	Halt
Cyclic Sync. velocity	Reserved	Reserved	Reserved	Halt
Cyclic Sync. torque	Reserved	Reserved	Reserved	Halt

4.03.2 Object 6041h - Statusword

Index	Name	Type	Range	Default	Unit	Map	Attributes
6041h	**Statusword**	UINT16	0..65535	0		Yes	ro

Bit	Statusword bits description
0	Ready to switch on
1	Switched on
2	Operation enabled
3	Fault
4	Voltage enabled (*)
5	Quick stop
6	Switch on disabled
7	Warning
8	Manufacturer specific
9	Remote
10	Target reached
11	Internal limit active (**)
12..13	Operation mode specific
14..15	Manufacturer specific

(*) When bit 4 of the Statusword is high it means that power is supplied to the motor.

(**) Only for the Press mode- when bit 11 of the Statusword is high it means that the underpressure limit has been reached.

PRELIMINARY

4.03.2.a Statusword bits related to the Drive status

Status	Bit statusword
NOT READY TO SWITCH ON	xxxx xxxx x0xx 0000
SWITCH ON DISABLE	xxxx xxxx x1xx 0000
READY TO SWITCH ON	xxxx xxxx x01x 0001
SWITCH ON	xxxx xxxx x01x 0011
OPERATION ENABLED	xxxx xxxx x01x 0111
QUICK STOP ACTIVE	xxxx xxxx x00x 0111
FAULT REACTION ACTIVE	xxxx xxxx x0xx 1111
FAULT	xxxx xxxx x0xx 1000

The bits marked with an "x" are irrelevant

4.03.2.b Statusword bits 12,13,14 and 15 - operating mode

Operating mode	Bit			
	12	13	14	15
Press	Overpressure	power-limit	Mode Press-Velocity	Manufacturer specific
Gearbox	Gearbox enabled	Following error	Manufacturer specific	measuring position
Velocity	Reserved	Inter. limit active	Manufacturer specific	measuring position
Position	Setpoint acknowledged	Following error	Pos-Input 1 reached	measuring position
Profile velocity	Zero speed	Inter. limit active	Manufacturer specific	measuring position
Profile Torque	Reserved	Reserved	Manufacturer specific	measuring position

Operating mode	Bit			
	12	13	14	15
Homing	Homing attained	Homing error	Manufacturer specific	measuring position
Interpolator	Ip. mode active	Reserved	Manufacturer specific	measuring position
Cyclic Sync. position	Target pos. used/ ignored	Following error	Manufacturer specific	measuring position
Cyclic Sync. velocity	Target vel. used/ ignored	Reserved	Manufacturer specific	measuring position
Cyclic Sync. torque	Target torque used/ ignored	Reserved	Manufacturer specific	measuring position

4.03.3 Object 6007h - Abort connection otpion code

Index	Name	Type	Range	Default	Unit	Map	Attributes
6007h	**Abort conn. option code**	UINT16	0..3	0		Yes	rw

This parameter defines how to respond if the communication with the network is lost.

Values	Abort connection option code	Description
0	No action	No action is performed
1	Malfunction	Power is disabled and the motor is free to move. The Drive goes on "***Switch On Disabled***" state.
2	Disable voltage	Power is disabled and the motor is free to move. The Drive goes on "***Switch On Disabled***" state.
3	Quick stop	A ***Quick Stop*** is perfomed with the deceleration obtained by the ratio between ***Velocity reference (index 604Eh)*** and ***Quick down time (index 6051h)***, at the end of the ramp the motor stand still with the power on and the Drive goes on "***Operation Enabled***" state.

4.03.4 Object 603Fh - Error code

Index	Name	Type	Range	Default	Unit	Map	Attributes
603Fh	**Error code**	UINT16	0..65535	0		Yes	ro

In this variable is saved the last alarm occurred in the Drive, the error code corresponds to the 16 least significative bits of the 32 bit value stored on sub-index 1 of the object ***Pre-defined error field (index 1003h)***.

4.03.5 Object 604Eh - Velocity reference

Index	Name	Type	Range	Default	Unit	Map	Attributes
604Eh	**Velocity reference**	UINT32	0..7FFFFFFFh	3000		No	rw

The "***Velocity reference***" is used by the Drive to calculate the deceleration to apply during the "***Quick Stop***" and "***Halt***" phases. This parameter has the same unit of measure as the ***vl target velocity (index 6042h)***, and is multiplied by the "***Numerator***" and divided by the "***Denominator***" of ***vl dimension factor (index 604Ch)***, to get the speed in [rpm].

4.03.6 Object 605Ah - Quick stop option code

Index	Name	Type	Range	Default	Unit	Map	Attributes
605Ah	**Quick stop option code**	INT16	0..6	0		Yes	rw

The "***Quick stop option code***" parameter defines how the Drive will respond when it goes on "***Quick stop Active***" state. The possible options are listed in the following table.

Value	Quick stop option	Description
0	disable Drive	Power is disabled and the motor is free to move. The Drive goes in "***Switch On Disabled*** " state.
1	Slow down ramp	The Drive switches to ***speed control***, and the motor is commanded a stop in ramp with a deceleration obtained by the ratio between "***Velocity reference***" (***index 604Eh***) and "***Slow down time***" (***index 6050h***). At the end of the ramp the Drive goes in "***Switch On Disabled***" state (power is disabled).
2	Quick stop ramp	The Drive switches to ***speed control***, and the motor is commanded a stop in ramp with a deceleration obtained by the ratio between "***Velocity reference***" (***index 604Eh***) and "***Quick down time***" (***index 6051h***). At the end of the ramp the Drive goes in "***Switch On Disabled***" state (power is disabled).
5	Slow stop ramp (in quick stop)	The Drive switches to ***speed control***, and the motor is commanded a stop in ramp with a deceleration obtained by the ratio between "***Velocity reference***" (***index 604Eh***) and "***Slow down time***" (***index 6050h***). At the end of the ramp the Drive goes in "***Quick Stop Active***" state (power is still supplied).
6	Quick stop ramp (in quick stop)	The Drive switches to ***speed control***, and the motor is commanded a stop in ramp with a deceleration obtained by the ratio between "***Velocity reference***" (***index 604Eh***) and "***Quick down time***" (***index 6051h***). At the end of the ramp the Drive goes in "***Quick Stop Active***" state (power is still supplied).

PRELIMINARY

4.03.7 Object 605Dh - Halt option code

Index	Name	Type	Range	Default	Unit	Map	Attributes
605Dh	**Halt option code**	INT16	0..2	0		Yes	rw

The "***Halt option code***" parameter defines how the Drive will respond when it's commanded an ***Halt*** (by setting bit 8 of the ***controlword*** to 1). Depending on the value set on this paramater the Drive will perform one of the following actions:

Values	Halt options	Description
0	disable Drive	• Power is disabled and the motor is free to move. • The Drive goes in "***Switch On Disabled*** " state.
1	Slow down ramp	• The Drive switches to ***speed control***, and the motor is commanded a stop in ramp with a deceleration obtained by the ratio between "***Velocity reference***" (***index 604Eh***) and "***Slow down time***" (***index 6050h***). • At the end of the ramp the Drive goes in "***Operation enabled***" state.
2	Quick stop ramp	• The Drive switches to ***speed control***, and the motor is commanded a stop in ramp with a deceleration obtained by the ratio between "***Velocity reference***" (***index 604Eh***) and "***Quick down time***" (***index 6051h***). • At the end of the ramp the Drive goes in "***Operation enabled***" state.

4.03.8 Object 6050h - Slow down time

Index	Name	Type	Range	Default	Unit	Map	Attributes
6050h	**Slow down time**	UINT32	0..60000	1000	ms	Yes	rw

This parameter is the time length of the slow deceleration ramp with which the motor will stop when an ***Halt*** or a ***Quick stop*** is commanded. The deceleration of this ramp is internally calculated by making the ratio between "***Velocity reference***" (***index 604Eh***) and the time set on this parameter. A longer time corresponds to a smoother ramp.

4.03.9 Object 6051h - Quick down time

Index	Name	Type	Range	Default	Unit	Map	Attributes
6051h	**Quick down time**	UINT32	0..60000	1000	ms	Yes	rw

This parameter is the time length of the fast deceleration ramp with which the motor will stop when an ***Halt*** or a ***Quick stop*** is commanded. The deceleration of this ramp is internally calculated by making the ratio between "***Velocity reference***" (***index 604Eh***) and the time set on this parameter. A shorter time corresponds to a quicker ramp.

PRELIMINARY

4.03.10 Object 605Eh - Fault reaction options

Index	Name	Type	Range	Default	Unit	Map	Attributes
605Dh	**Opzioni Fault reaction**	INT16	0..3	0		No	rw

The "***Fault reaction options***" parameter defines what action the Drive will perform when certain faults occur. In these cases the Drive remains in "***Fault reaction***" state while it's executing this action, and when it's completed the power is disabled and the Drive switch to the "***Fault***" state.

Values	Fault reaction options	Description
0	disable Drive	• Power is disabled and the motor is free to move. • The Drive goes in "***Fault***" state.
1	Slow down ramp	• The Drive switches to ***speed control***, and the motor is commanded a stop in ramp with a deceleration obtained by the ratio between "***Velocity reference***" (***index 604Eh***) and "***Slow down time***" (***index 6050h***). • At the end of the ramp the Drive goes in "***Fault***" state.
2	Quick stop ramp	• The Drive switches to ***speed control***, and the motor is commanded a stop in ramp with a deceleration obtained by the ratio between "***Velocity reference***" (***index 604Eh***) and "***Quick down time***" (***index 6051h***). • At the end of the ramp the Drive goes in "***Fault***" state.
3	Torque limit	• The Drive stops without exceeding the torque limit set on "***Fault torque limit***" (***index 201Dh***). • At the end of the ramp the Drive goes in "***Fault***" state.

The action selected by the "***Fault reaction options***" parameter will be performed only for the following alarms:

Alarms managed by the fault reaction options		
Error code	**DS-301 description**	**Detailed Description**
3220h	DC link Under Voltage	DC BUS under the voltage limit
8611h	Following Error	Position error is too high

4.03.11 Object 6060h - Modes of operation

Index	Name	Type	Range	Default	Unit	Map	Attributes
6060h	**Modes of operation**	INT8	-2..10	2		Yes	rw

This parameter allows you to set the desired type of control, depending on the application requirements. In the table below are reported the control types supported by the Drive:

Value	operating mode	Description
-2	***Press***	This mode is used to control the pressure of an hydraulic control unit; this is done controlling the servo pump installed on it. the current loop, speed loop and pressure loop are abilitated. The pressure loop takes as a reference an analogic signal, and confronts it with the analogic signal produced by a transducer that measures the real pressure. The pressure loop then produces the speed reference for the speed loop. The speed profile and the pressure profile can be performed with trapezoidal shape or with an "S" shape. This is a custom control, not included in the ***DS-402***
-1	***Gearbox***	With this control type the motor follows the position reference resulting by the product of the number of pulses received from an external encoder and the chosen gearbox ratio. This is a custom control, not included in the ***DS-402***
1	***Position***	In this mode the current loop, speed loop and position loop are abilitated; once you have set the desired position the drive moves towards it. The movement can be performed following a speed profile with trapezoidal shape or with an "S" shape; the latters are smoothed by the jerk set on ***Position jerk (index 2011h)***.
2	***Velocity***	The current loop and the speed loop are activated, the drives follow the speed reference sent by the Master according with the parameters related with this mode
3	***Profile Velocity***	The current loop and the speed loop are activated, the drives follow the speed reference sent by the Master according with the parameters related with this mode
4	***Torque***	In this mode the motor torque is controlled by adjusting the quadrature current with a current loop (PID)
6	***Homing***	The current, speed and position loops are activated, the Drive is set up to execute an home position procedure according with the set parameters.
7	***Interpolator***	The current, speed and position loops are activated, the Drive executes the interpolation of the position data sent by the Master.
8	***Cyclic Sync. position Mode***	This mode is basically the same as the "***Position***" mode, with the excepition that the calculation of the position profile (limits and ramps included) is entirely made by the Master, and not by the Drive. The Master send the position reference to the Drive periodically, and optionally it could also send a speed offset and a current offset to be used as feedforward to improve the position control.
9	***Cyclic Sync. velocity mode***	This mode is basically the same as the "***velocity***" mode, with the excepition that the calculation of the velocity profile (limits and ramps included) is entirely made by the Master, and not by the Drive. The Master send the speed reference to the Drive periodically, and optionally it could also send a current offset to be used as feedforward to improve the speed control.
10	***Cyclic Sync. torque mode***	This mode is basically the same as the "***torque***" mode, with the excepition that the calculation of the torque profile (limits and ramps included) is entirely made by the Master, and not by the Drive. The Master send the torque reference to the Drive periodically.

4.03.12 Object 6061h - Modes of operation display

Index	Name	Type	Range	Default	Unit	Map	Attributes
6061h	**Modes of operation display**	INT8	----	----		Yes	ro

This parameter shows the operating mode, i.e. the control type currently used by the Drive, set with the parameter ***Modes of operation (index 6060h)***.

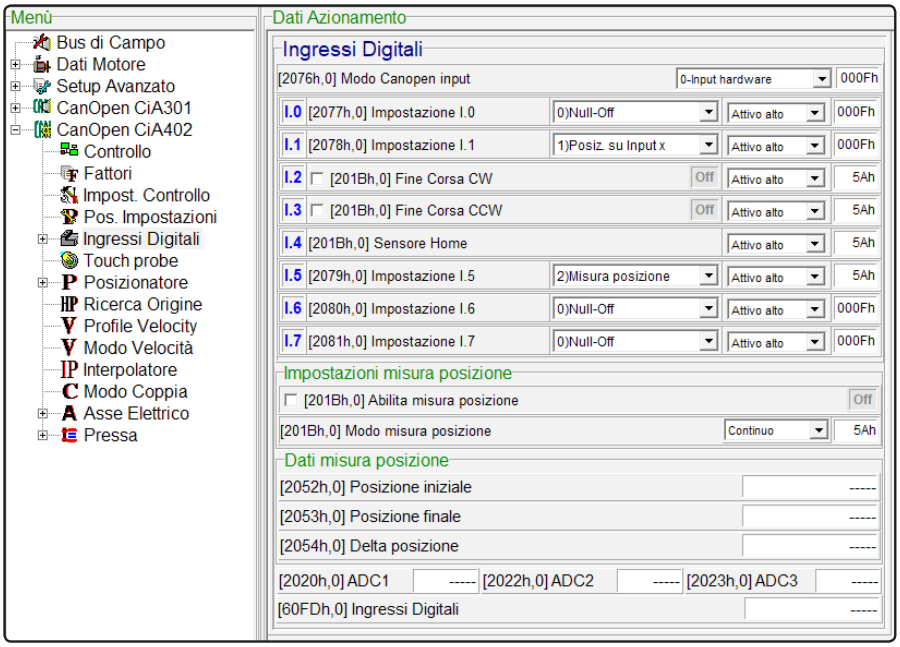
4.04 Digital Inputs

This section reports the settings and functions related to the digital and analogical inputs provided.

Index	Name	Type	Map	Attributes
60FDh	Digital input	UINT32	Yes	ro
201Bh	Input setting	UINT16	No	rw
2020h	ADC 1	INT16	Yes	ro
2022h	ADC 2	INT16	Yes	ro
2023h	ADC 3	INT16	Yes	ro
2052h	Measured initial position	INT32	Yes	ro
2053h	Measured final position	INT32	Yes	ro
2054h	Measured Delta position	INT32	Yes	ro
2055h	Filter time input 0..7	(ARRAY) UINT16	No	rw
2055h	Filter time input 8.. 10	(ARRAY) UINT16	No	rw
2076h	Canopen input mode	UINT16	No	rw
2077h	Setting I.0	UINT8	No	rw
2078h	Setting I.1	UINT8	No	rw
2079h	Setting I.5	UINT8	No	rw
2080h	Setting I.6	UINT8	No	rw
2081h	Setting I.7	UINT8	No	rw

PRELIMINARY

These parameters can be seen in the menu "***(EtherCat)-CanOpen CiA402-Digital input***" in which the following screen appears:



4.04.1 Object 60FDh - Digital input

Index	Name	Type	Range	Default	Unit	Map	Attributes
60FDh	**Digital input**	UINT32	-----	0		Yes	ro

The Drive contains some digital inputs, these bits are located between bit 16 and bit 24 of this object. The "***limit Switch CW***", "***limit Switch CCW***" and "***Home Switch***", used for the homing function, have to be connected via hardware to the predisposed inputs which are:

- ***Input 2***: for the "***limit switch CW***"
- ***Input 3***: for the "**limit switch CCW**"
- ***Input 4***: for the "***Home Switch***"

The inputs predisposed for the limit switches are reported not only on bits 18,19 and 20, but also on bits 0,1 and 2 as prescribed by the ***DS-402*** standard. Bits 16,17,21,22 and 23 are assigned with other functions, that will be described in the manual. For the wiring of the inputs, consult the hardware guide of the Drive.

PRELIMINARY

Bit Input	Description
0	Input limit switch CW
1	Input limit switch CCW
2	Input Home Switch
0..15	Not used
16	input 0: generic
17	input 1: generic
18	Input 2: limit switch CW
19	Input 3: limit switch CCW
20	Input 4: Home switch
21	Input 5: generic
22	Input 6: generic
23	Input 7: generic
24..31	Not used

4.04.2 Object 201Bh – Input setting

Index	Name	Type	Range	Default	Unit	Map	Attributes
201Bh	**Input setting**	UINT8	1..FFh	5Ah		No	rw

This parameter contains a series of flags used to set the inputs, as described in the table below:

Bit	Description	
0	**Enable/disable limit switch CW**	
	Value	**Description**
	0	***Limit switch CW disabled***
	1	***Limit switch CW enabled***: if the motor is turning clockwise when the ***limit switch CW*** is activated it get stopped. (*) with the "***Homing***" mode it's not necessary to enable the ***limit switch CW*** to execute the homing procedure, since in this case it's automatically enabled and disabled during its execution.

Bit	Description	
1	**Contact type Limit switch CW**	
	Value	**Description**
	0	The ***limit switch CW*** is considered active when the voltage on its input is 0 [V]
	1	The ***limit switch CW*** is considered active when the voltage on its input is 24 [V]
2	**Enable/disable limit switch CCW**	
	Value	**Description**
	0	***Limit switch CCW disabled***
	1	***Limit switch CCW enabled***: if the motor is turning counterclockwise when the ***limit switch CCW*** is activated it get stopped. (*) with the "***Homing***" mode it's not necessary to enable the ***limit switch CCW*** to execute the homing procedure, since in this case it's automatically enabled and disabled during its execution.
3	**Contact type Limit switch CCW**	
	Value	**Description**
	0	The ***limit switch CCW*** is considered active when the voltage on its input is 0 [V]
	1	The ***limit switch CCW*** is considered active when the voltage on its input is 24 [V]
4	**Contact type home switch**	
	Value	**Description**
	0	The ***home switch*** is considered active when the voltage on its input is 0 [V]
	1	The ***home switch*** is considered active when the voltage on its input is 24 [V]
5	**Enable/disable position measuring from assigned input**	
	Value	**Description**
	0	The storing of the measured position is disabled.
	1	the function that stores the measured position on the rising and falling edge of the selected input is enabled. The value acquired on the rising edge of the selected input is stored on the parameter "***measured initial position*** ***(index 2052h)***, the value acquired on the falling edge of the selected input is stored on the parameter "***measured final position*** ***(index 2053h)***, and the difference of these two values is stored on the parameter "***measured delta position*** ***(index 2054h)***. By setting the bit 7 of the "***Input setting*** ***(index 201Bh)*** parameter to 0, during sampling phase of the two edges the bit 15 of the ***statusword*** is put to 1, and the value of "***measured delta position*** ***(index 2054h)*** is valid only after the return of this bit to 0. Instead, By setting the bit 7 of the "***Input setting***" to 1, the bit 15 of the ***statusword*** rises to 1 only after the sampling on the second edge of the selected input, to show that "***measured delta position***" was calculated.
6	**Contact type input 5**	
	Value	**Description**
	0	Input 5 is considered active when the voltage on its input is 0 [V]
	1	Input 5 is considered active when the voltage on its input is 24 [V]
7	**Measuring position mode**	
	Value	**Description**
	0	***Continuous mode***: with this mode the function that measures the position on the edges of the assigned input continues to acquire new positions as long as bit 5 of "***input setting***" (***index 201Bh***) stays to 1.
	1	***One shot mode***: with this mode, when bit 5 of "***input setting***" (***index 201Bh***) is equal to 1, the Drive will acquire only the first two positions measured on the rising and falling edges of the assigned input, "***measured delta position***" (***index 2054h***) will be calculated only once and then the measuring function will be disabled. To turn it back on, it's necessary to reset and restore to 1 the bit 5 of "***input setting***" (***index 201Bh***) parameter.

PRELIMINARY

4.04.3 Object 2020h – ADC1

Index	Name	Type	Range	Default	Unit	Map	Attributes
2020h	**ADC 1**	INT16	-32678.. +32677			Yes	ro

This parameter contains the voltage value read by the external ad-converter of the main differential input. The voltage value can varies in the range ±10 [V], and has a resolution of 16 bit for ***NTT***, and12 bit for ***DGFox EVO*** and ***TomCat EVO***.

4.04.4 Object 2022h – ADC2

Index	Name	Type	Range	Default	Unit	Map	Attributes
2022h	**ADC 2**	INT16	-32678.. +32677			Yes	ro

This parameter contains the voltage value read by the external ad-converter of the first auxiliary differential input. The voltage value can varies in the range ±10 [V], and has a resolution of 12 bit.

4.04.5 Object 2024h – ADC3

Index	Name	Type	Range	Default	Unit	Map	Attributes
2024h	**ADC 3**	INT16	-32678.. +32677			Yes	ro

This parameter contains the voltage value read by the external ad-converter of the second auxiliary differential input. The voltage value can varies in the range ±10 [V], and has a resolution of 12 bit. This differential input is present only in the ***NTT***.

4.04.6 Position measuring function from input - setting

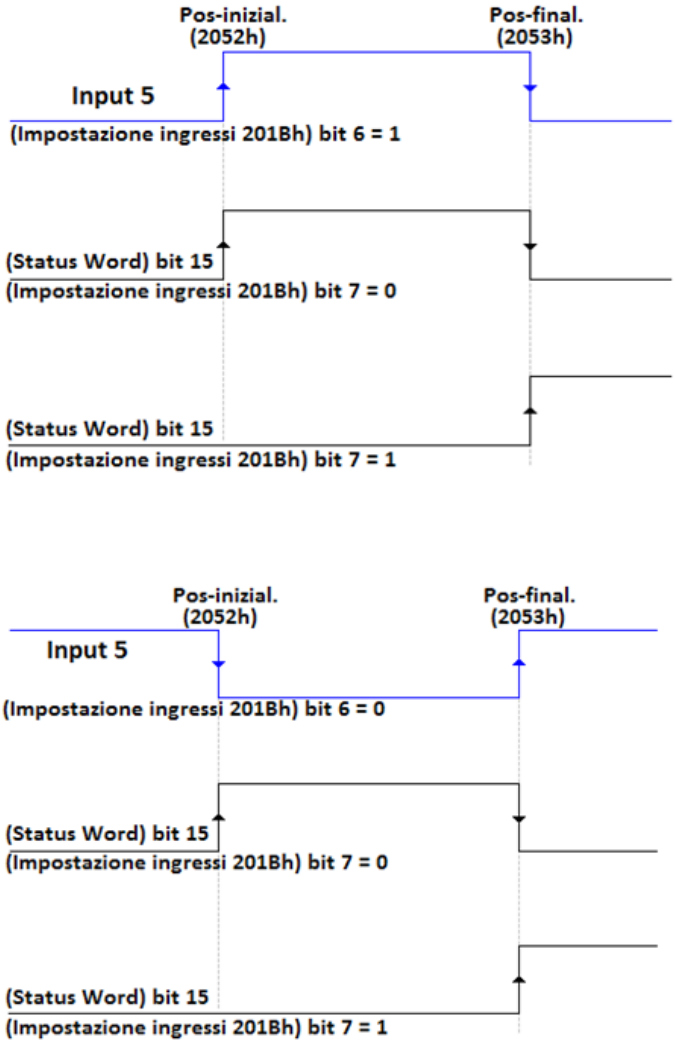
The function measuring two positions on command is enabled by putting bit 5 of "***input setting***" (***index 201Bh***) to 1 and by setting one of the digital inputs as a "reference" for these measures. The motor position read during a rising edge of the assigned input is saved on "***measured initial position***" (***index 2052h)*** parameter; the motor position read during a falling edge of the assigned input is saved on "***measured final position***" (***index 2053h***) parameter. The difference between these two values is calculated when the "***measured final position***" is saved, and is saved on "***measured delta position***" (***index 2054h***).

According to the value of bit 7 of the parameter "***input setting***" (***index 201Bh***), the measuring function could work in one of the two following modes:

- ***Continuous mode (Bit 7 = 0):*** the function that stores two motor positions when two edges are detected in the assigned input (and calculates their difference) continues to work as long as bit 5 of "***input setting***" (***index 201Bh***) is 1. Therefore these two positions and their difference are costantly updated. In this case bit 15 of the ***statusword*** will stay high only when the position are measured, i.e. when the input assigned to this function has an edge.
- ***One shot mode (Bit 7 = 1):*** with this mode, the two motor positions are acquired during the edge on the assigned input only once after bit 5 of "***input setting***" (***index 201Bh***) has been put to 1, and also their difference is calculated only once. To reactivate the measurements again it's necessary to reset and restore to 1 the bit 5 of "***input setting***". With this mode bit 15 of ***statusword*** is equal to 1 only when "***measured final position***" (***index 2053h***) is updated and "***measured delta position***" (***index 2054h***) is calculated; bit 15 returns to 0 when bit 5 of "***input setting***" is set to 0.

When the contact type of the input assigned to this function is set to "***active low***" the position measurement is done with the edges reversed, i.e. the value of ***measured initial position***" (***index 2052h)*** is acquired on the falling edge, while the "***measured final position***" (***index 2053h***) is acquired on the rising edge.

PRELIMINARY



PRELIMINARY

4.04.6.a Object 2052h - Measured initial position

Index	Name	Type	Range	Default	Unit	Map	Attributes
2052h	**Measured initial position**	INT32	----	----		Yes	ro

This object is used only if the position measurement is enabled using Bit 5 of "***input setting***" (see *"4.04.2 Object 201Bh – Input setting" pag. 50*). When one of the inputs is assigned to the "***Position measuring***" function, the position measured during the rising edge of this input is stored on the "***measured initial position***" (***index 2052h)*** parameter. This position is then converted from the internal position reference of the Drive to the Caliper's reference by using the parameters "***Numerator***" and "***Feedconstant***" of the ***"Pos factor"*** (***index 6093h***), with the formula reported below:

$$initial\_position = \frac{(Feed\_costant \times drive\_initial\_position)}{(Numerator)}$$

If the "***measuring position mode***" is set to "***continuous***" (bit 7 of 201Bh = 0) then the initial position will be costantly updated at every rising edge of the digital input.

Instead, if the "***measuring position mode***" is set to "***one shot***" (bit 7 of 201Bh = 1) then the initial position will be saved only at the first rising edge of the digital input and it's not updated anymore, unless this function is switched off and on again.

4.04.6.b Object 2053h - measured final position

Index	Name	Type	Range	Default	Unit	Map	Attributes
2053h	**measured final position**	INT32	----	----		Yes	ro

This object is used only if the position measurement is enabled using Bit 5 of "***input setting***" (see *"4.04.2 Object 201Bh – Input setting" pag. 50*). When one of the inputs is assigned to the "***Position measuring***" function, the position measured during the falling edge of this input is stored on the "***measured final position***" (***index 2053h)*** parameter. This position is then converted from the internal position reference of the Drive to the Caliper's reference by using the parameters "***Numerator***" and "***Feedconstant***" of the ***"Pos factor"*** (***index 6093h***), with the formula reported below:

$$final\_position = \frac{(Feed\_constant \times drive\_final\_position)}{(Numerator)}$$

If the "***measuring position mode***" is set to "***continuous***" (bit 7 of 201Bh = 0) then the final position will be costantly updated at every falling edge of the digital input.

Instead, if the "***measuring position mode***" is set to "***one shot***" (bit 7 of 201Bh = 1) then the final position will be saved only at the first falling edge of the digital input and it's not updated anymore, unless this function is switched off and on again.

4.04.6.c Object 2054h – measured delta position

Index	Name	Type	Range	Default	Unit	Map	Attributes
2054h	**measured delta position**	INT32	----	----		Yes	ro

This object is used only if the position measurement is enabled using Bit 5 of "***input setting***" (see *"4.04.2 Object 201Bh – Input setting" pag. 50*). it represents the difference between the "***measured final position***" (***index 2053h)*** and the "***measured initial position***" (***index 2052h***)parameter. This value is then converted from the internal position reference of the Drive to the Caliper's reference by using the parameters "***Numerator***" and "***Feedconstant***" of the ***"Pos factor"*** (***index 6093h***), with the formula reported below:

$$delta\_position = \frac{(Feed\_constant \times drive\_delta\_position)}{(Numerator)}$$

If the "***measuring position mode***" is set to "***continuous***" (bit 7 of 201Bh = 0) then the "***delta position***" will be costantly updated at every falling edge of the digital input.

Instead, if the "***measuring position mode***" is set to "***one shot***" (bit 7 of 201Bh = 1) then the "***delta position***" will be calculated only at the first falling edge of the digital input and it's not updated anymore, unless this function is switched off and on again.

4.04.7 Touch Probe

The ***Touch Probe*** function supported by the H.D.T.'s Drives is very similar to the "***position measuring function***" described in *"4.04.6 Position measuring function from input - setting" pag. 52.* The ***Touch Probe*** indeed allows the user to save the motor position read by the Drive at a certain moment. This moment will correspond to the rising or falling edge of a signal read by a digital input set by the user as the "reference" for the ***Touch Probe***. Our Drives can work with 2 ***Touch Probe*** at the same time, and each of them can save 2 positions, 1 for the rising edge and 1 for the falling edge (for a total of 4 positions). Only the fastest inputs can be used as reference for the ***Touch Probe***, these inputs differs depending on the Drive model:

» For the ***NTT*** the suitable inputs are the 6th and the 7th;

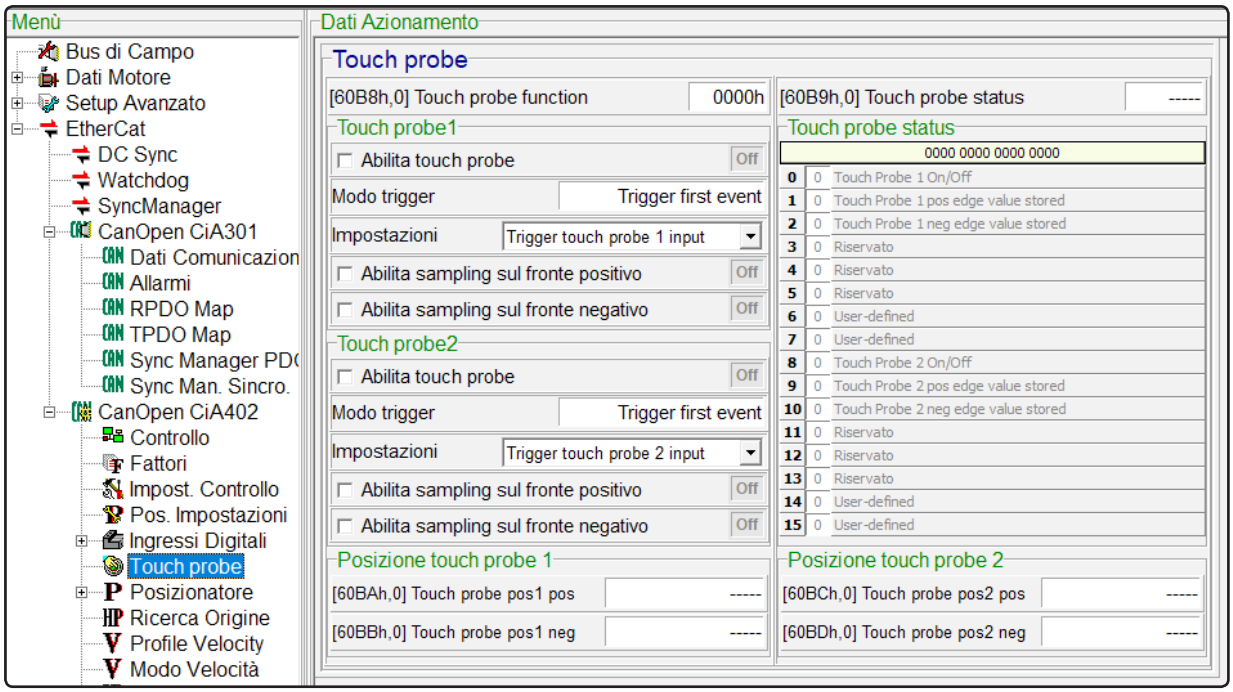
» For the ***DGFox*** the suitable inputs are the 1st and the 5th;

» For the ***TomCat*** the suitable inputs are the 1st and the 5th.

In the table below are reported the object used by the ***Touch Probe*** function:

Index	Name	Type	Map	Attributes
60B8h	Touch probe function	UINT16	Yes	rw
60B9h	Touch probe status	UINT16	Yes	rw
60BAh	Touch probe 1 pos. position	INT32	Yes	ro
60BBh	Touch probe 1 neg. position	INT32	Yes	ro
60BCh	Touch probe 2 pos. position	INT32	Yes	ro
60BDh	Touch probe 2 neg. position	INT32	Yes	ro

These parameters can be seen in the menu "***(EtherCat)-CanOpen CiA402-Touch Probe***" in which the following screen appears:



PRELIMINARY

4.04.7.a Object 60B8h – Touch Probe function

Index	Name	Type	Range	Default	Unit	Map	Attributes
60B8h	**Touch Probe function**	UINT16	0.. 65535	0		Yes	rw

This obect contains a series of flags, used to enable/disable the ***Touch Probe*** and to set how they work. These bits are described in the table below:

Touch Probe function - 60B8h		
Bit	**Value**	**Description**
0	0	Disable ***Touch Probe 1***
	1	Enable ***Touch Probe 1***
1	0	Trigger ***Touch Probe 1*** only on the first event
	1	Trigger ***Touch Probe 1*** continuous (not used)
2	0	Trigger on the input of the ***Touch Probe 1***
	1	Trigger of the ***Touch Probe 1*** on the zero signal of the encoder
3	-	not used
4	0	disable sampling on the rising edge of ***Touch Probe 1***
	1	enable sampling on the rising edge of ***Touch Probe 1***
5	0	disable sampling on the falling edge of ***Touch Probe 1***
	1	enable sampling on the falling edge of ***Touch Probe 1***
6,7	-	not used
8	0	Disable ***Touch Probe 2***
	1	Enable ***Touch Probe 2***
9	0	Trigger ***Touch Probe 2*** only on the first event
	1	Trigger ***Touch Probe 2*** continuous (not used)
10	0	Trigger on the input of the ***Touch Probe 2***
	1	Trigger of the ***Touch Probe 2*** on the zero signal of the encoder
11	-	not used
12	0	disable sampling on the rising edge of ***Touch Probe 2***
	1	enable sampling on the rising edge of ***Touch Probe 2***
13	0	disable sampling on the falling edge of ***Touch Probe 2***
	1	enable sampling on the falling edge of ***Touch Probe 2***
14,15	-	not used

4.04.7.b Object 60B9h – Touch Probe status

Index	Name	Type	Range	Default	Unit	Map	Attributes
60B9h	**Touch Probe status**	UINT16	0.. 65535	0		Yes	rw

This object reports the ***Touch Probe*** status. In the table below the bits of this object are described:

Touch Probe status - 60B9h		
Bit	**Value**	**Description**
0	0	***Touch Probe 1*** disabled
	1	***Touch Probe 1*** enabled
1	0	There is no position read on the rising edge of the ***Touch Probe 1*** in memory
	1	There is a position read on the rising edge of the ***Touch Probe 1*** in memory
2	0	There is no position read on the falling edge of the ***Touch Probe 1*** in memory
	1	There is a position read on the falling edge of the ***Touch Probe 1*** in memory
3...5	0	Reserved
6,7	-	User defined
8	0	***Touch Probe 2*** disabled
	1	***Touch Probe 2*** enabled
9	0	There is no position read on the rising edge of the ***Touch Probe 2*** in memory
	1	There is a position read on the rising edge of the ***Touch Probe 2*** in memory
10	0	There is no position read on the falling edge of the ***Touch Probe 2*** in memory
	1	There is a position read on the falling edge of the ***Touch Probe 2*** in memory
11...13	0	Reserved
14,15	-	User defined

4.04.7.c Object 60BAh – Touch Probe 1 pos. position

Index	Name	Type	Range	Default	Unit	Map	Attributes
60BAh	**Touch Probe 1 pos. position**	INT32	±7FFFFFFFh	---		Yes	ro

In this object is saved the motor position acquired on the rising edge of the ***Touch Probe 1***.

4.04.7.d Object 60BBh – Touch Probe 1 neg. position

Index	Name	Type	Range	Default	Unit	Map	Attributes
60BBh	**Touch Probe 1 neg. position**	INT32	±7FFFFFFFh	---		Yes	ro

In this object is saved the motor position acquired on the falling edge of the ***Touch Probe 1***.

4.04.7.e Object 60BCh – Touch Probe 2 pos. position

Index	Name	Type	Range	Default	Unit	Map	Attributes
60BCh	**Touch Probe 2 pos. position**	INT32	±7FFFFFFFh	---		Yes	ro

In this object is saved the motor position acquired on the rising edge of the ***Touch Probe 2***.

4.04.7.f Object 60BDh – Touch Probe 2 neg. position

Index	Name	Type	Range	Default	Unit	Map	Attributes
60BDh	**Touch Probe 2 neg. position**	INT32	±7FFFFFFFh	---		Yes	ro

In this object is saved the motor position acquired on the falling edge of the ***Touch Probe 2***.

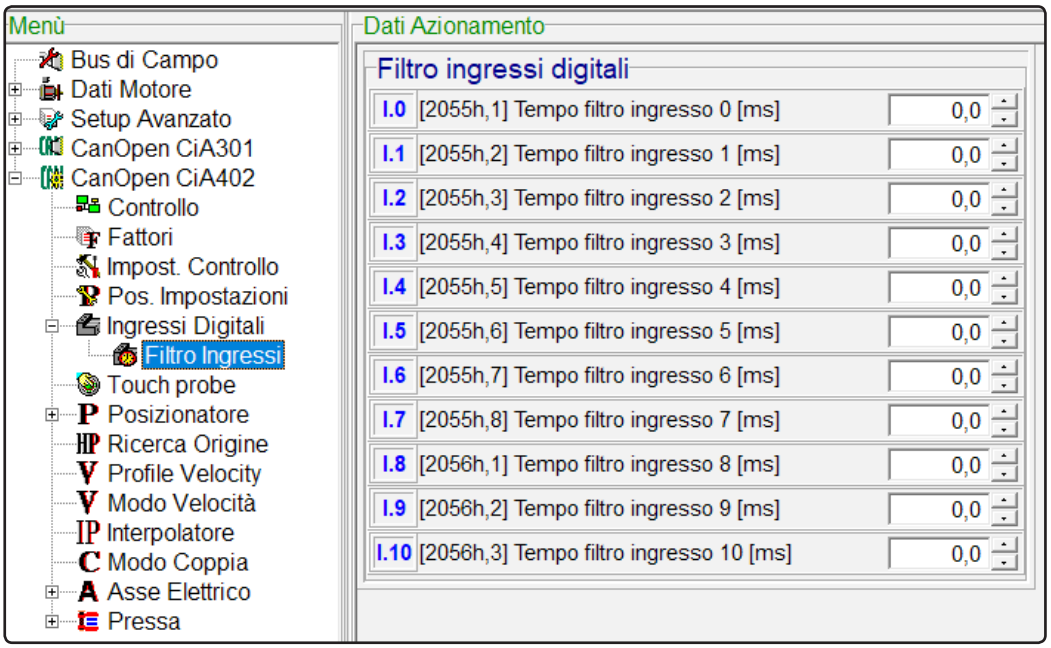
4.04.8 Objects 2055h and 2056– Inputs filter

Index	sub-index	Name	Type	Range	Default	Unit	Map	Attributes
2055h		Inputs filter	ARRAY					
	0	number of input	UINT8	8	8	----	No	ro
	1	Filter time input 0	UINT16	0..60000	0	ms/10	No	rw
	2	Filter time input 1	UINT16	0..60000	0	ms/10	No	rw
	3	Filter time input 2	UINT16	0..60000	0	ms/10	No	rw
	4	Filter time input 3	UINT16	0..60000	0	ms/10	No	rw
	5	Filter time input 4	UINT16	0..60000	0	ms/10	No	rw
	6	Filter time input 5	UINT16	0..60000	0	ms/10	No	rw
	7	Filter time input 6	UINT16	0..60000	0	ms/10	No	rw
	8	Filter time input 7	UINT16	0..60000	0	ms/10	No	rw

Index	sub-index	Name	Type	Range	Default	Unit	Map	Attributes
2056h		Inputs filter	ARRAY					
	0	number of input	UINT8	3	3	----	No	ro
	1	Filter time input 8	UINT16	0..60000	0	ms/10	No	rw
	2	Filter time input 9	UINT16	0..60000	0	ms/10	No	rw
	3	Filter time input 10	UINT16	0..60000	0	ms/10	No	rw

These objects are two array (for a total of 11 terms) of 16 bit integers accessible via their ***sub-index***, used to set a filter for every input. More specifically, each term of this arrays contains a time for every input, that works as a "threshold": if an input receives a signal with a different logic value it will not switch istantly, instead it will wait a period equals to the time assigned to it. If the signal last longer, the input will switch consequently, otherwise the signal will be ignored and considered as a disturbance.

These parameters can be set in the menu "***(EtherCat)-CanOpen CiA402-Digital input- Filter inputs***" in which the following screen appears:



4.04.9 Object 2076h – Canopen input mode

Index	Name	Type	Range	Default	Unit	Map	Attributes
2076h	Canopen input mode	UINT16	0.. FFFFh	Fh		No	rw

This parameter contains a series of flags used to set the inputs, as described in the table below:

Bit	Description	
0	**Contact type input 0**	
	Value	**Description**
	0	Input 0 is considered active when the voltage on its input is 0 [V]
	1	Input 0 is considered active when the voltage on its input is 24 [V]
1	**Contact type input 1**	
	Value	**Description**
	0	Input 1 is considered active when the voltage on its input is 0 [V]
	1	Input 1 is considered active when the voltage on its input is 24 [V]
2	**Contact type input 6**	
	Value	**Description**
	0	Input 6 is considered active when the voltage on its input is 0 [V]
	1	Input 6 is considered active when the voltage on its input is 24 [V]
3	**Contact type input 7**	
	Value	**Description**
	0	Input 7 is considered active when the voltage on its input is 0 [V]
	1	Input 7 is considered active when the voltage on its input is 24 [V]
4	**Input mode**	
	Value	**Description**
	0	Hardware input
	1	Logic input

PRELIMINARY

4.04.10 Objects 2077h,2078h,2079h,2080h e 2081h – Input setting

Index	Name	Type	Range	Default	Unit	Map	Attributes
2077h	Input setting 0	UINT8	0..6	0		No	rw
2078h	Input setting 1	UINT8	0..6	1		No	rw
2079h	Input setting 5	UINT8	0..6	2		No	rw
2080h	Input setting 6	UINT8	0..6	0		No	rw
2081h	Input setting 7	UINT8	0..6	0		No	rw

The objects described in this paragraph are used to assign to every digital input a specific function, depending on the value set here. These functions are described in the table below:

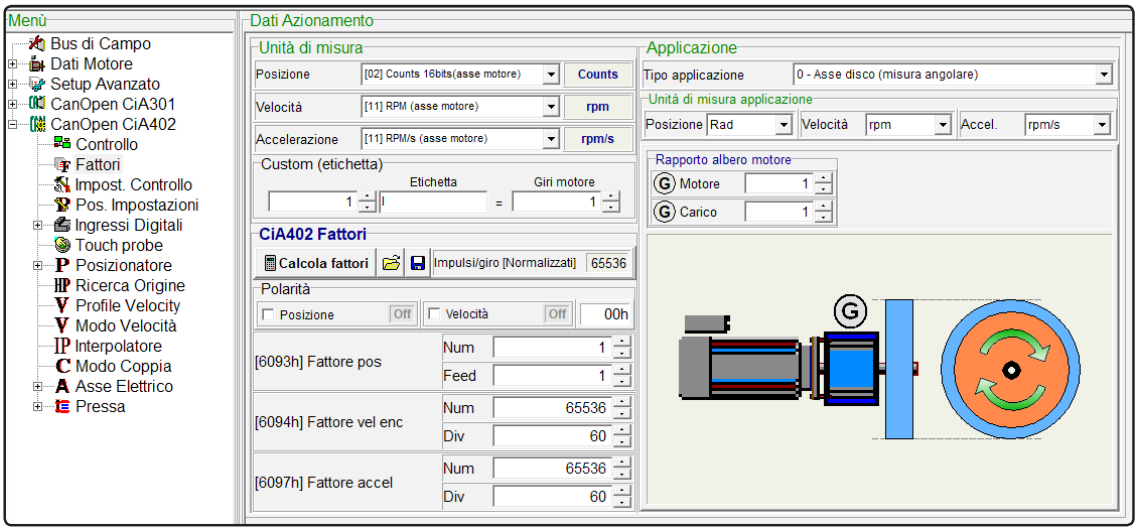
Value	Input setting	Description
0	Null off	No function selected
1	Position on Input X	When a rising edge occurs on input X (to which was assigned this function) the relative positioning set is executed (see *"4.07.11 Relative positioning launched by input X" pag. 76*)
2	Measuring position	Input used for the position measurement function (see *"4.04.6 Position measuring function from input - setting" pag. 52)*
3	Touch Probe 1	This input used to "trigger" the ***Touch Probe 1*** (see *"4.04.7 Touch Probe" pag. 55*). this function can be assigned only to the two fastest inputs
4	Touch Probe 2	This input used to "trigger" the ***Touch Probe 2*** (see *"4.04.7 Touch Probe" pag. 55*). this function can be assigned only to the two fastest inputs
5	Check motor brake	This input is connected with the brake's winding; this function is used for safety, to verify if the brake is really active or not
6	Fieldbus/IO selection	This function is used to bypass the Fieldbus and directly control the Drive in IO

4.05 Conversion factors

In this paragraph are reported the conversion factors used by the Drive to convert the unit of measure of the position, speed and acceleration variables.

Index	Name	Type	Map	Attributes
604Ch	dim. Factor	(ARRAY) UINT32	No	rw
6089h	Position notation index	INT8	No	rw
608Ah	Position dimension index	INT8	No	rw
608Bh	Velocity notation index	INT8	No	rw
608Ch	Velocity dimension index	INT8	No	rw
608Dh	Acceleration notation index	INT8	No	rw
608Eh	Acceleration dimension index	INT8	No	rw
6093h	Pos. factor	(ARRAY) UINT32	No	rw
6094h	Vel. encoder factor	(ARRAY) UINT32	No	rw
6097h	Accel. factor	(ARRAY) UINT32	No	rw
607Eh	Polarity	UINT8	No	rw

These parameters can be set in the menu "***(EtherCat)-CanOpen CiA402-Factors***" in which the following screen appears:



4.05.1 Object 604Ch - Dim. factor

Index	sub-index	Name	Type	Range	Default	Unit	Map	Attributes
604Ch		Dim. factor	ARRAY					
	0	num. of terms	UINT8	2	2		No	ro
	1	Numerator	INT32	± 7FFFFFFFh	1		No	rw
	2	Denominator	INT32	± 7FFFFFFFh	1		No	rw

The ***Dim. factor*** is composed by two 32 bit-values, the "***Numerator***" which multiplies the "***vl target velocity*** " ***(index 6042h)*** and the "***Denominator***" which divides the obtained result. These operation are necessary to convert the "***vl target velocity*** " in [rpm], then the converted result is used as input for the speed ramp generator.

$$target\_velocity[rpm] = \frac{(Numerator \times target\_velocity[counts])}{(Denominator)}$$

PRELIMINARY

4.05.2 Object 6089h - Position notation index

Index	Name	Type	Range	Default	Unit	Map	Attributes
6089h	Position notation index	INT8	0	0	--	No	rw

This parameter is the order of magnitude of the position's unit of measure. For our Drives this value is fixed to 0; in order to change the unit's order of magnitude you can use the "***Position factor***" (***index 6093h***).

4.05.3 Object 608Ah - Position dimension index

Index	Name	Type	Range	Default	Unit	Map	Attributes
608Ah	Position dimension index	INT8	ACh	ACh	--	No	rw

Position dimension index defines the position unit of measure. For our Drives this parameter is fixed to ACh, which means "pulses"; to modify this unit of measure you have to set the "***Pos. factor***" (***index 6093h***). The number of pulses for a single turn of the position sensor (defined with the "***Pulses/revolution***" (***index 2032h***) parameter) are internally converted in 65536 pulses per turn; this conversion will be used later by the applications that control the motor position, as the ***Interpolator*** and the ***Positioner***.

4.05.4 Object 608Bh - Velocity notation index

Index	Name	Type	Range	Default	Unit	Map	Attributes
608Bh	Velocity notation index	INT8	0	0	--	No	rw

This parameter is the order of magnitude of the speed's unit of measure. For our Drives this value is fixed to 0; in order to change the order of magnitude of this unit you can use the "***Vel. encoder factor***" (***index 6094h***).

4.05.5 Object 608Ch - Velocity dimension index

Index	Name	Type	Range	Default	Unit	Map	Attributes
608Ch	Velocity dimension index	INT8	A4h	A4h	--	No	rw

Velocity dimension index defines the speed unit of measure. For our Drives this parameter is fixed to A4h, which means [rpm]; to modify this unit of measure you have to set the "***Vel. encoder factor***" (***index 6094h***).

4.05.6 Object 608Dh - Acceleration notation index

Index	Name	Type	Range	Default	Unit	Map	Attributes
608Dh	Acceleration notation index	INT8	0	0	--	No	rw

This parameter is the order of magnitude of the acceleration's unit of measure. For our Drives this value is fixed to 0; in order to change the order of magnitude of this unit you can use the "***Accel. factor***" (***index 6097h***).

4.05.7 Object 608Eh - Acceleration dimension index

Index	Name	Type	Range	Default	Unit	Map	Attributes
608Eh	Acceleration dimension index	INT8	A3h	A3h	--	No	rw

Acceleration dimension index defines the acceleration unit of measure. For our Drives this parameter is fixed to A3h, which means [rpm/s]; to modify this unit of measure you have to set the "***Accel. factor***" (***index 6097h***).

PRELIMINARY

4.05.8 Object 6093h - Pos. factor

Index	sub-index	Name	Type	Range	Default	Unit	Map	Attributes
6093h		Pos. factor	ARRAY					
	0	num. of terms	UINT8	2	2		No	ro
	1	Numerator	UINT32	0..FFFFFFFFh	65536		No	rw
	2	Feed costant	UINT32	1..FFFFFFFFh	65536		No	rw

The "***Pos. factor***" is composed by two 32 bit-terms, the "***Numerator***" and the "***Feed costant***", used to convert the target position to the internal unit of measure of the Drive for the positions. This conversion is done by multiplying the "***target position***" (***index 607Ah***) by the ***Numerator*** and then dividing the result for the ***Feed constant***.

$$Drive\_ref\_position = \frac{(Numerator \times ref\_position)}{(Feedconstant)}$$

In practice the ***Numerator*** corresponds to the number of pulses per turn, that is 65536. The ***Feed costant*** instead represents the space covered during a motor turn, expressed with the same unit of measure of the "***target position***" (***index 607Ah***). Therefore by changing the values of the ***Numerator*** and ***Feed constant*** you affect on how are calculated the internal target position and the position unit of measure.

4.05.9 Object 6094h - Vel. encoder factor

Index	sub-index	Name	Type	Range	Default	Unit	Map	Attributes
6094h		Vel. encoder factor	ARRAY					
	0	num. of terms	UINT8	2	2		No	ro
	1	Numerator	UINT32	0..FFFFFFFFh	65536	--	No	rw
	2	Divisor	UINT32	1..FFFFFFFFh	60	--	No	rw

The "***vel. encoder factor***" converts the values of the "***profile velocity***" (***index 6081h***) and the "***target velocity***" (***index 60FFh***) in the internal unit of measure of the Drive, i.e. pulses per second. The formula used for the conversion is reported below:

$$Profile\_velocity[counts/s] = \frac{(Numerator \times Profile\_velocity[rpm])}{Divisor}$$

For our Drive the unit of measure of the "***profile velocity***" is set by default in [rpm]; the conversion to pulses per second is done by setting the ***Numerator*** to 65536 and the ***Divisor*** to 60.

4.05.10 Object 6097h - Acceleration factor

Index	sub-index	Name	Type	Range	Default	Unit	Map	Attributes
6097h		acceleration factor	ARRAY					
	0	num. of terms	UINT8	2	2		No	ro
	1	Numerator	UINT32	0..FFFFFFFFh	65536	--	No	rw
	2	Divisor	UINT32	1..FFFFFFFFh	60	--	No	rw

The "***acceleration factor***" converts the values of the "***profile acceleration***" (***index 6083h***) and the "***profile deceleration***" (***index 6084h***) in the internal unit of measure of the Drive, i.e. pulses per s^2. The formula used for the conversion is reported below:

$$Profile\_acceleration\ [counts/s^2] = \frac{(Numerator \times Profile\_acceleration[\frac{rpm}{s}])}{Divisor}$$

For our Drive the unit of measure of the acceleration and the deceleration is set by default in [rpm/s]; the conversion to pulses per s^2 is done by setting the ***Numerator*** to 65536 and the ***Divisor*** to 60.

4.05.11 Object 607Eh - Polarity

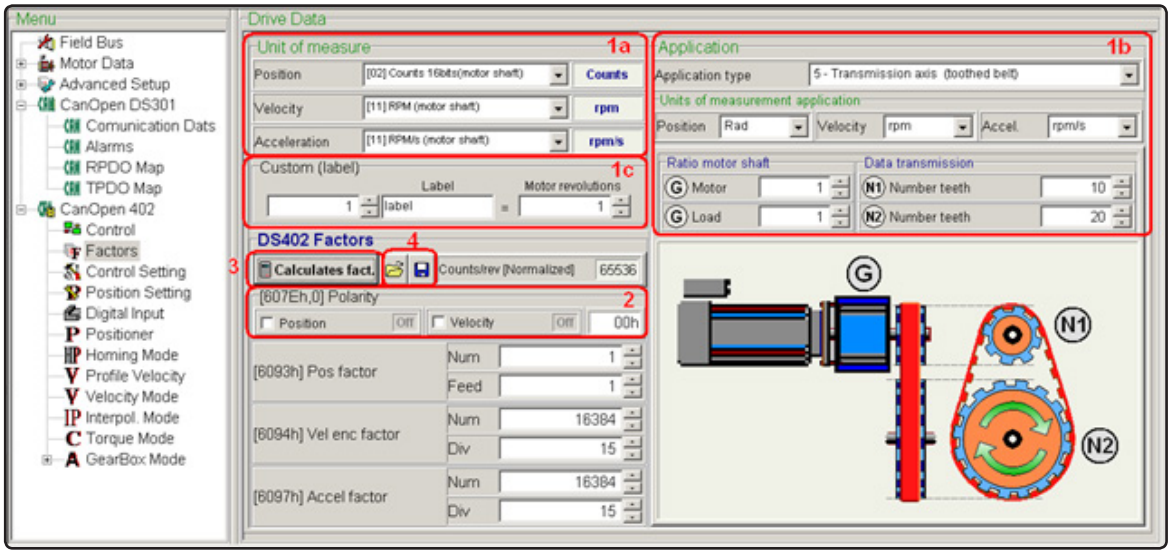
Index	Name	Type	Range	Default	Unit	Map	Attributes
607Eh	Polarity	UINT8	0..FF	0	--	No	rw

Some bits of the "***Polarity***" parameter are associated with variables. If one of this bit is equal to 0, the value of the associated variable remains unchanged; if the bit is equal to 1, the value's sign is reversed.

Bit	Description
7	***Position polarity***, referred to the "***target position***" (***index 607Ah***)
6	***Velocity Polarity***, referred to the "***target velocity***" (***index 60FFh***)
0..5	reserved

4.05.12 Utility for the factors setting

The Caliper software support an utility that eases the calculation of the position,speed and acceleration factors used by the "***Positioner***", "***Profile Velocity***", "***Homing***" and "***Interpolator***" control type.



By selecting "***Factors***" in the menu tree, a window appears on the right which contains several variables. The values related to the "***Pos. factor*** ", "***vel. encoder factor***" and "***acceleration factor***" could be set up manually by activating the access to these parameters with the "***local control***" button, or they could be calculated by following these steps:

1. By the use of the drop-down menu in the ***1a*** zone (see figure), you can choose the unit of measure of the position, speed and acceleration: if you select the "***Custom (Application)***" item, you must set the parameters in the ***1b (Application)*** zone for the factors calculation; if you select the "***Custom (Label)***" item, you must set the parameters in the ***1c*** zone; otherwise, if you choose another item, no further parameters need to be set.
2. The position's and speed's polarities can be set by pressing the "***Local control***" button, that allows you to directly modify these parameters. In any case, the "***polarity***" parameters don't affect the factors calculation.
3. By pressing the "***Calculates factors*** " button, the factors are calculated and shown on a window, to send them to the Drive you have to press the "***Updates***" button.
4. If you decide to use the "***Custom (Application)***" item or the "***Custom (Label)***" item, you can save the parameters used for the factors on a file or you can upload them from a file by using two other buttons avaiable in the window.

PRELIMINARY

4.05.12.a Custom (label)

The "***Custom (label)***" allows the user to create a custom unit of measure used by the final application. The custom unit of measure is defined by setting two parameters:

» For a rotative motor you must set in the "***motor revolutions***" box the number of motor revolutions which will correspond (with a proportion) to a certain number of the custom unit of measure, set on the first box on the left.

» For a linear motor you must set in the "***Pole pitches***" box the number of motor pole pitches which will correspond (with a proportion) to a certain number of the custom unit of measure, set on the first box on the left.

The "***label***" box is used to assign a string as a label for the custom unit of measure.

	U.M. angular	U.M. linear	U.M. standard
Position	radiants, degrees	cm, mm, µm, nm, inches, mils	counts
Speed	radiants/s, degrees/s	cm/s, mm/s, µm/s, nm/s, inches/s, mils/s	counts/s, RPM, RPS
Acceleration	radiants/s^2, degrees/s^2	cm/s^2, mm/s^2, µm/s^2, nm/s^2, inches/s^2, mils/s^2	counts/s^2, RPM/s, RPS/s

Note: There are other unit of measure that you can choose from the same drop-down menu, which are tenths, hundredths and thousandths of some of the main unit of measure.

4.05.12.b Factors calculation for a rotative motor with the "Custom (Application)"

The ***Custom applications*** supported by the Drive for a rotative motor are reported below:

1. ***Disk axis (angular measurement)***: This application uses an angular unit of measure referred directly to the load, which is treated as a rotating disk at the end of an eventual gearbox. The parameters to be set are:
   - the unit of measure
   - the gearbox ratio
2. ***Disk axis (linear measurement)***: This application uses a linear unit of measure referred directly to the load, which is treated as a rotating disk at the end of an eventual gearbox. The parameters to be set are:
   - the unit of measure
   - the gearbox ratio
   - the disk diameter
3. ***Axis conveyor belt (flat belts)***: This application uses a linear unit of measure referred directly to the load, which is treated as a belt at the end of an eventual gearbox. The parameters to be set are:
   - the unit of measure
   - the gearbox ratio
   - the pulley diameter
4. ***Axis conveyor belt (toothed belts)***: This application uses a linear unit of measure referred directly to the load, which is treated as a toothed belt at the end of an eventual gearbox. The parameters to be set are:
   - the unit of measure
   - the gearbox ratio
   - number of teeth and length of the tooth pitch
5. ***Transmission axis (flat belts)***: This application uses an angular unit of measure referred directly to the load, which is treated as a pulley moved by a flat belt, at the end of an eventual gearbox. The parameters to be set are:
   - the unit of measure
   - the gearbox ratio
   - diameter of the "driving" pulley and of the "driven" pulley
6. ***Transmission axis (toothed belts)***: This application uses an angular unit of measure referred directly to the load, which is treated as a pulley moved by a toothed belt, at the end of an eventual gearbox. The parameters to be set are:
   - the unit of measure
   - the gearbox ratio
   - number of teeth of the "driving" pulley and of the "driven" pulley
7. ***Axis with worm***: This application uses a linear unit of measure referred directly to the load, which is treated as a worm gear, at the end of an eventual gearbox. The parameters to be set are:
   - the unit of measure
   - the gearbox ratio
   - the screw pitch of the worm gear
8. ***Axis motor with hollow shaft***: This application uses a linear unit of measure referred directly to the load, which is treated as a motor with an hollow shaft. The parameters to be set are:
   - the unit of measure
   - the screw pitch of the hollow shaft

4.05.12.c Factors calculation for a linear motor with the "Custom (Application)"

There is only one ***Custom application*** supported by the Drive for a linear motor, and the calculation of its factors are equals to the one described for the rotative motor.

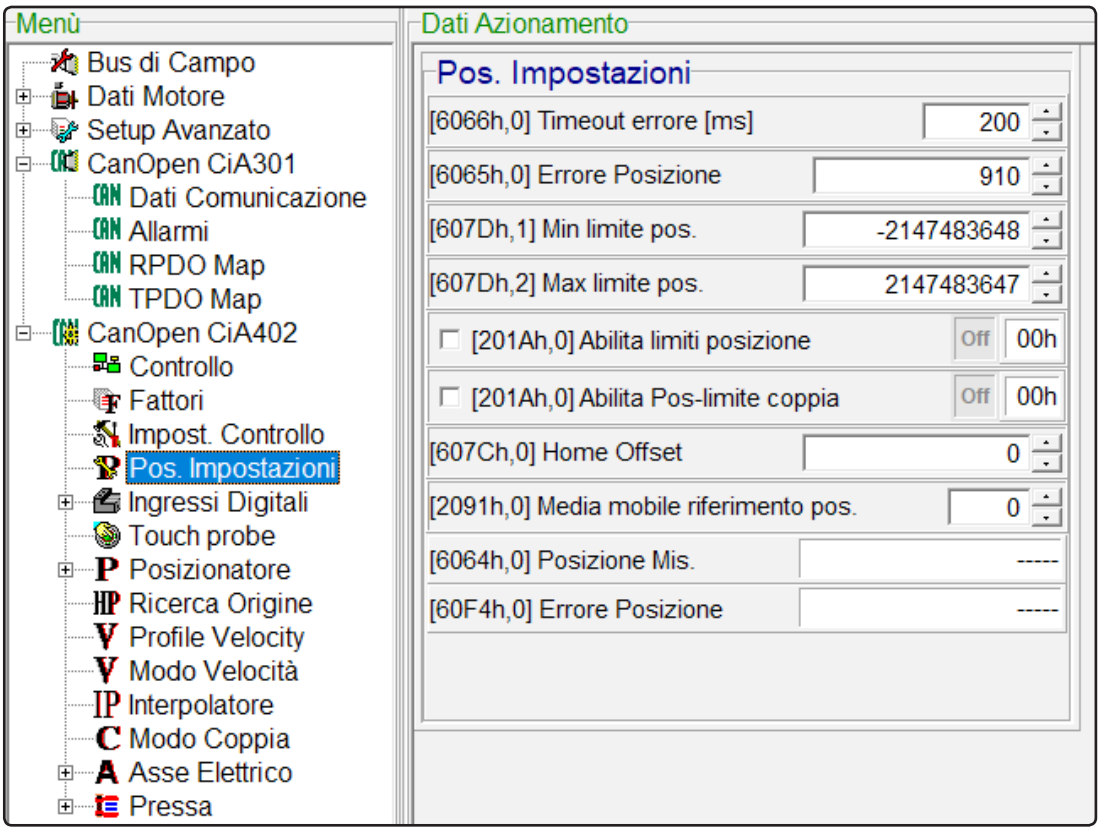
4.06 Parameters of the position setting

In this paragraph are reported the common parameters used for the position control by both the ***Positioner*** and the ***Interpolator***.

Index	Name	Type	Map	Attributes
6062h	Pos. demand value	INT32	Yes	ro
6063h	Pos. demand value*	INT32	Yes	ro
6064h	Measured position	INT32	Yes	ro
6065h	Position error window	UINT32	No	rw
6066h	Timeout error	UINT16	No	rw
607Ch	Home offset	INT32	Yes	rw
607Dh	position limits	(ARRAY) UINT32	No	rw
60F4h	position error	INT32	Yes	ro
60FCh	Pos. demand value*	INT32	Yes	ro
201Ah	position- flag	UINT8	No	rw
2091h	Moving average reference pos.	UINT16	No	rw

(*) The parameters marked with an asterisk are converted to the internal unit of measure of the Drive.

These parameters can be set in the menu "***(EtherCat)-CanOpen CiA402-Position setting***" in which the following screen appears:



4.06.1 Object 6062h - Pos. demand value

Index	Name	Type	Range	Default	Unit	Map	Attributes
6062h	Pos. demand value	INT32	±7FFFFFFFh	----		Yes	ro

This parameter represents the position reference given as an input to the position loop. This value is converted to the same unit of measure of the "***target position***" (***index 607Ah***) by using the "***Pos. Factor***" (***index 6093h***) as showed in the formula below:

$$ref\_position = \frac{(Feedconstant \times Drive\_ref\_position)}{(Numerator)}$$

4.06.2 Object 6063h - Pos. demand value*

Index	Name	Type	Range	Default	Unit	Map	Attributes
6063h	Pos. demand value*	INT32	----	----		Yes	ro

This is a read-only variable that shows the value of the measured position expressed with the internal unit of measure of the Drive, which convert one motor revolution to 65536 pulses

4.06.3 Object 6064h - Measured position

Index	Name	Type	Range	Default	Unit	Map	Attributes
6064h	Measured position	INT32	----	----		Yes	ro

This parameter shows the measured position of the motor converted to the same unit of measure of the "***target position***" (***index 607Ah***) by using the "***Pos. Factor***" (***index 6093h***) as showed in the formula below:

$$measured\_position = \frac{(Feedconstant \times Drive\_measured\_position)}{(Numerator)}$$

The default values of the ***Feed constant*** and ***Numerator*** of the "***Pos. factor***" are both 65536 , therefore in this case the value of the measured position corresponds to the one reported on index 6063h.

4.06.4 Object 6065h - Position error window

Index	Name	Type	Range	Default	Unit	Map	Attributes
6065h	Position error window	UINT32	0..FFFFFFFFh	910		No	rw

This parameter is used to set the maximum position error which can be tolerated by an application that controls the position. The unit of measure of this parameter is the same of the "***target position***" (***index 607Ah***). This value is converted by the Drive to its internal unit of measure using the "***Pos. Factor***" (***index 6093h***).

The default values of the ***Feed constant*** and ***Numerator*** of the "***Pos. factor***" are both 65536 ,which corresponds to one motor turn, therefore the default error of 910 pulses corresponds to 5°. Internally the maximum position error is limited to 32768 pulses, that correspond to 180°.

4.06.5 Object 6066h - Timeout error

Index	Name	Type	Range	Default	Unit	Map	Attributes
6066h	Timeout error	UINT16	1..10000	200	ms	No	rw

On this parameter is set the maximum tolerated time period in which the position error can surpass the "***position error window***" (***index 6065h***). If this period is surpassed the bit 13 of the ***statusword*** is set to 1 and the Drive switch to the ***Fault*** state.

4.06.6 Object 607Ch - Home offset

Index	Name	Type	Range	Default	Unit	Map	Attributes
607Ch	Home offset	INT32	±7FFFFFFFh	0		Yes	rw

The "***Home offset*** " is the difference between the zero position of the application and the zero position of the Drive found with the ***homing*** procedure. This value affects the measured position reported on the parameter on index 6064h, as shown in the following formula:

$$measured\_position = sensor\_position + home\_offset$$

Inside the Drive each motor turn is converted to 65536 pulses, which correspond to 16 bit, but the position variable is a 32 bit-value with sign, therefore in addition to the fraction of motor revolution this parameter can count up to ±32767 motor turns.

The position measued by the sensor is reset to 0 after every power on or when is executed an ***homing*** procedure. In order to set to zero the "***Measured position***" (***index 6064h***) you have to write on "***Home offset***" the measured position with the sign reversed; you can see the value to write by initially putting the "***Home offset***" to 0 and reading the "***Measured position***" (***index 6064h***). If the bit 7 of the "***Polarity***" (***index 607Eh***) is active the value necessary to put to 0 the "***Measured position***" (***index 6064h***) is still the position measured by the sensor, but in this case it is not necessary to invert its sign.

PRELIMINARY

4.06.7 Object 607Dh - Position limits

Index	sub-index	Name	Type	Range	Default	Unit	Map	Attributes
607Dh		Position limits	ARRAY					
	0	num. of terms	UINT8	2	2		No	ro
	1	Min. limit	INT32	0x80000000.. 7FFFFFFFh	0x80000000	--	Yes	rw
	2	Max. limit	INT32	0x80000000.. 7FFFFFFFh	+7FFFFFFFh	--	Yes	rw

The "***Position limits***" array contains 2 parameters used to limit the absolute position of the motor when the Drive is working with the ***Positioner*** or the ***Interpolator*** control, the value of sub-index 1 represents the minimum limit, the value of sub-index 2 represents the maximum limit. They have the same unit of measure of the "***target position***" (***index 607Ah***). This limits are enabled by putting to 1 the bit 0 of "***Position-Flag***" (***index 201Ah***). If one of the limits is reached by the position reference (expressed in absolute value with respect to the origin) then the bit 11 (i.e. "***internal limit active***") of the ***statusword*** is turned on.

4.06.8 Object 60F4h - Position error

Index	Name	Type	Range	Default	Unit	Map	Attributes
60F4h	Position error	INT32	±7FFFFFFFh			Yes	ro

This parameter shows the position error, which is converted to the same unit of measure of the "***target position***" (***index 607Ah***) by using the "***Pos. Factor***" (***index 6093h***). The position error is converted from the internal unit of the Drive as shown in the following formula:

$$error\_position = \frac{(Feedconstant \times Drive\_error\_position)}{(Numerator)}$$

4.06.9 Object 60FCh - Pos. demand value*

Index	Name	Type	Range	Default	Unit	Map	Attributes
60FCh	Pos. demand value*	INT32	±7FFFFFFFh	0		Yes	ro

This parameter contains the reference position of the position loop, expressed in internal unit of the Drive.

4.06.10 Object 201Ah – Position-Flag

Index	Name	Type	Range	Default	Unit	Map	Attributes
201Ah	Position-flag	UINT8	0..FFh	0		No	rw

The bits of this object are used as flags to configure the ***Positioner***, as described in the table below:

Bit	Description	
0	**Enable/disable position limits**	
	Value	**Description**
	0	software position limits disabled
	1	software position limits enabled
1	**Enable/disable Torque limits**	
	Value	**Description**
	0	Torque limit disabled
	1	***Torque limit enabled***: when the Drive torque reaches the limit (because there is an obstacle on the path or because "***current limit***" (***index 2006h***) has been decreased) then the position profile is adapted to the measured position; when the Drive torque is no longer at its limit,then the execution of the position profile is resumed to reach the target set.
2	**Pos. rounding mode (valid when "Motion profile type" (6086h) is set to 3)**	
	Value	**Description**
	0	The position profile is rounded by using the "***position- jerk***" (***index 2011h***)
	1	The position profile is rounded by using the "***rounding time***" (***index 2021h***)
3	**Enable/disable positioning on input X**	
	Value	**Description**
	0	The ***positioning on input X*** is disabled.
	1	***positioning on input X*** enabled: Con questa impostazione viene eseguito uno spostamento relativo sul fronte di salita dell'ingresso X, lo spostamento e definito dal parametro "***Offset spostamento relativo input X" (index 2019h)***
4..7	Reserved	

PRELIMINARY

4.06.11 Object 2091h – Moving average ref. pos.

Index	Name	Type	Range	Default	Unit	Map	Attributes
2091h	Moving average ref. pos.	UINT16	0..30	0		No	Rw

The position control can use as a reference the average of the last tot. reference positions received from the Drive. This object is used to set how many terms to use for this moving average. If it's set to 0 the moving average will not be performed, and the control will directly use only the last received reference position.

Note: this is a custom variable, not included in ***DS-402***.

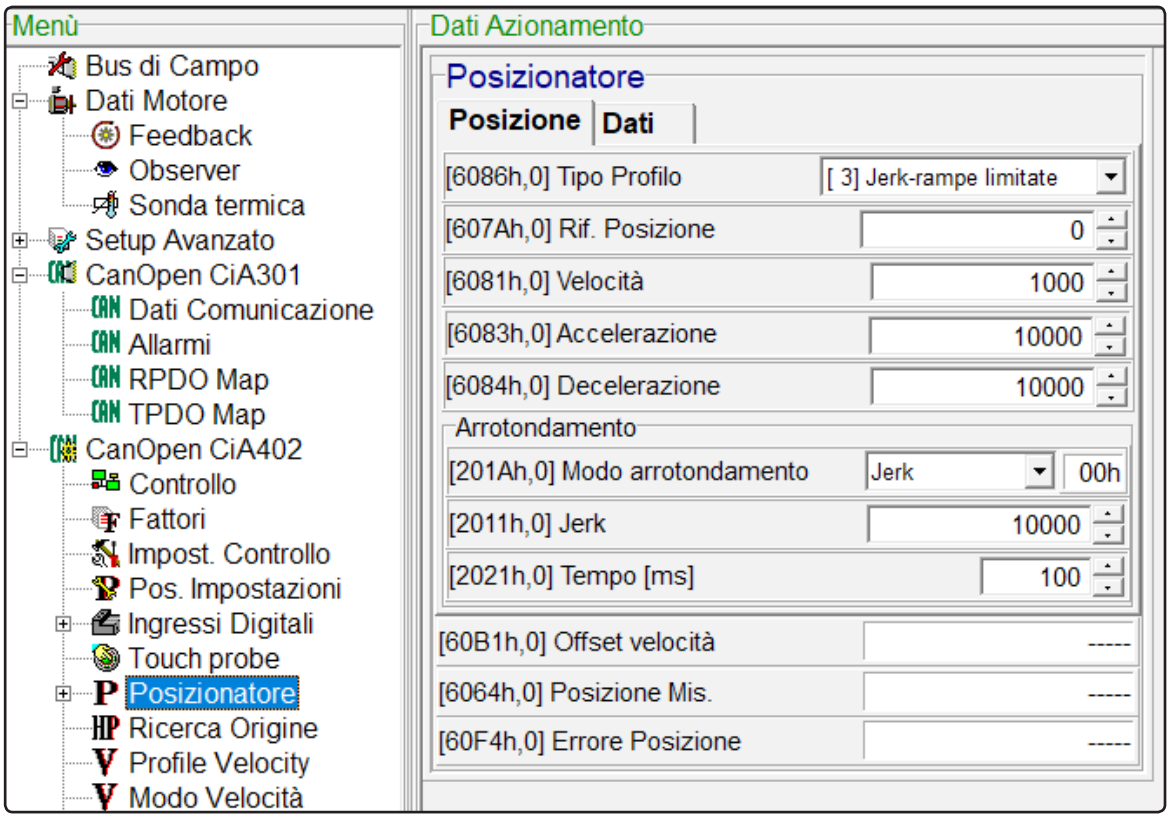
4.07 Profile Position Mode

In this section are reported the parameters used by the ***Positioner*** control type.

Index	Name	Type	Map	Attributes
607Ah	Target Position	INT32	Yes	rw
6081h	Profile velocity	UINT32	Yes	rw
6083h	Profile Acceleration	UINT32	Yes	rw
6084h	Profile Deceleration	UINT32	Yes	rw
6086h	Profile Type	INT16	No	rw
2011h	Profile Jerk	UINT32	No	rw
2021h	Rounding time	UINT16	No	rw

PRELIMINARY

These parameters can be seen in the Caliper menu "***(EtherCat-)CanOpen CiA402- Positioner***" in which the following screen appears:



4.07.1 Setting of the Controlword bits for the Positioner

Bit	Description	
0	***Switch on***	
1	***Enable voltage***	
2	***Quick stop***	
3	***Enable operation***	
4	***New set-point***	
	Value	**Description**
	0	The motor stand stills
	1	A new set-point is acquired, and the positioning is started
5	***Change set immediately***	
	Value	**Description**
	0	The positionings are executed in succession: once the current positioning is terminated, the new data required for the next positioning is accepted
	1	The values of the setpoints and the speeds are accepted and updated even with a positioning in course
6	***abs / rel***	
	Value	**Description**
	0	The new setpoint is an absolute value, defined with respect to the zero of the application
	1	The new setpoint is a relative value, defined with respect to the current setpoint
7	***Fault reset***	
8	***Halt***	
	Value	**Description**
	0	The positionings are executed normally
	1	The motor stops in ramp according to the "***profile deceleration***" (***index 6084h***)
10..9	***Reserved***	
15..11	***Manufacturer specific***	

4.07.2 Statusword bits values for the Positioner

Bit	Description
0	***Ready to switch on***
1	***Switched on***
2	***Operation enable***
3	***Fault***
4	***Voltage enabled***
5	***Quick stop***
6	***Switch on disabled***
7	***Warning***
8	***Manufacturer specific***
9	***Remote***

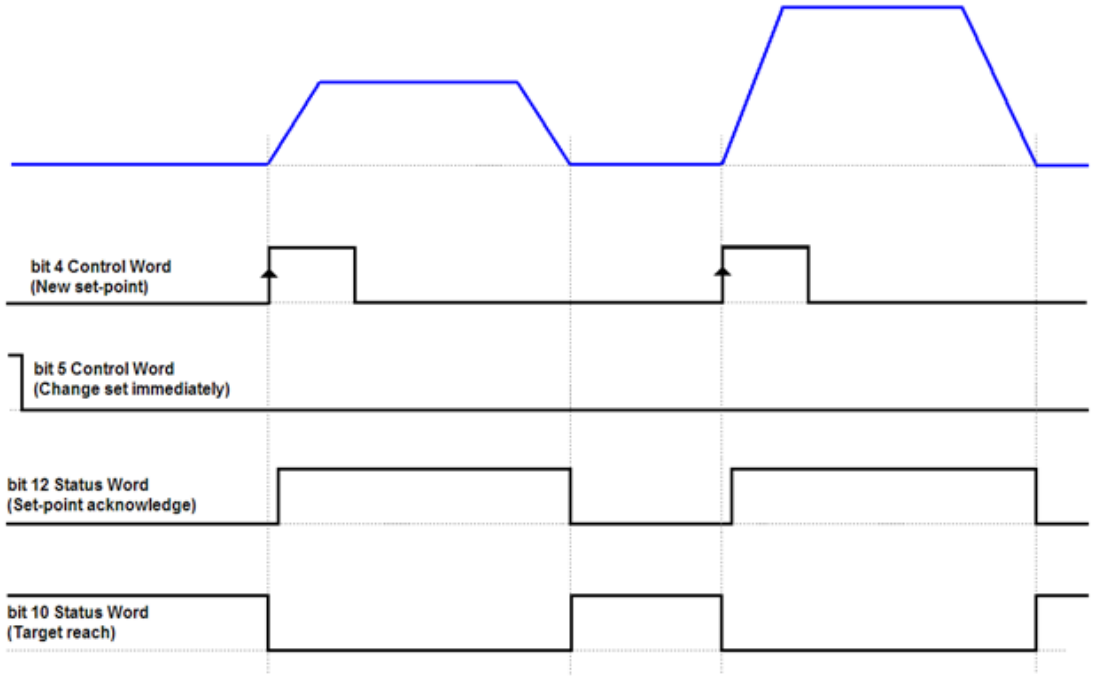
PRELIMINARY

Bit	Description		
10	***Target reached***		
	Value	**Controlword Halt Bit**	**Description**
	0	0	Target position not reached
		1	Deceleration phase of the motor
	1	0	Target position reached
		1	the motor stands still
11	***Internal limit active*** (enabled only when "***Enable Position limits***" -***index 201Ah***- is on)		
	Value	**Description**	
	0	The absolute value of the target position is not limited by "***Pos. limits***" (***index 607Dh***)	
	1	The absolute value of the target position is limited by "***Pos. limits***" (***index 607Dh***)	
12	***Set-point acknowledge***		
	Value	**Description**	
	0	Setpoint not yet acquired by the function that generates the position profile	
	1	Setpoint acquired by the function that generates the position profile	
13	***Following error***		
	Value	**Description**	
	0	No following error revealed	
	1	Following error revealed	
14	***Pos-Input X Raggiunto***		
	Value	**Description**	
	0	Relative positioning started by input X not reached.	
	1	It shows that the relative positioning set on "***Position-input X***" (***index 2019h***) has been completed. The function concerning this relative positioning is enabled by putting bit 3 of "***Position-Flag***" (***index 201Ah***) to 1, and it starts on the rising edge of the define input X	
15	***Position measuring*** (enabled by "***Input setting***" – ***index 201Bh***)		
	Value	**Description**	
	0	• ***(index 201Bh bit7=0) continuous mode:*** the position between the rising and falling edge of the chosen input has been calculated. • ***(index 201Bh bit7=1) one shot mode:*** the two values required to calculates the position between the edges of the chosen input have not yet been acquired.	
	1	• ***(index 201Bh bit7=0) continuous mode:*** the position between the rising and falling edge of the chosen input has not yet been calculated. • ***(index 201Bh bit7=1) one shot mode:*** the two values required to calculates the position between the edges of the chosen input have been acquired.	

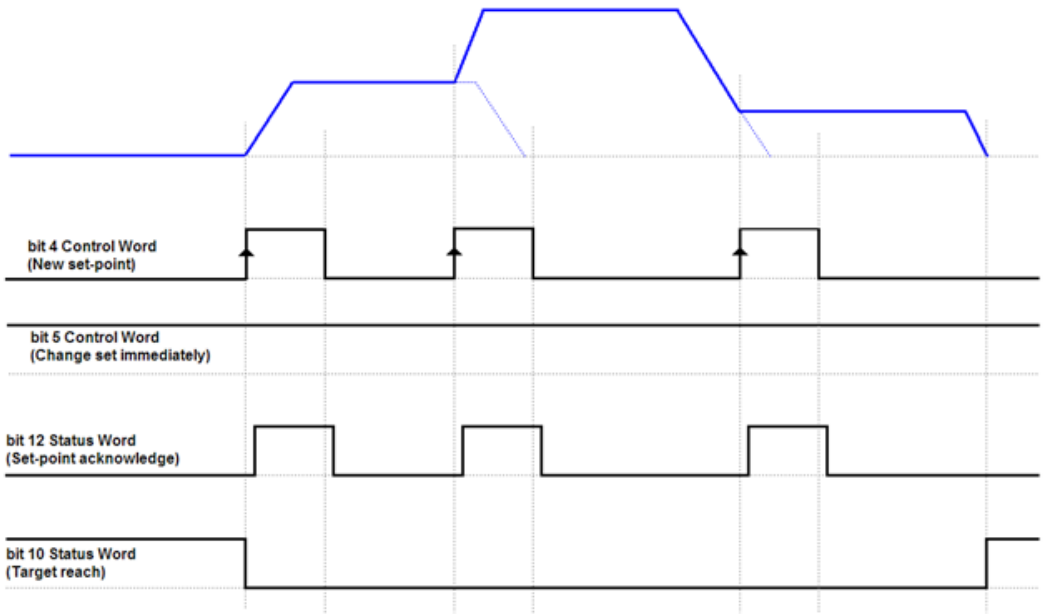
4.07.3 Setting of the bit 5 ("Change set immediately") of the Controlword

The bit 5 ("***Change set immediately***") of the ***Controlword*** affects how the reference positions are handled, when is set to 0 the ***Positioner*** works in "***Single Setpoint***" mode, when is set to 1 the ***Positioner*** works in "***Change set immediately***" mode.

In "***Single Setpoint***" mode the Drive executes the positionings one by one, the new "target position" is acquired only at the end of the current positioning, when the speed has returned to 0. The new setpoint will be acquired and executed only when bit 4 ("***New set-point***") of the ***controlword*** is set to 1 and bit 12 ("***set-point acknowledge***") of the ***statusword*** is set to 0. Right after the acknowledgement the bit 12 of the ***statusword*** is set to 1 and bit 10 ("***Target reach***") of the ***statusword*** is set to 0. When a positioning is terminated the bit 12 of the ***statusword*** is set again to 0 and bit 10 ("***Target reach***") of the ***statusword*** is set to 1. In "***Single Setpoint***" mode the "***profile type***" (***index 6086h***) can be set on "***Linear ramp*** (***trapezoidal profile***)" or on "***Jerk-limited ramp***".



Even in "***Change set immediately***" mode the positioning is started on the rising edge of bit 4 ("***New set-point***") of the ***controlword***, and bit 12 ("***set-point acknowledge***") of the ***statusword*** is set to 1 to show that a new setpoint has been acquired. When bit 4 of the ***controlword*** returns to 0 bit 12 of the ***statusword*** returns to 0 too, to show that the Drive is ready to acquire and execute a new positioning even if the current one has not terminated yet. The bit 10 ("***Target reach***") of the ***statusword*** works as in the previous case. If bit 6 ("***abs/rel***") of the ***controlword*** is set to 1, the new positioning will be relative with respect to the current position.



4.07.4 Object 607Ah - Target position

Index	Name	Type	Range	Default	Unit	Map	Attributes
607Ah	Target position	INT32	±7FFFFFFFh	0		Yes	rw

This parameter contains the position that must be reached. Its value is expressed in pulses[1], and is converted in internal units of the Drive by using the "***Pos. factor***" (***index 6093h***). The formula for the conversion is shown below:

$$Drive\_ref\_position = \frac{(Numerator \times ref\_position)}{(Feedconstant)}$$

(1): a motor revolution corresponds to 65536 pulses for the Drive.

4.07.5 Object 6081h - Profile velocity

Index	Name	Type	Range	Default	Unit	Map	Attributes
6081h	Profile velocity	UINT32	1..FFFFFFFFh	1000		Yes	rw

This parameter contains the maximum speed that could be reached during the positioning. Inside of the Drive it is expressed in pulses per second [pulses/s]. This parameter is converted in the internal unit of measure of the Drive by using the "***vel. encoder factor***" (***index 6094h***) array, as showed in the formula below:

$$Profile\_velocity[counts/s] = \frac{(Numerator \times Profile\_velocity[rpm])}{Divisor}$$

The unit of measure of the "***profile velocity***" is set by default in [rpm] for the Drive, the conversion to pulses per second is done by setting the ***Numerator*** to 65536 and the ***Divisor*** to 60.

4.07.6 Object 6083h - Profile acceleration

Index	Name	Type	Range	Default	Unit	Map	Attributes
6083h	Profile acceleration	UINT32	1..FFFFFFFFh	1000		Yes	rw

This parameter contains the acceleration used for the positioning.Inside of the Drive it is expressed in pulses per second squared [pulses/s^2]. This parameter is converted in the internal unit of measure of the Drive by using the "***acceleration factor***" (***index 6097h***) array, as showed in the formula below:

$$Profile\_acceleration\ [counts/s^2] = \frac{(Numerator \times Profile\_acceleration[\frac{rpm}{s}])}{Divisor}$$

By using the default values of the "***acceleration factor***", i.e. ***Numerator***=65536 and ***Divisor***=60, the ***Profile acceleration*** is converted in [rpm/s].

4.07.7 Object 6084h - Profile deceleration

Index	Name	Type	Range	Default	Unit	Map	Attributes
6084h	Profile deceleration	UINT32	1..FFFFFFFFh	1000		Yes	rw

This parameter contains the deceleration used for the positioning. Inside of the Drive it is expressed in pulses per second squared [pulses/s^2]. This parameter is converted in the internal unit of measure of the Drive by using the "***acceleration factor***" (***index 6097h***) array, as described for the "***Profile acceleration***" (***index 6083h***) parameter.

4.07.8 Object 6086h - Profile type

Index	Name	Type	Range		Default	Unit	Map	Attributes
6086h	Profile type	INT16	-1	Linear ramp with torque limit	3		No	rw
			0	Linear ramp				
			3	Jerk- limited ramp				

The "***Profile type"*** is used to choose the form of the position profile to follow during a positioning. This parameter cannot be modified while the Drive is in the "***Operational***" state.

Value	Motion profile type	Description
0	Linear ramp (trapezoidal profile)	• In this mode the position profile is generated by using constant accelerations, therefore the speed profile has a trapezoidal shape, and the position profile has a parabolic shape. • When bit 5 of the ***controlword*** ("***Change set immediately***") is set to 1, the control allows you to change on the fly the setpoint (***target position*** – ***index 607Ah***), and the position profile is updated when bit 4 of the ***controlword*** is switched from 0 to 1. The acceleration ramps (***profile acceleration*** - ***index 6083h***), deceleration ramps (***profile deceleration*** - ***index 6084h***) and the "rounding values" (***profile jerk***- ***index 2011h*** or r***ounding time-index 2021h***) are adjusted when the motor is standstill and a new positioning is starting. The speed instead (***profile velocity*** - ***index 6081h***) is immediately adjusted to the set value.
3	Jerk- limited ramp	• In this mode the accelerations and decelerations change with a trapezoidal shape, therefore the speed has a parabolic shape, and the position has a cubic shape. This profile type reduces the mechanical stress compared to a constant accelerations profile, and it also improves the precision during the deceleration phase, when the motor is approaching the setpoint. • When bit 5 of the ***controlword*** ("***Change set immediately***") is set to 1, the control allows you to change on the fly the setpoint (***target position*** – ***index 607Ah***), and the position profile is updated when bit 4 of the ***controlword*** is switched from 0 to 1. The acceleration ramps (***profile acceleration*** - ***index 6083h***), deceleration ramps (***profile deceleration*** - ***index 6084h***) and the "rounding values" (***profile jerk***- ***index 2011h*** or ***rounding time-index 2021h***) are adjusted when the motor is standstill and a new positioning is starting. The speed instead (***profile velocity*** - ***index 6081h***) is immediately adjusted to the set value.
-1	Linear ramp with torque limit	• This mode acts as the mode 0 since it executes the positionings using a speed profile with a trapezoidal shape. However, if the positioning is blocked by an obstacle, the algorithm generating the position profile stops, and it will not resume as long as the motor keeps pushing with the maximum torque (set with the parameter ***Current limit*** – ***index 2006h***). When the torque falls below its limit, the positioning is resumed to reach the chosen setpoint. • In this mode the position error may not be detected since the position profile adapts to the measured position value when the torque limit is reached.

4.07.9 Object 2011h – Profile Jerk

Index	Name	Type	Range	Default	Unit	Map	Attributes
2011h	Profile Jerk	UINT32	1..FFFFFFFFh	100000		No	rw

The "***Profile jerk***" is used by the ***Positioner*** control when the "***Profile type***" (***index 6097h***) is set to "***jerk-limited ramp***" and bit 2 of "***position-flag***" (***index 201Ah***) is set to 0. This parameter affects the sections of the position profile with a cubic curve by changing the acceleration's variation. A higher jerk value implies bigger acceleration variations, which reduce the time needed for the positioning but increase the mechanical stress, vice versa a lower jerk increases the time needed for the positioning but reduce the mechanical stress. The unit of measure of this parameter is "acceleration over time", which corresponds inside of the Drive to "pulses per second to the cube" [pulses/s^3]. The "***acceleration factor***" (***index 6097h***) is used to convert the jerk to the internal unit of measure of the Drive, as shown in the formula below:

$$Profile\_Jerk[counts/s^3] = \frac{(Numerator \times Profile\_Jerk[rpm/s^2])}{Divisor}$$

By using the default values of the "***Acceleration factor***", i.e. ***Numerator***=65536 and ***Divisor***=60, the profile jerk is expressed in [rpm/s^2].

4.07.10 Object 2021h – Rounding time

Index	Name	Type	Range	Default	Unit	Map	Attributes
2021h	Rounding time	UINT16	1..65535	100	ms	No	rw

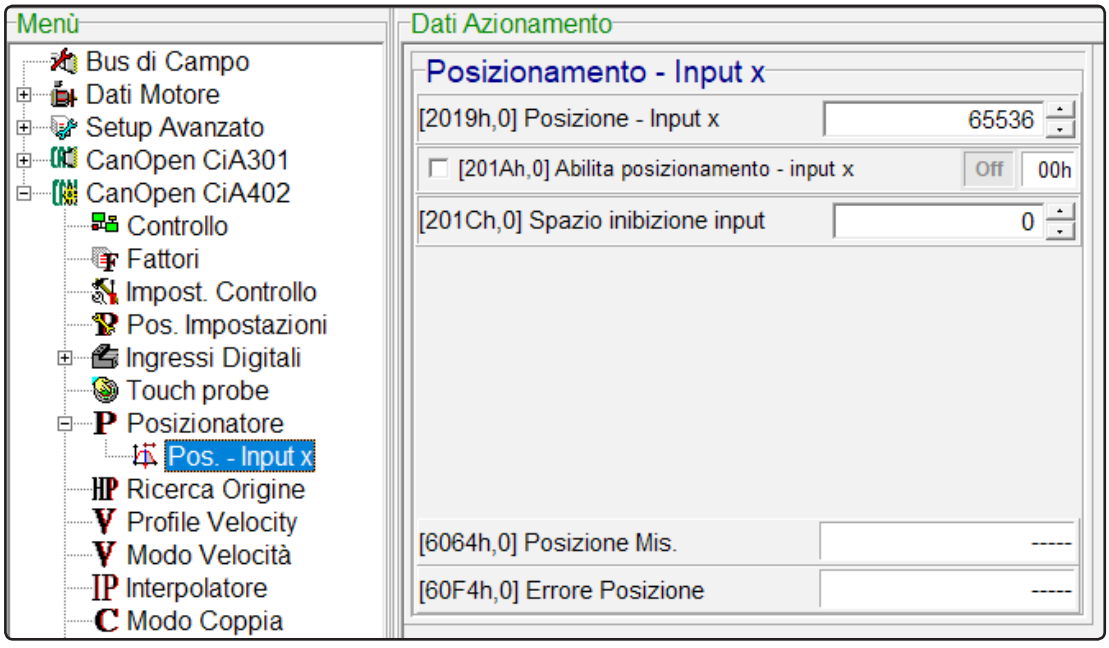
When the "***profile type***" (***index 6086h***) is set to "***jerk- limited ramp***" and bit 2 of "***position-Flag***" (***index 201Ah***) is set to 1, the length of the rounded section of the position profile (with an "S" shape) corresponds to the value set in this parameter.

PRELIMINARY

4.07.11 Relative positioning launched by input X

Our Drives support a function that starts a relative positioning with respect to the current position when a rising edge is detected on a chosen input. Since this could be any input, we will call it "***Input X***" from now on. This function could be used only with the ***Positioner*** control type, and it is enabled by putting to 1 the bit 3 of "***Position-Flag***" (***index 201Ah***).

The parameters required by this function can be seen in the Caliper menu "(***EtherCat-)CanOpen CiA402-Positioner-Pos. Input x***" in which the following screen appears:



4.07.12 Object 2019h – Position - input x

Index	Name	Type	Range	Default	Unit	Map	Attributes
2019h	**Position input x**	UINT32	0..7FFFFFFFh	65536		No	rw

This parameter is used to set the length of the relative positioning to start -with respect to the current measured position- when a rising edge is detected on "***Input X***". This value does not have a sign, because the positioning direction is calculated via software to make it equal with the direction of the current speed.

4.07.13 Object 201Ch – Inhibit space input

Index	Name	Type	Range	Default	Unit	Map	Attributes
201Ch	**Inhibit space input**	UINT32	0..7FFFFFFFh	0		No	rw

This parameter represents the minimum space that the motor must cover before the ***Input X*** can launch a new positioning. If the ***Input X*** detects a rising edge before the motor covers this space, it will consider it a disturbance and ignore it.

4.07.14 Bit 14 of the statusword for the recording of the positioning on input X

To verify that the relative positioning has been completed, after the Input X detects a rising edge the bit 14 of the ***statusword*** is allocated. This bit is active only when bit 3 of "***Position-Flag***" (***index 201Ah***) is 1, and it becomes significant only when the positioning is over, i.e. when bit 10 ("***Target reached***") of the statusword is 1. In this situation the bit 14 could take on one of these meanings:

- 0: positioning completed without detecting a rising edge on input X.
- 1: the current positioning is completed, and a rising edge has been detected by the Input X, so a new relative positioning is started, according to the "***position-input X***" (***index 2019h***) value.

4.07.15 Setting of the "positioning on Input X" function

The function that launches a relative positioning on input X command is activated by putting to 1 bit 3 of the "***position-flag***" (***index 201Ah***). Whem this function is enabled the Drive is ready to automatically start a relative positioning on input X command, i.e. when bit 4 ("***New set-point***") of the ***controlword*** is set to 1. When the first rising edge is detected by input X, a relative positioning is commanded with respect to the current position, such that its speed has the same sign as the current speed of the motor and the space covered is equal to the "***Position-input X***" (***index 2019h***) parameter.

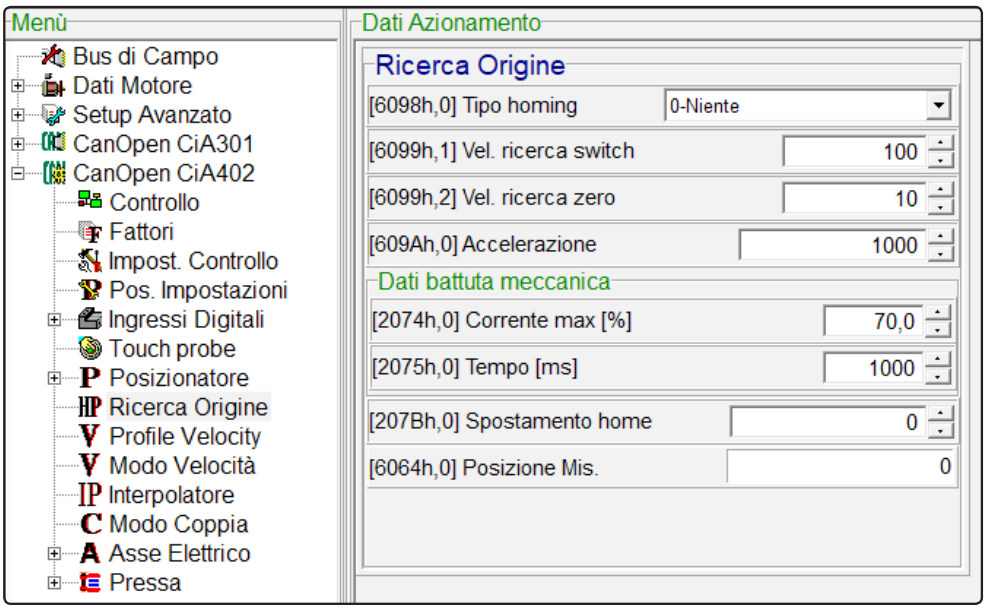
The bit 14 of the ***statusword*** is set to 1 only when the relative positioning is over, otherwise it remains to 0.

4.08 Homing Mode

This chapter describes the methods and the parameters of the procedure used to find the "position 0" of the application. Depending on the case, certain limit switch may be used to affect this search.

Index	Name	Type	Map	Attributes
6098h	Homing method	INT8	No	rw
6099h	Homing speed	(ARRAY) UINT32	Yes	rw
609Ah	Homing acceleration	UINT32	No	rw
2074h	Current max.	UINT16	No	rw
2075h	Mechanical stop time	UINT16	No	rw
207Bh	Home shift	INT32	No	rw

These parameters can be seen in the Caliper menu "***(EtherCat-)CanOpen CiA402- Homing mode***" in which the following screen appears:



4.08.1 Controlword bits setting - Homing Mode

Bit	Description	
0	***Switch on***	
1	***Enable voltage***	
2	***Quick stop***	
3	***Enable operation***	
4	***Homing operation start***	
	Value	**Description**
	0	***Homing mode*** disabled
	0 -> 1	The homing procedure starts
	1	***Homing mode*** enabled
	1 ->0	The homing procedure stops
5..6	***Reserved***	
7	***Fault reset***	

Bit	Description	
8	***Halt***	
	Value	**Description**
	0	The istructions of bit 4 are executed
	1	The motor stops with a ramp, the deceleration applyed is equal to acceleration ramp of the homing procedure.
10..9	***Reserved***	
15..11	***Manufacturer specific***	

4.08.2 Statusword bits - Homing Mode

Bit	Description		
0	***Ready to switch on***		
1	***Switched on***		
2	***Operation enable***		
3	***Fault***		
4	***Voltage enabled***		
5	***Quick stop***		
6	***Switch on disabled***		
7	***Warning***		
8	***Manufacturer specific***		
9	***Remote***		
10	***Target reached***		
	value	**Bit Halt Controlword**	**Description**
	0	0	***Home position*** not reached
		1	The axis is decelerating
	1	0	***Home position*** reached
		1	the axis is stationary
11	***Internal limit active***		
12	***Home attained***		
	value	**Description**	
	0	homing procedure not yet completed	
	1	homing procedure completed successfully	
13	***Homing error***		
	value	**Description**	
	0	No homing error detected	
	1	Homing error detected	
14	***Manufacturer specific***		
15	***Position measuring*** (enabled by "***Input setting***" – ***index 201Bh***)		
	Value	**Description**	
	0	• ***(index 201Bh bit7=0) continuous mode:*** the position between the rising and falling edge of the chosen input has been calculated. • ***(index 201Bh bit7=1) one shot mode:*** the two values required to calculates the position between the edges of the chosen input have not yet been acquired.	
	1	• ***(index 201Bh bit7=0) continuous mode:*** the position between the rising and falling edge of the chosen input has not yet been calculated. • ***(index 201Bh bit7=1) one shot mode:*** the two values required to calculates the position between the edges of the chosen input have been acquired.	

PRELIMINARY

4.08.3 Object 6098h – Homing method

Index	Name	Type	Range	Default	Unit	Map	Attributes
6098h	**Homing method**	INT8	0..35 (eccetto 15,16, 31 e 32)	0		No	rw

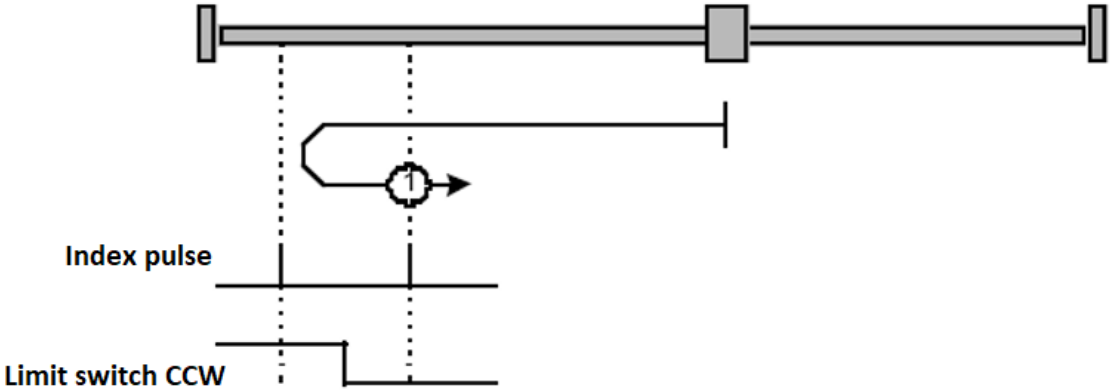
The ***Homing method*** parameter allows you to choose the method to be used to search for the position 0 of the application. Our Drives support all the CanOpen homing methods, from method 0 to 35, with the exception of methods 15,16,31, and 32 which are reserved.

4.08.3.a Method 0 - No homing

When the drive is turned on the value of the measured position is reset to zero and set as the home position of the Drive.

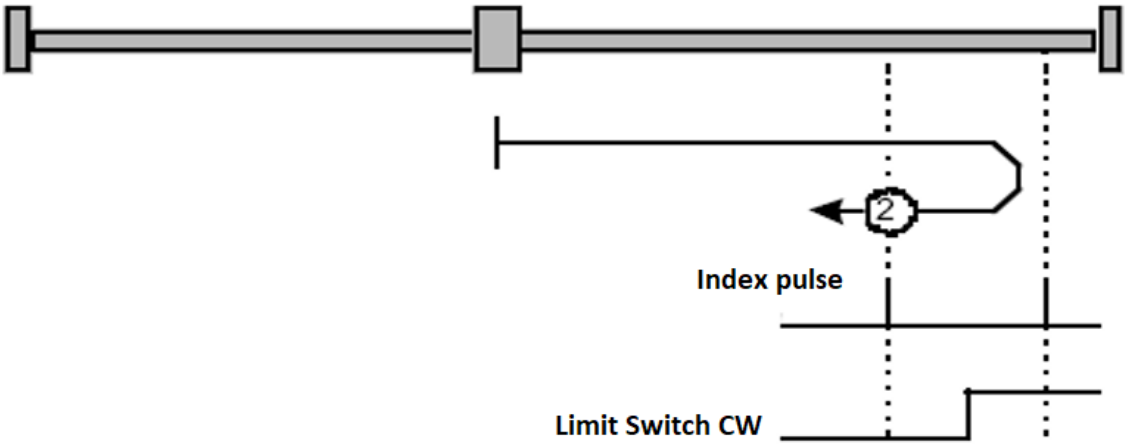
4.08.3.b Method 1 - Homing on negative limit switch and index pulse

The drive performs the ***homing*** by moving the motor counterclockwise until it finds the limit switch sensor CCW. Once it has reached the sensor, it changes direction and moves at low speed to exit from the limit switch sensor and then it continues until an encoder index pulse is found; that point will then become the home position of the drive.



4.08.3.c Method 2 - Homing on positive limit switch and index pulse

The drive performs the ***homing*** by moving the motor clockwise until it finds the limit switch sensor CW. Once it has reached the sensor, it changes direction and moves at low speed to exit from the limit switch sensor and then it continues until an encoder index pulse is found; that point will then become the home position of the drive.

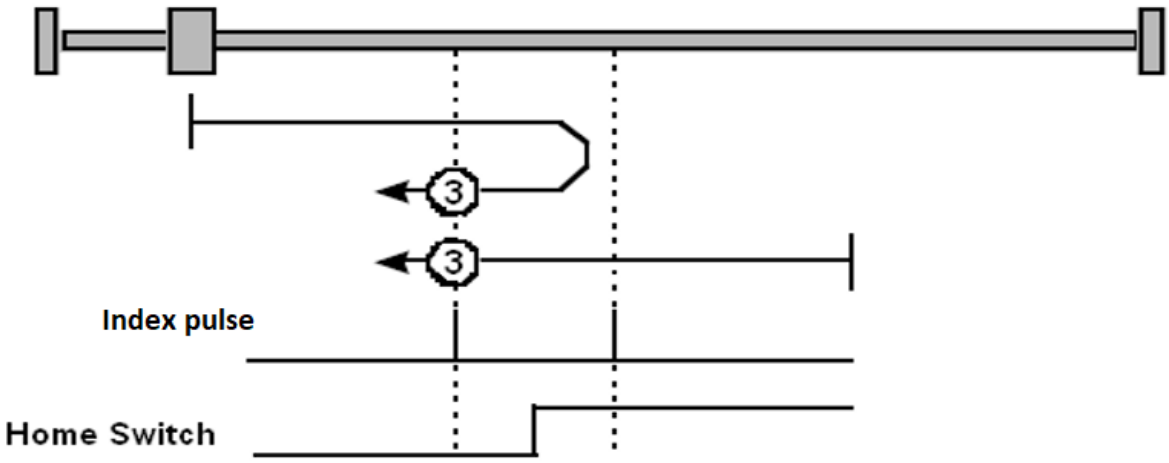


4.08.3.d Method 3 – Homing on positive home switch and index pulse

The logic value of the input assigned to the home sensor determines the direction of the search of the home sensor:

» If the assigned input is low, the motor rotates clockwise and when the home sensor commutes the motor stops and slowly moves counterclockwise until it finds an index pulse of the encoder, which will become the home position.

» If the assigned input is high, the motor rotates counterclockwise and when the home sensor commutes the motor slows down but keeps moving counterclockwise until it finds an index pulse of the encoder, which will become the home position.

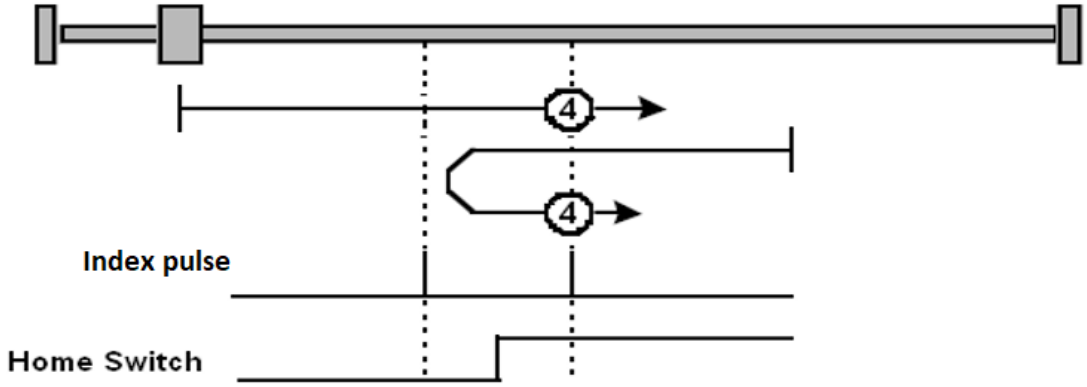


4.08.3.e Method 4 - Homing on positive home switch and index pulse

The logic value of the input assigned to the home sensor determines the direction of the search of the home sensor:

» If the assigned input is high, the motor rotates counterclockwise and when the home sensor commutes the motor stops and slowly moves clockwise until it finds an index pulse of the encoder, which will become the home position.

» If the assigned input is low, the motor rotates clockwise and when the home sensor commutes the motor slows down but keeps moving clockwise until it finds an index pulse of the encoder, which will become the home position.

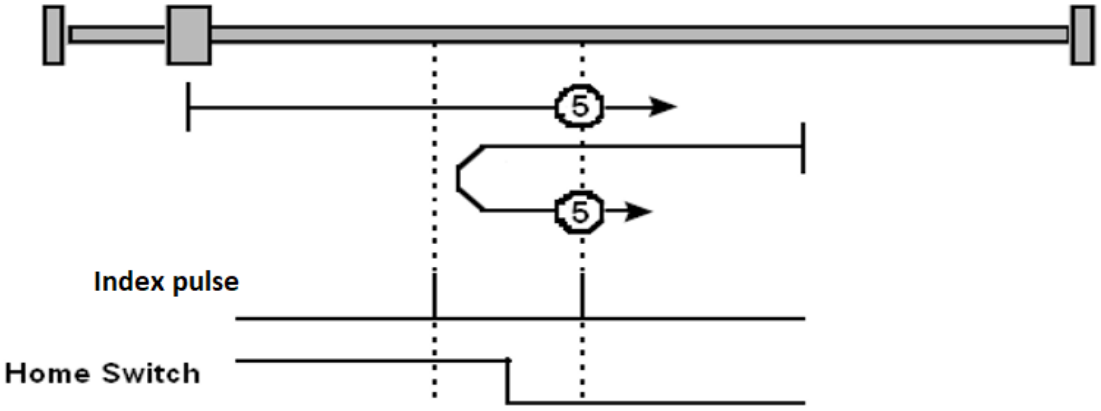


4.08.3.f Method 5 - Homing on negative home switch and index pulse

The logic value of the input assigned to the home sensor determines the direction of the search of the home sensor:

» If the assigned input is low, the motor rotates counterclockwise and when the home sensor commutes the motor stops and slowly moves clockwise until it finds an index pulse of the encoder, which will become the home position.

» If the assigned input is high, the motor rotates clockwise and when the home sensor commutes the motor slows down but keeps moving clockwise until it finds an index pulse of the encoder, which will become the home position.



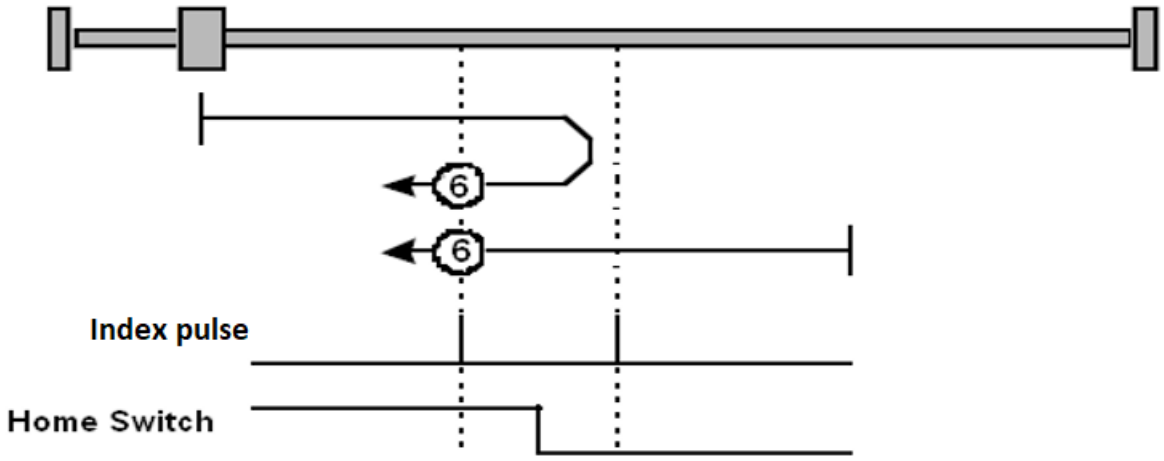
PRELIMINARY

4.08.3.g Method 6 - Homing on negative home switch and index pulse

The logic value of the input assigned to the home sensor determines the direction of the search of the home sensor:

» If the assigned input is high, the motor rotates clockwise and when the home sensor commutes the motor stops and slowly moves counterclockwise until it finds an index pulse of the encoder, which will become the home position.

» If the assigned input is low, the motor rotates counterclockwise and when the home sensor commutes the motor slows down but keeps moving counterclockwise until it finds an index pulse of the encoder, which will become the home position.



4.08.3.h Method 7 - Homing on home switch and index pulse

The drive moves the motor clockwise searching for the home sensor. Once it finds it the motor stops and slowly moves counterclockwise to exit the sight of the sensor, and after that it continues moving counterclockwise until it finds an index pulse of the encoder. The position reached will become the home position. If during this search the motor reaches the limit switch CW, the sense of rotation is switched to counterclockwise to guide the motor towards the home sensor.

4.08.3.i Method 8 - Homing on home switch and index pulse

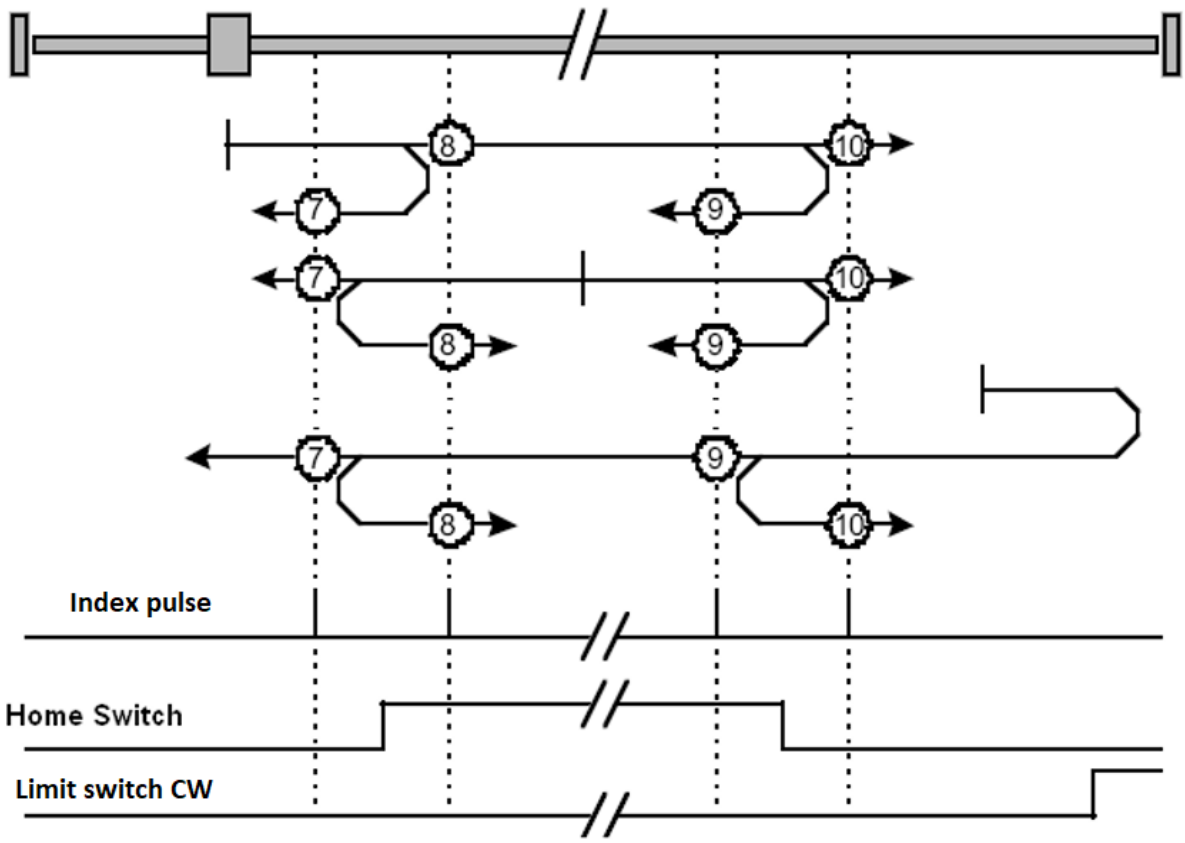
The drive moves the motor clockwise searching for the home sensor. Once it finds it the motor stops and slowly moves counterclockwise to exit the sight of the sensor, and after that it returns moving clockwise until it finds an index pulse of the encoder. The position reached will become the home position. If during this search the motor reaches the limit switch CW, the sense of rotation is switched to counterclockwise to guide the motor towards the home sensor.

4.08.3.j Method 9 - Homing on home switch and index pulse

The drive moves the motor clockwise searching for the home sensor. Once it finds it the motor slows down but keeps moving clockwise to exit the sight of the sensor, and after that it starts moving counterclockwise until it finds an index pulse of the encoder. The position reached will become the home position. If during this search the motor reaches the limit switch CW, the sense of rotation is switched to counterclockwise to guide the motor towards the home sensor.

4.08.3.k Method 10 - Homing on home switch and index pulse

The drive moves the motor clockwise searching for the home sensor. Once it finds it the motor slows down but keeps moving clockwise to exit the sight of the sensor, and after that it continues moving clockwise until it finds an index pulse of the encoder. The position reached will become the home position. If during this search the motor reaches the limit switch CW, the sense of rotation is switched to counterclockwise to guide the motor towards the home sensor.



PRELIMINARY

4.08.3.l Method 11 - Homing on home switch and index pulse

The drive moves the motor counterclockwise searching for the home sensor. Once it finds it the motor stops and slowly moves clockwise to exit the sight of the sensor, and after that it continues moving clockwise until it finds an index pulse of the encoder. The position reached will become the home position. If during this search the motor reaches the limit switch CCW, the sense of rotation is switched to clockwise to guide the motor towards the home sensor.

4.08.3.m Method 12 - Homing on home switch and index pulse

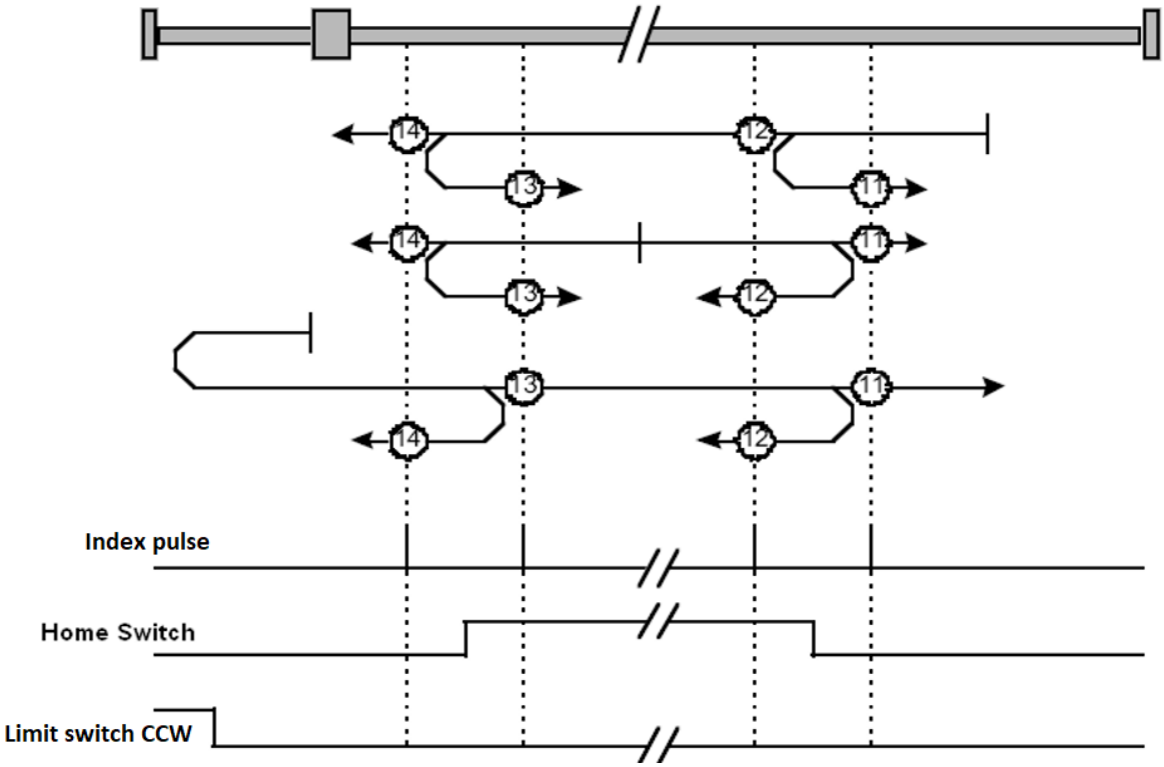
The drive moves the motor counterclockwise searching for the home sensor. Once it finds it the motor stops and slowly moves clockwise to exit the sight of the sensor, and after that it returns moving counterclockwise until it finds an index pulse of the encoder. The position reached will become the home position. If during this search the motor reaches the limit switch CCW, the sense of rotation is switched to clockwise to guide the motor towards the home sensor.

4.08.3.n Method 13 - Homing on home switch and index pulse

The drive moves the motor counterclockwise searching for the home sensor. Once it finds it the motor slows down but keeps moving counterclockwise to exit the sight of the sensor, and after that it starts moving clockwise until it finds an index pulse of the encoder. The position reached will become the home position. If during this search the motor reaches the limit switch CCW, the sense of rotation is switched to clockwise to guide the motor towards the home sensor.

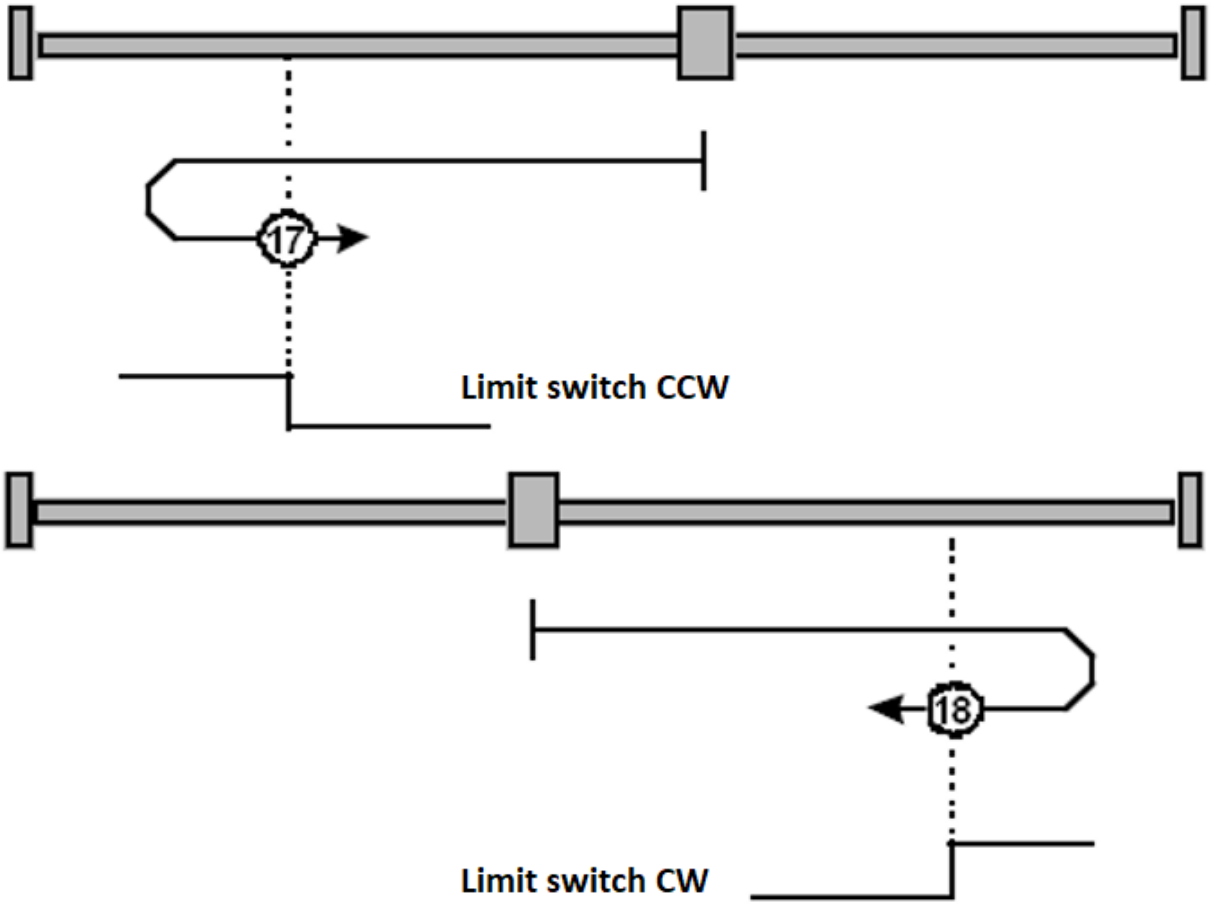
4.08.3.o Method 14 - Homing on home switch and index pulse

The drive moves the motor counterclockwise searching for the home sensor. Once it finds it the motor slows down but keep moving counterclockwise to exit the sight of the sensor, and after that it continues moving counterclockwise until it finds an index pulse of the encoder. The position reached will become the home position. If during this search the motor reaches the limit switch CCW, the sense of rotation is switched to clockwise to guide the motor towards the home sensor.



4.08.3.p Homing without index pulse

The homing methods ranging from 17 to 30 correspond with the respective methods ranging from 1 to 14, the only difference is that in this case the index pulse of the encoder is not searched. For example, in the picture below we can see how the homing methods 17 and 18 are excecuted.

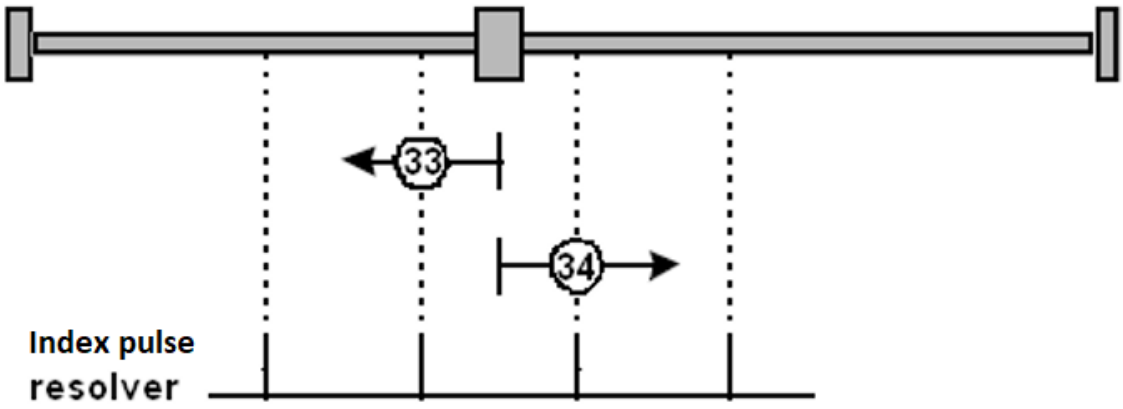


4.08.3.q Method 33 – Homing on index pulse (counterclockwise)

Starting from the position where it is, the drive moves counterclockwise to reach an index pulse of the encoder; the position reached will become the home position.

4.08.3.r Method 34 -Homing on index pulse (clockwise)

Starting from the position where it is, the drive moves clockwise to reach an index pulse of the encoder; the position reached will become the home position.



4.08.3.s Method 35 - Homing on the current position

When this method is escecuted the current position will become the home position.

4.08.4 Object 6099h - Homing speed

Index	sub-index	Name	Type	Range	Default	Unit	Map	Attributes
6099h		Homing speed	ARRAY					
	0	num. of terms	UINT8	2	2		No	ro
	1	switch search speed	UINT32	1..FFFFFFFFh	100		Yes	rw
	2	zero search speed	UINT32	1..FFFFFFFFh	10		Yes	rw

The "***Homing speed***" array includes two parameters, the "***switch search speed***" , that is the speed of the motor during the search of the limit switch, and the "***zero search speed***" , that is the speed of the motor during the search of the zero position of the sensor.

The values of these two parameters are converted to the internal units of the Drive by using the "***vel. encoder factor***" (***index 6094h***), as showed in the following formulas:

$$switch\_search\_speed[\frac{counts}{s}] = \frac{(Numerator \times switch\_search\_speed[rpm])}{Divisor}$$

$$zero\_search\_speed[\frac{counts}{s}] = \frac{(Numerator \times zero\_search\_speed[rpm])}{Divisor}$$

4.08.5 Object 609Ah – Homing acceleration

Index	Name	Type	Range	Default	Unit	Map	Attributes
609Ah	Homing acceleration	UINT32	1..FFFFFFFFh	1000	rpm /s	No	rw

This parameter sets the acceleration and deceleration value of the motor during the homing procedure. This value, expressed in [rpm/s], is converted to the internal units of the Drive by using the "***acceleration factor***" (***index 6097h***), as showed in the following formula:

$$Homing\_acceleration[\frac{counts}{s^2}] = \frac{(Numerator \times Homing\_acceleration[\frac{rpm}{s}])}{Divisor}$$

4.08.6 Object 2074h - Current max.

Index	Name	Type	Range	Default	Unit	Map	Attributes
2074h	Current max.	UINT16	1.. 95	70	%	No	Rw

This parameter sets the limit value of current used to identify when a mechanical stop is found: if the motor is absorbing a current greater than or equal to this limit the Drive assumes that it is pushing against the mechanical stop. This current limit is expressed as a percentage of the nominal current.

4.08.7 Object 2075h - Mechanical stop time

Index	Name	Type	Range	Default	Unit	Map	Attributes
2075h	Mechanical stop time	UINT16	10.. 10000	1000	ms	No	Rw

This parameter sets a time period, used to identify when a mechanical stop is found. if the motor is absorbing a current greater than or equal to the limit seto on ***index 2074h***, for a period longer than the one set here, the Drive assumes that it is pushing against the mechanical stop.

4.08.8 Object 207Bh - Home shift

Index	Name	Type	Range	Default	Unit	Map	Attributes
207Bh	Home shift	INT32	±7FFFFFFFh	0		No	Rw

This parameter, if it differs from 0, allows you to add another offset to the position where the motor would stop after an homing procedure, regardless of the homing method used.

4.08.9 Specific settings for the limit switch

Our Drives include two inputs that,when you're working with CanOpen, are prepared for the limit switches. These inputs are set using the "***input setting***" (***index 201Bh***) parameter.

4.09 Interpolalated Position Mode

This mode is used to control the motor position; the target position of the motor is created by the Drive using a linear interpolation of the position and time data received by the Master.

Index	Name	Type	Map	Attributes
60C0h	Interpol. mode select	INT8	No	rw
60C1h	Interpolator data record	(ARRAY) UINT32	Yes	rw
60C2h	Interpolator time period	RECORD	No	rw

These parameters can be seen in the Caliper menu "***(EtherCat-)CanOpen CiA402- Interpol. mode***" in which the following screen appears:

PRELIMINARY

4.09.1 Controlword bits setting - Interpolated position mode

Bit	Description	
0	***Switch on***	
1	***Enable voltage***	
2	***Quick stop***	
3	***Enable operation***	
4	***Enable ip mode***	
	Value	**Description**
	0	Disables the Interpolator mode
	1	Enables the Interpolator mode
5..6	***Reserved***	
7	***Fault reset***	
8	***Halt***	
	Value	**Description**
	0	the commands of bit 4 are executed
	1	The motor stops with a ramp
10..9	***Reserved***	
15..11	***Manufacturer specific***	

4.09.2 Meaning of the Statusword bits - Interpolated position mode

<table>
<tr><th>Bit</th><th colspan="3">Description</th></tr>
<tr><td>0</td><td colspan="3">Ready to switch on</td></tr>
<tr><td>1</td><td colspan="3">Switched on</td></tr>
<tr><td>2</td><td colspan="3">Operation enable</td></tr>
<tr><td>3</td><td colspan="3">Fault</td></tr>
<tr><td>4</td><td colspan="3">Voltage enabled</td></tr>
<tr><td>5</td><td colspan="3">Quick stop</td></tr>
<tr><td>6</td><td colspan="3">Switch on disabled</td></tr>
<tr><td>7</td><td colspan="3">Warning</td></tr>
<tr><td>8</td><td colspan="3">Manufacturer specific</td></tr>
<tr><td>9</td><td colspan="3">Remote</td></tr>
<tr><td>10</td><td colspan="3">Target reached</td></tr>
<tr><td></td><td>Value</td><td>Controlword Halt Bit</td><td>Description</td></tr>
<tr><td></td><td rowspan="2">0</td><td>0</td><td>Target position not reached yet</td></tr>
<tr><td></td><td>1</td><td>The axis is decelerating</td></tr>
<tr><td></td><td rowspan="2">1</td><td>0</td><td>Target position reached</td></tr>
<tr><td></td><td>1</td><td>the axis is stationary</td></tr>
<tr><td>11</td><td colspan="3">Internal limit active (enabled only when “Enable Position limits” -index 201Ah- is on)</td></tr>
<tr><td></td><td>Value</td><td colspan="2">Description</td></tr>
<tr><td></td><td>0</td><td colspan="2">The absolute value of the target position is not limited by “Pos. limits” (index 607Dh)</td></tr>
<tr><td></td><td>1</td><td colspan="2">The absolute value of the target position is limited by “Pos. limits” (index 607Dh)</td></tr>
<tr><td>12</td><td colspan="3">Ip mode active</td></tr>
<tr><td></td><td>Value</td><td colspan="2">Description</td></tr>
<tr><td></td><td>0</td><td colspan="2">Interpolator disabled</td></tr>
<tr><td></td><td>1</td><td colspan="2">Interpolator enabled</td></tr>
<tr><td>13</td><td colspan="3">Reserved</td></tr>
<tr><td>14</td><td colspan="3">Manufacturer specific</td></tr>
<tr><td>15</td><td colspan="3">Position measuring (enabled by “Input setting” – index 201Bh)</td></tr>
<tr><td></td><td>Value</td><td colspan="2">Description</td></tr>
<tr><td></td><td>0</td><td colspan="2">• (index 201Bh bit7=0) continuous mode: the position between the rising and falling edge of the chosen input has been calculated.
• (index 201Bh bit7=1) one shot mode: the two values required to calculates the position between the edges of the chosen input have not yet been acquired.</td></tr>
<tr><td></td><td>1</td><td colspan="2">• (index 201Bh bit7=0) continuous mode: the position between the rising and falling edge of the chosen input has not yet been calculated.
• (index 201Bh bit7=1) one shot mode: the two values required to calculates the position between the edges of the chosen input have been acquired.</td></tr>
</table>

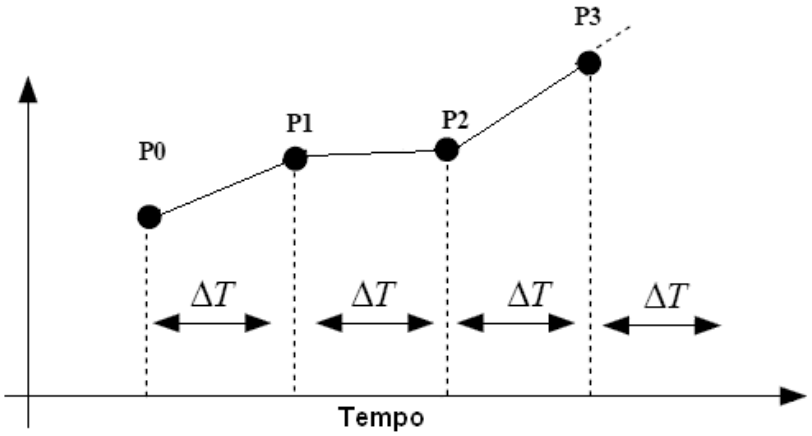
4.09.3 Object 60C0h - Interpol. mode select

Index	Name	Type	Range	Default	Unit	Map	Attributes
60C0h	Interpol. mode select	INT16	0	0		No	Rw

This parameter sets what kind of interpolation to use for the motor positioning.

Note: only the linear interpolation is supported at the moment

Value	Profile type	Description
0	Linear Interpolation	The Drive calculates a linear interpolation from point to point according with the time set on "***Interpolator time period***" (***index 60C2h***). For this kind of interpolation only the position data is required by the Master.



PRELIMINARY

4.09.4 Object 60C1h - Interpolator data record

Index	sub-index	Name	Type	Range	Default	Unit	Map	Attributes
60C1h		**Interpolator data record**	ARRAY					
	0	**num. of terms**	UINT8	2	2		No	ro
	1	**ip. function 1° parameter**	INT32				Yes	rw
	2	**ip. function 2° parameter**	INT32				Yes	rw

This parameter contains the interpolation data sent by the Master. According to the value of "***Interpol. mode select***" (***index 60C0h***), sub-indexes 1 and 2 could assume one of the following meanings:

Interpolator mode	Data record Sub-Index	Meaning	Type
0 = Linear interpolation	**ip. function 1° parameter**	Position data	INT32
	ip. function 2° parameter	Not used	INT32
1 = Cubic interpolation	**ip. function 1° parameter**	Position data	INT32
	ip. function 2° parameter	Speed data	INT32

The position data are converted into the internal units of the Drive using the "***Position factor***" (***index 6093h***), as shown in the following formula:

$$Drive\_ref\_position = \frac{(Numerator \times IPdata\_position)}{(Feedconstant)}$$

The position data are converted into the internal units of the Drive using the "***vel. encoder factor***" (***index 6094h***), as shown in the following formula:

$$Drive\_ref\_speed = \frac{(Numerator \times IPdata\_speed)}{Divisor}$$

4.09.5 Object 60C2h - Interpolator time period

Index	sub-index	Name	Type	Range	Default	Unit	Map	Attributes
60C2h		**Interpolator time period**	RECORD					
	0	**num. of terms**	UINT8	2	2		No	ro
	1	**Interpolation time units**	UINT8	1.. 265	2	ms	No	rw
	2	**Interpolation time index**	INT8	-3	-3		No	rw

PRELIMINARY

The "***Interpolator time period***" defines the period of time between two position values sent by the Master. Sub-index 1 sets its value, Sub-index 2 sets the order of magnitude of its unit of measure. For our Drives sub-index 2 is a readonly variable fixed to -3, which means tha the unit of measure of the interpolator period is [ms].

$$measure\_unit = 10^{time\_index}[s] \rightarrow 10^{-3}[s] = 1\,[ms]$$

$$Interpolation\_time\_period = time\_units[ms]$$

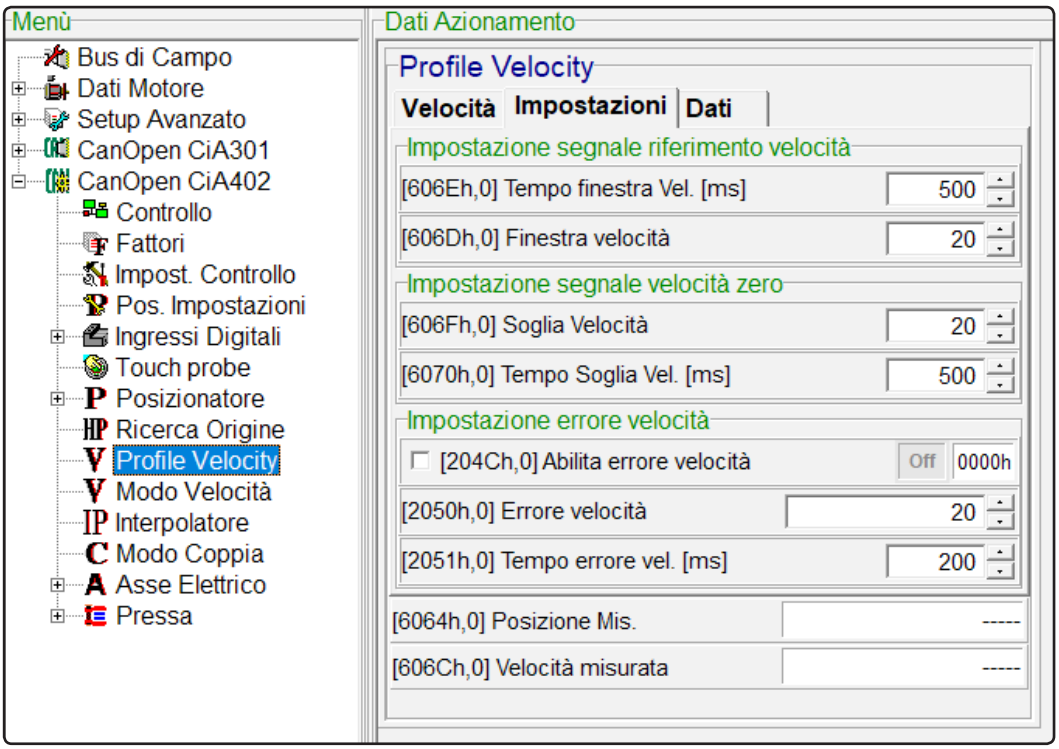
4.10 Profile velocity Mode

With the "***profile velocity***" mode the speed loop is activated, the speed reference is affected by the acceleration and deceleration ramps respectively set by the "***profile acceleration***" (***index 6083h***) and the "***profile deceleration***" (***index 6084h***) parameters of the ***Positioner***.

The parameters related to the "***profile velocity***" mode are reported in the following table:

Index	Name	Type	Map	Attributes
6069h	Velocity sensor actual value	INT32	Yes	ro
606Bh	Velocity demand value	INT32	Yes	ro
606Ch	Velocity actual value	INT32	Yes	ro
606Dh	Velocity window	UINT16	No	rw
606Eh	Velocity window time	UINT16	No	rw
606Fh	Velocity threshold	UINT16	No	rw
6070h	Velocity threshold time	UINT16	No	rw
60FFh	Target velocity	INT32	Yes	rw
204Ch	profile velocity- flag	UINT16	No	rw
2050h	Velocity error	UINT32	No	rw
2051h	Velocity error time	UINT16	No	rw

These parameters can be seen in the Caliper menu "***(EtherCat-)CanOpen CiA402- profile velocity***" in which the following screen appears:



4.10.1 Controlword bits setting - Profile Velocity Mode

Bit	Description	
0	***Switch on***	
1	***Enable voltage***	
2	***Quick stop***	
3	***Enable operation***	
4..6	***Reserved***	
7	***Fault reset***	
8	***Halt***	
	Value	**Description**
	0	the motor moves according with the target velocity
	1	The motor stops with a ramp
10..9	***Reserved***	
15..11	***Manufacturer specific***	

4.10.2 Meaning of theStatusword bits - Profile Velocity Mode

Bit	Description		
0	***Ready to switch on***		
1	***Switched on***		
2	***Operation enable***		
3	***Fault***		
4	***Voltage enabled***		
5	***Quick stop***		
6	***Switch on disabled***		
7	***Warning***		
8	***Manufacturer specific***		
9	***Remote***		
10	***Target reached***		
	Value	**Controlword Halt Bit**	**Description**
	0	0	Target speed not reached yet
		1	The axis is decelerating
	1	0	Target speed reached
		1	the axis is stationary
11	***Internal limit active***		
12	***Zero Speed***		
	Value	**Description**	
	0	The motor speed differs from 0	
	1	The motor speed is equal to 0	
13	***Software limit switch***		
14	***Manufacturer specific***		

PRELIMINARY

Bit	Description	
15	***Position measuring*** (enabled by "***Input setting***" – ***index 201Bh***)	
	Value	**Description**
	0	• ***(index 201Bh bit7=0) continuous mode:*** the position between the rising and falling edge of the chosen input has been calculated. • ***(index 201Bh bit7=1) one shot mode:*** the two values required to calculates the position between the edges of the chosen input have not yet been acquired.
	1	• ***(index 201Bh bit7=0) continuous mode:*** the position between the rising and falling edge of the chosen input has not yet been calculated. • ***(index 201Bh bit7=1) one shot mode:*** the two values required to calculates the position between the edges of the chosen input have been acquired.

4.10.3 Object 6069h - Velocity sensor actual value

Index	Name	Type	Range	Default	Unit	Map	Attributes
6069h	Velocity sensor actual value	INT32	----	----		Yes	ro

The "***Velocity sensor actual value***" reports the value of the speed calculated using the position sensor. It's expressed in pulses per second [pulses/s].

4.10.4 Object 606Bh - Velocity demand value

Index	Name	Type	Range	Default	Unit	Map	Attributes
606Bh	Velocity demand value	INT32	----	----	rpm	Yes	ro

The "***Velocity demand value***" reports the speed value produced at the output of the function that generates the speed ramps. It's expressed in [rpm].

4.10.5 Object 606Ch - Velocity actual value

Index	Name	Type	Range	Default	Unit	Map	Attributes
606Ch	Velocity actual value	INT32	----	----	----	Yes	ro

This parameter reports the measured speed value, used as feedback by the internal speed loop. The measured speed is expressed inside the Drive in pulses per second, and it's converted to the same unit of measure as the ***Target Velocity (index 60FFh)*** by multiplying it by the "***Divisor***" and dividing it by the "***Numerator***" terms of the "***vel. encoder factor***" (***index 6094h***), as shown in the following formula:

$$Profile\_Velocity[rpm] = \frac{(Profile\_Velocity[\frac{counts}{s}] \times Divisor)}{Numerator}$$

By using the default values of the "***vel. encoder factor***", that is ***Numerator***=65536 and ***Divisor***=60, the measured speed is expressed in [rpm].

4.10.6 Object 606Dh - Velocity window

Index	Name	Type	Range	Default	Unit	Map	Attributes
606Dh	Velocity window	UINT16	0..FFFF	20		No	rw

The "***velocity window***" is used to set an interval around the target velocity, as shown in the following formula:

range = [target velocity – velocity window, target velocity + velocity window]

If the measured speed stays inside this range for a period longer than the one set on ***index 606Eh***, then the target velocity is considered to be reached and bit 10 of the ***statusword*** is set to 1. The unit of measure of this range is the same as the "***Target Velocity*** " (***index 60FFh***), and it's converted to the internal units of the Drive (i.e. pulses per second) by using the "***vel. encoder factor***" (***index 6094h***).

4.10.7 Object 606Eh - Velocity window time

Index	Name	Type	Range	Default	Unit	Map	Attributes
606Eh	Velocity window time	UINT16	0..20000	500	ms	No	rw

This parameter is used to set a time period . If the difference between the measured speed and the reference speed remains below a certain value, set on index ***606Dh***, for a longer period than the one set here, then the target velocity is considered to be reached and bit 10 of the ***statusword*** is set to 1.

4.10.8 Object 606Fh - Velocity threshold

Index	Name	Type	Range	Default	Unit	Map	Attributes
606Fh	Velocity threshold	UINT16	0..FFFFh	20		No	rw

This parameter sets a speed threshold. If the absolute value of the measured speed exceeds this one for a period longer than the one set on "***Velocity threshold time***" (***index 6070h***), then bit 12 of the ***statusword*** is set to 0, vive versa in the opposite case bit 12 of the statusword is set to1, to show that the motor is stopped.

The unit of measure of this parameter is the same as the "***Target Velocity*** " (***index 60FFh***), and it's converted to the internal units of the Drive (i.e. pulses per second) by using the "***vel. encoder factor***" (***index 6094h***).

4.10.9 Object 6070h - Velocity threshold time

Index	Name	Type	Range	Default	Unit	Map	Attributes
6070h	Velocity threshold time	UINT16	0..20000	500	ms	No	rw

This parameter sets the time period during which the Drive checks if the measured speed exceeds the value set on "***Velocity threshold*** " (***index 6070h***).

4.10.10 Object 60FFh - Target Velocity

Index	Name	Type	Range	Default	Unit	Map	Attributes
60FFh	Target Velocity	INT32	±7FFFFFFFh	0		Yes	rw

This parameter sets the target velocity given as input to the function generating the acceleration and deceleration ramps.

This parameter is multiplied by the "***Numerator***" and divided by the "***Divisor***" of the "***vel. encoder factor*** " (***index 6094h***); by using the default values ***Numerator***=65536 and ***Divisor***=60, the target velocity obtained is expressed in [rpm].

4.10.11 Object 204Ch – Profile velocity-Flag

Index	Name	Type	Range	Default	Unit	Map	Attributes
204Ch	Profile velocity-Flag	UINT16	0..FFFFh	0		No	rw

This bitwise variable is used to enable/disable two functions used with the "***Profile velocity***" mode.

Bit	Description	
0	***Enable/disable monitoring of the velocity error***: when this bit is set to1and the Drive is set on "***Profile velocity mode***", then the velocity error is monitored. If this error exceeds the threshold set on "***Velocity error***" (***index 2050h***) for a time longer than the "***Velocity error time***" (***index 2051h***) value, an emergency message is sent with the error code 8400h, the Drive goes to the "***fault***" state, the power is disabled and the motor coasts to a stop.	
	Value	**Description**
	0	Monitoring of the velocity error disabled
	1	Monitoring of the velocity error enabled
1	***Position Limits***	
	Value	**Description**
	0	The absolute value of the measured position is not limited by "***Pos. limits***" (***index 607Dh***)
	1	The absolute value of the measured position is limited by "***Pos. limits***" (***index 607Dh***)
2..15	***Reserved***	

4.10.12 Object 2050h – Velocity error

Index	Name	Type	Range	Default	Unit	Map	Attributes
2050h	Velocity error	UINT32	0.. 2147483647	20		No	rw

The "***Velocity error***" parameter sets a speed threshold. If the difference between the measured speed and the reference speed exceeds this threshold for a time greater than the one seets on "***Velocity error time***" (***index 2051h***) and bit 0 of the object ***204Ch*** is set to 1, an emergency message is sent with the error code 8400h and the Drive goes to the "***fault***" state.

The unit of measure of this parameter is the same as the "***Target Velocity*** " (***index 60FFh***), and it's converted to the internal units of the Drive (i.e. pulses per second) by using the "***vel. encoder factor***" (***index 6094h***).

4.10.13 Object 2051h – Velocity error time

Index	Name	Type	Range	Default	Unit	Map	Attributes
2051h	Velocity error time	UINT16	0.. 60000	200	ms	No	rw

The "Velocity error time" value is used together with the "***Velocty error***" (***index 2050h***) and the "***Profile velocity-flag***" (***index 204Ch***) to managee the "***velocity error***" alarm (see *"4.10.12 Object 2050h – Velocity error" pag. 96*).

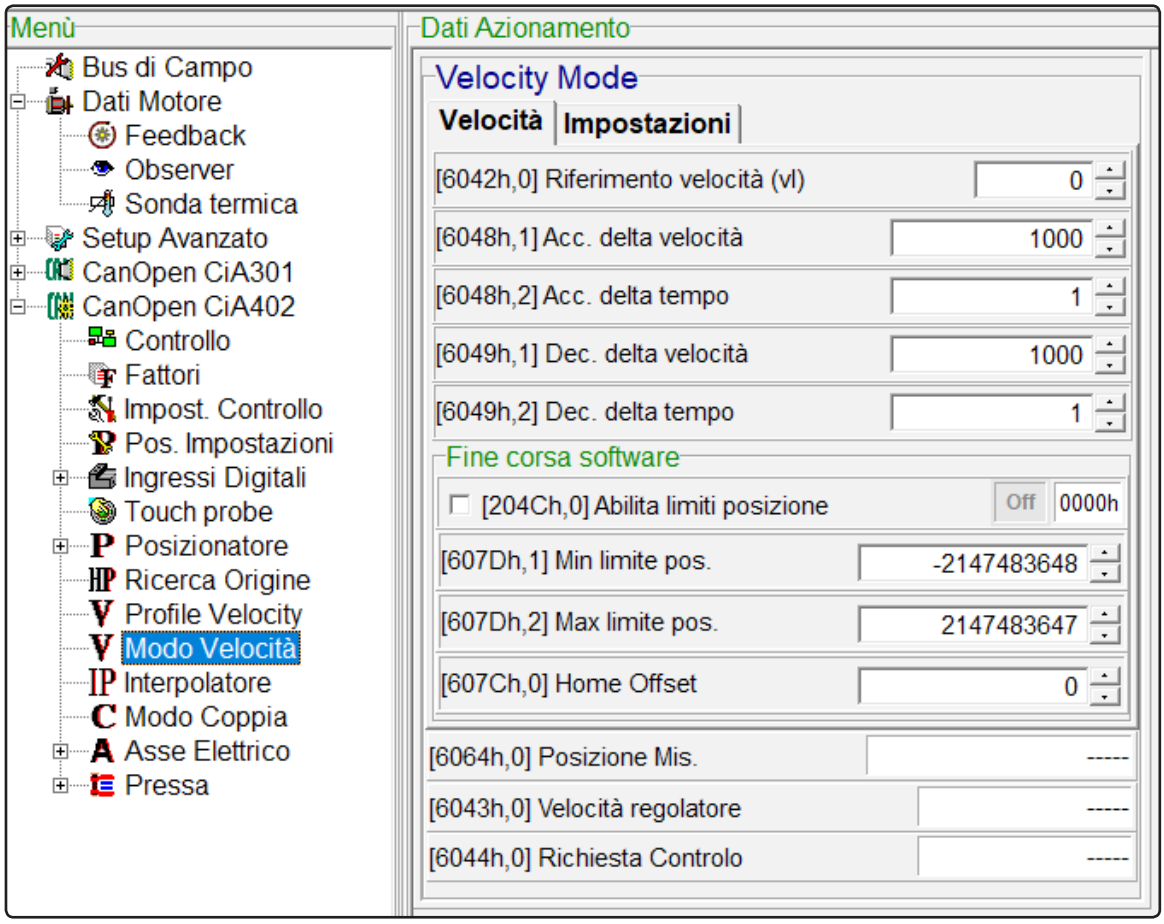
4.11 Velocity mode

With the "***velocity***" mode the speed loop is activated, and a set of parameters are avaiable to modify and limit the reference speed, for example by using speed ramps.

Index	Name	Type	Map	Attributes
6042h	Target velocity (vl)	INT16	Yes	rw
6043h	velocity demand value	INT16	Yes	ro
6044h	Control effort	INT16	Yes	ro
6046h	min/max velocity amount	(ARRAY) UINT32	No	rw
6048h	Acceleration (vl)	RECORD	Yes	rw
6049h	Deceleration (vl)	RECORD	Yes	rw

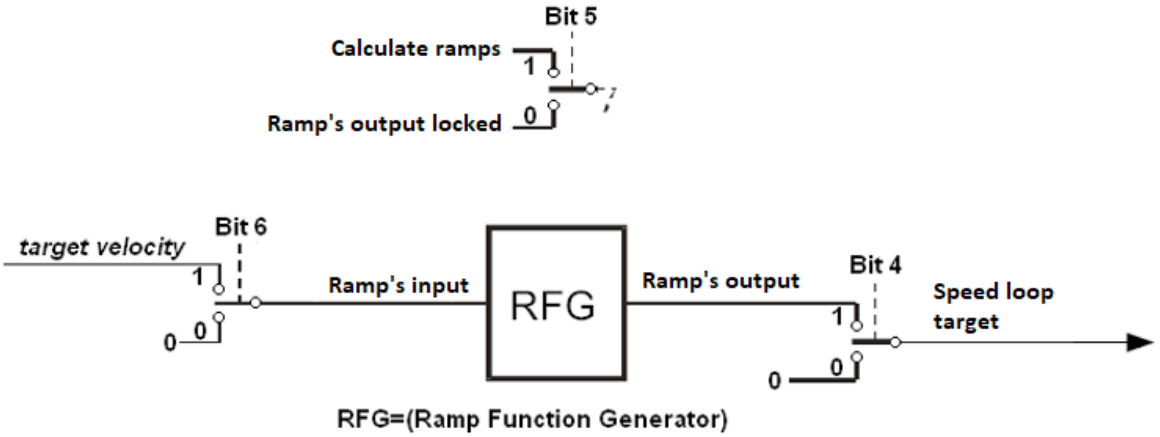
PRELIMINARY

These parameters can be seen in the Caliper menu "***(EtherCat-)CanOpen CiA402- Velocity mode***" in which the following screen appears:



4.11.1 Controlword bits setting- Velocity mode

Bit	Description	
0	***Switch on***	
1	***Enable voltage***	
2	***Quick stop***	
3	***Enable operation***	
4	***Enable ramp function***	
	Value	**Description**
	0	The motor is stopped with the power on
	1	The reference for the speed loop is enabled and corresponds to the output of the acceleration and deceleration ramps
5	***Ramp function unlock***	
	Value	**Description**
	0	The output of the speed ramp is blocked to the actual value
	1	The output variable of the speed ramp function is enabled, and is updated according to the "***acceleration (vl)***" (***index 6048h***) and "***deceleration (vl)***" (***index 6049h***) parameters.
6	***Enable reference ramp***	
	Value	**Description**
	0	The input of the speed ramp function is set to 0
	1	The "***Target velocity (vl)***" (***index 6042h***) value is given as input to the speed ramp function
7	***Fault reset***	
8	***Halt***	
	Value	**Description**
	0	The motor moves according with the speed reference
	1	The motor stops with a ramp
15..9	***Manufacturer specific***	



RFG=(Ramp Function Generator)

PRELIMINARY

4.11.2 Meaning of the Statusword bits - Velocity mode

Bit	Descrizione
0	***Ready to switch on***
1	***Switched on***
2	***Operation enable***
3	***Fault***
4	***Voltage enabled***
5	***Quick stop***
6	***Switch on disabled***
7	***Warning***
8	***Manufacturer specific***
9	***Remote***
10	***reserved***
11	***Internal limit active***
12..13	***reserved***
14	***Manufacturer specific***
15	***Position measuring*** (enabled by "***Input setting***" – ***index 201Bh***)

Value	Description
0	• ***(index 201Bh bit7=0) continuous mode:*** the position between the rising and falling edge of the chosen input has been calculated. • ***(index 201Bh bit7=1) one shot mode:*** the two values required to calculates the position between the edges of the chosen input have not yet been acquired.
1	• ***(index 201Bh bit7=0) continuous mode:*** the position between the rising and falling edge of the chosen input has not yet been calculated. • ***(index 201Bh bit7=1) one shot mode:*** the two values required to calculates the position between the edges of the chosen input have been acquired.

4.11.3 Object 6042h – Target velocity (vl)

Index	Name	Type	Range	Default	Unit	Map	Attributes
6042h	Target velocity (vl)	INT16	±7FFF	100		Yes	rw

This parameter is multiplied by the "***Numerator***" and then divided by the "***Denominator***" of the "***Dimensional factor***" (***index 604Ch***), the obtained result is the reference speed, expressed in [rpm], that is given as input to the function that generates the speed ramps.

4.11.4 Object 6043h - Velocity demand value

Index	Name	Type	Range	Default	Unit	Map	Attributes
6043h	velocity demand value	INT16	±7FFF			Yes	ro

This parameter represents the speed calculated by the function that generates the speed ramps. This value is the reference given to the speed loop, and it's expressed in the same unit of measure of "***Target velocity (vl)*** " ***(index 6042h)***.

4.11.5 Object 6044h - Control effort

Index	Name	Type	Range	Default	Unit	Map	Attributes
6044h	Control effort	INT16	±7FFF			Yes	ro

This parameter corresponds to the measured speed of the motor, expressed in the same unit of measure of "***Target velocity (vl)*** " ***(index 6042h)***.

4.11.6 Object 6046h - min/max velocity amount

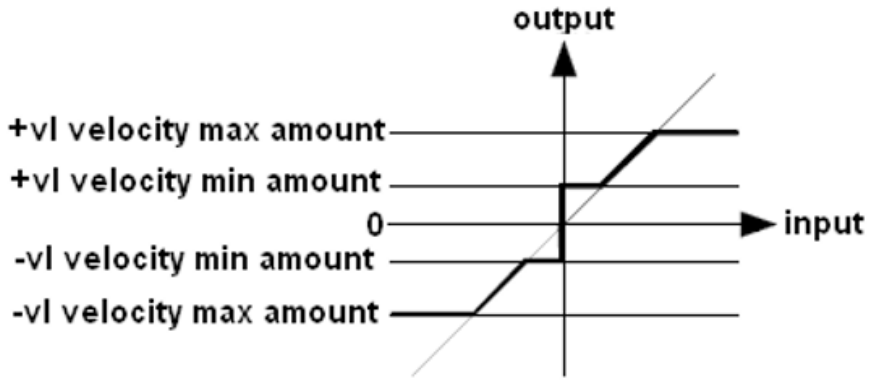
Index	sub-index	Name	Type	Range	Default	Unit	Map	Attributes
6046h		min/max velocity amount	ARRAY					
	0	num. of terms	UINT8	2	2		No	ro
	1	min velocity amount	UINT32	0..7FFFFFFF	0		No	rw
	2	max velocity amount	UINT32	0..7FFFFFFF	5000		No	rw

The "***min/max velocity amount***" array contains two terms that are used to limit the absolute value of the "***target velocity (vl)***" ***(index 6042h)*** within a certain range.

The "***min velocity amount***" parameter sets the minimum absolute value that the reference speed can reach, which means that if the reference speed is positive, its minimum achievable value is "+ ***min velocity amount***", otherwise if the reference speed is negative, its maximum achievable value is "- ***min velocity amount***".

The "***max velocity amount***" parameter sets the maximum absolute value that the reference speed can reach, which means that if the reference speed is positive, its maximum achievable value is "+ ***max velocity amount***", otherwise if the reference speed is negative, its minimum achievable value is "- ***max velocity amount***".

"***max velocity amount***" must be greater than "***min velocity amount***", if you try to do the opposite the error code (in hexadecimal) 06090030 is returned.



4.11.7 Object 6048h - Acceleration (vl)

Index	sub-index	Name	Type	Range	Default	Unit	Map	Attributes
6048h		Acceleration (vl)	RECORD					
	0	num. of terms	UINT8	2	2		No	ro
	1	delta speed	UINT32	1..FFFFFFFFh	1000		Yes	rw
	2	delta time	UINT16	0..FFFFh	1	s	Yes	rw

The "***Acceleration (vl)***" record contains two parameters used the set the acceleration.

The "***Delta speed***" parameter sets a speed variation, expressed in the same unit of measure of the "***target velocity (vl)***" ***(index 6042h)***. The "***Delta time***" sets a time interval, expressed in seconds. The acceleration is calculated using these parameters as shown in the following formula:

$$ref\_acceleration = \frac{(Delta\_speed)}{(Delta\_time)}$$

If "***Delta time***" is set to 0 the reference speed is followed instantly, any increment of the reference speed is done istantly and is not.

4.11.8 Object 6049h - Deceleration (vl)

Index	sub-index	Name	Type	Range	Default	Unit	Map	Attributes
6048h		**Deceleration (vl)**	RECORD					
	0	**num. of terms**	UINT8	2	2		No	ro
	1	**delta speed**	UINT32	1..FFFFFFFFh	1000		Yes	rw
	2	**delta time**	UINT16	1..FFFFh	1	s	Yes	rw

The "***deceleration (vl)***" record contains two parameters used to set the deceleration of the speed ramps in velocity mode. The "***delta speed***" parameter sets a speed variation, expressed in the same unit of measure of the "***target velocity (vl)***" (***index 6042h***). The "***delta time***" parameter sets a period, expressed in seconds. The deceleration is calculated by these parameters as shown in the following formula:

$$ref\_deceleration = \frac{(Delta\_speed)}{(Delta\_time)}$$

If "***delta time***" is equal to 0, the reference speed is used directly, and speed decrements are followed instantly with no decelerating ramps.

PRELIMINARY

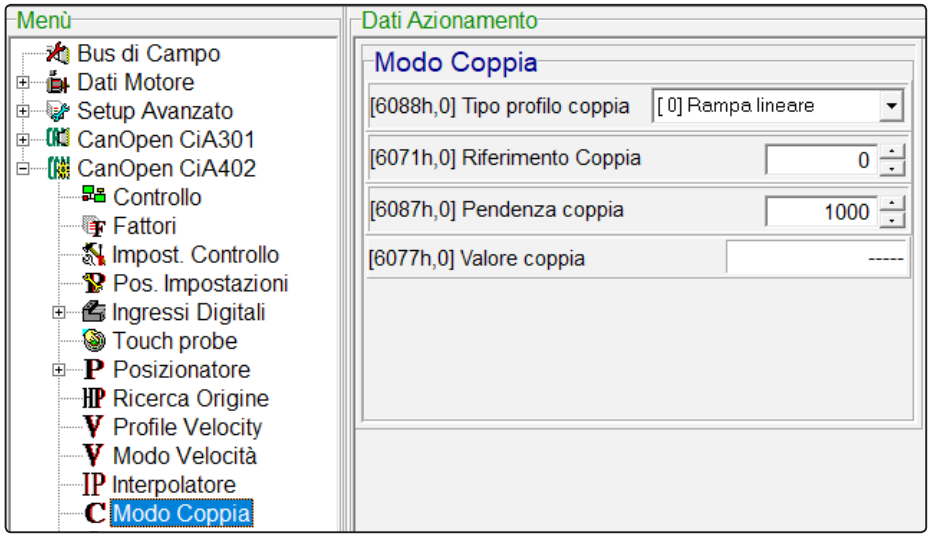
4.12 Profile Torque Mode

The "***Profile Torque***" mode is selected by setting to 4 the "***Modes of operation***" (***index 6060h***) parameter. With this mode only the quadrature current loop is enabled which controls the motor torque. The reference value for this loop is expressed in parts per thousand of the "***Nominal current***" (***index 2001h***), for example by putting the "***Target torque***" (***index 6071h***) to 1000, the motor will absorb a torque current equal to the nominal current.

Depending on the "***Torque profile type***" (***index 6088h***) parameter, the torque reference can be used by the current loop directly or after being smoothed with linear ramps. In the second case, the bit 10 of the ***statusword*** is set to 1 when the torque ramp is terminated and the torque reference corresponds to the set value.

Index	Name	Type	Map	Attributes
6071h	Target torque	INT16	Yes	rw
6077h	Torque actual value	INT16	Yes	ro
6087h	Torque slope	UINT32	No	rw
6088h	Torque profile type	INT8	No	rw

The parameters required by this mode can be seen in the Caliper menu "***(EtherCat-)CanOpen CiA402- Torque mode***" in which the following screen appears:



4.12.1 Controlword bits setting- Torque mode

Bit	Description	
0	***Switch on***	
1	***Enable voltage***	
2	***Quick stop***	
3	***Enable operation***	
4..6	***Reserved***	
7	***Fault reset***	
8	***Halt***	
	Value	**Description**
	0	The motor moves according with the torque reference
	1	The motor stops with a ramp
10.. 9	***Reserved***	
15.. 11	***Manufacturer specific***	

4.12.2 Meaning of the Statusword bits - Torque mode

Bit	Description	
0	***Ready to switch on***	
1	***Switched on***	
2	***Operation enable***	
3	***Fault***	
4	***Voltage enabled***	
5	***Quick stop***	
6	***Switch on disabled***	
7	***Warning***	
8	***Manufacturer specific***	
9	***Remote***	
10	***Target reached***	
	Value	**Description**
	0	the reference has not yet been reached
	1	the reference has been reached
11	***Internal limit active***	
12..14	***Manufacturer specific***	
15	***Position measuring*** (enabled by "***Input setting***" – ***index 201Bh***)	
	Value	**Description**
	0	• ***(index 201Bh bit7=0) continuous mode:*** the position between the rising and falling edge of the chosen input has been calculated. • ***(index 201Bh bit7=1) one shot mode:*** the two values required to calculates the position between the edges of the chosen input have not yet been acquired.
	1	• ***(index 201Bh bit7=0) continuous mode:*** the position between the rising and falling edge of the chosen input has not yet been calculated. • ***(index 201Bh bit7=1) one shot mode:*** the two values required to calculates the position between the edges of the chosen input have been acquired.

PRELIMINARY

4.12.3 Object 6071h – Target torque

Index	Name	Type	Range	Default	Unit	Map	Attributes
6071h	Target torque	INT16	±2500	0		Yes	rw

The "***Target torque***" parameter sets the torque reference value for the current loop. This value is expressed in parts per thousand of the "***Nominal current***" (***index 2001h***). The current reference is calculated using this value as shown in the following formula:

$$Torque\_current = \frac{(Target\_Torque \times Nominal\_current)}{(1000)}$$

4.12.4 Object 6077h – Torque actual value

Index	Name	Type	Range	Default	Unit	Map	Attributes
6077h	Torque actual value	INT16	±2500	0		Yes	ro

This parameter reports the measured value of the "torque" expressed in parts per thousand of the "***Nominal current***" (***index 2001h***).

4.12.5 Object 6087h – Torque slope

Index	Name	Type	Range	Default	Unit	Map	Attributes
6087h	Torque slope	UINT32	1..25000	1000		No	rw

This parameter sets the current variation per second applied when the torque reference is changing and the linear ramps are enabled, i.e when "***Torque profile type***" (***index 6088h***) is set to 0. It is expressed in parts per thousand of the "***Nominal current***" (***index 2001h***).

4.12.6 Object 6088h - Torque profile type

Index	Name	Type	Range		Default	Unit	Map	Attributes
6088h	Torque profile type	INT8	-1	Immediate	0		No	rw
			0	Linear ramp				

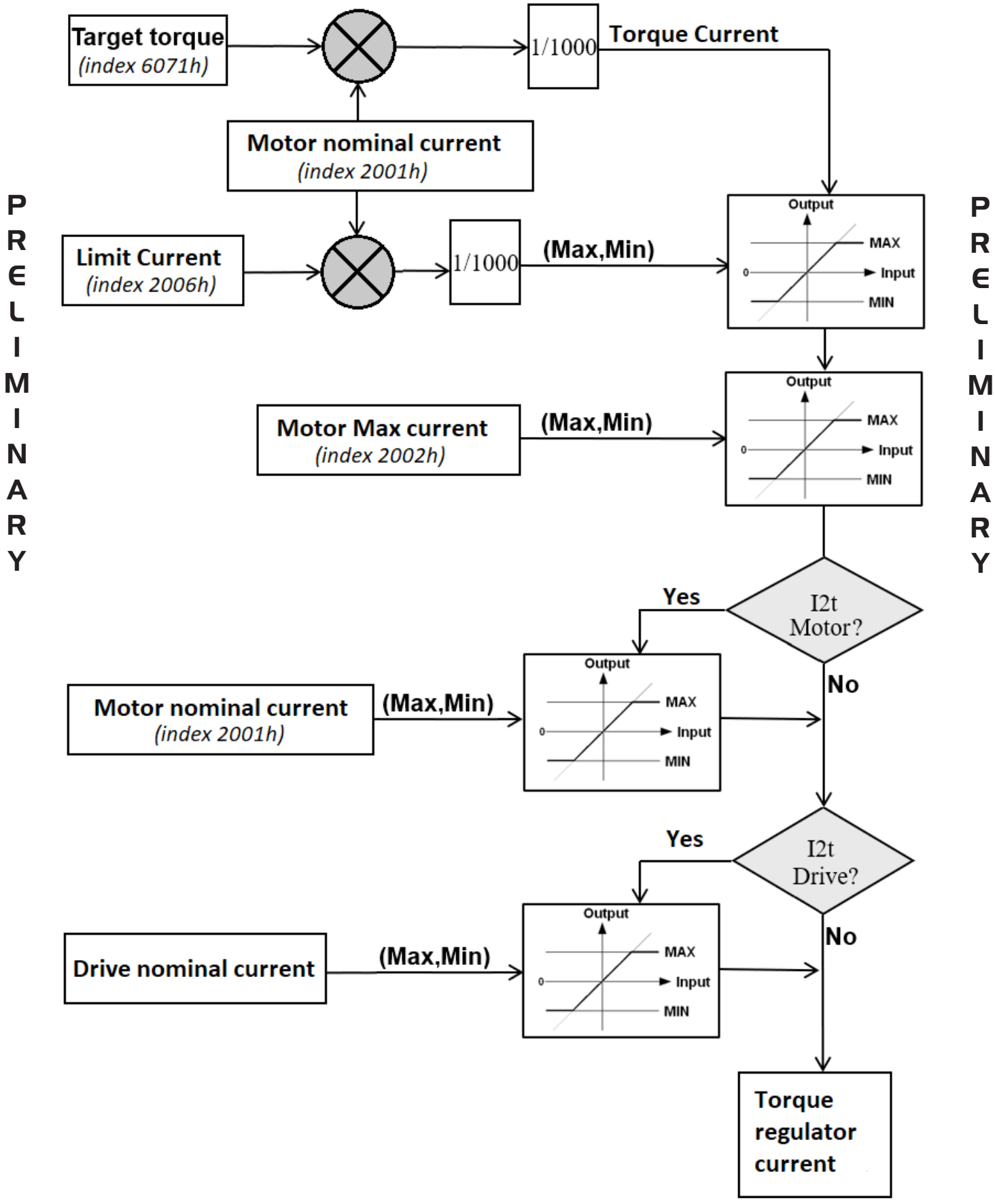
This parameter is used to set how the reference current will be processed before entering the current loop.

Value	Torque profile type	Description
-1	Immediate	The "***Target torque***" (***index 6071h***) is converted directly into the torque current, its value is expressed in parts per thousand of the "***Nominal current***" (***index 2001h***). This value then is applied by the regulator as it is.
0	Linear ramp	The reference current obtained by the "***Target torque***" (***index 6071h***) is smoothed with linear ramps that can be set using the ***"Torque slope" (index 6078h)*** parameter.

4.12.7 Torque current limits

The torque current obtained by the "***Target torque***" (***index 6071h***) and the "***Nominal current***" (***index 2001h***) is subsequently limited by other parameters of the Drive. The first limit is obtained by the "***Current limit***" (***index 2006h***) parameter, which is expressed in parts per thousand of the "***Nominal current***" (***index 2001h***). Its value corresponds to the maximum limit; the minimum limit is equal to the maximum but with the opposite sign. The torque current then is limited again by the "***Peak current***" (***index 2002h***) parameter and by the maximum current that a Drive could produce, which is an internal parameter that cannot be modified.

If the motor I2t alarm occurs, the torque current is limited to the "***Nominal current***" (***index 2001h***) value; if instead the Drive I2t alarm occurs, the torque current is limited to the maximum current that the Drive can produce.



4.13 Gearbox mode

With the "***Gearbox mode***" the Drive controls the position. The reference position is obtained by multiplying the pulses received by an external encoder for the "***Numerator ratio***" (***index 2200h***) parameter and then dividing the result for the "***Denominator ratio***" (***index 2201h***) parameter. The number of pulse of the reference position obtained corresponds to a physical position following the porportion 65536 pulses=1 turn.

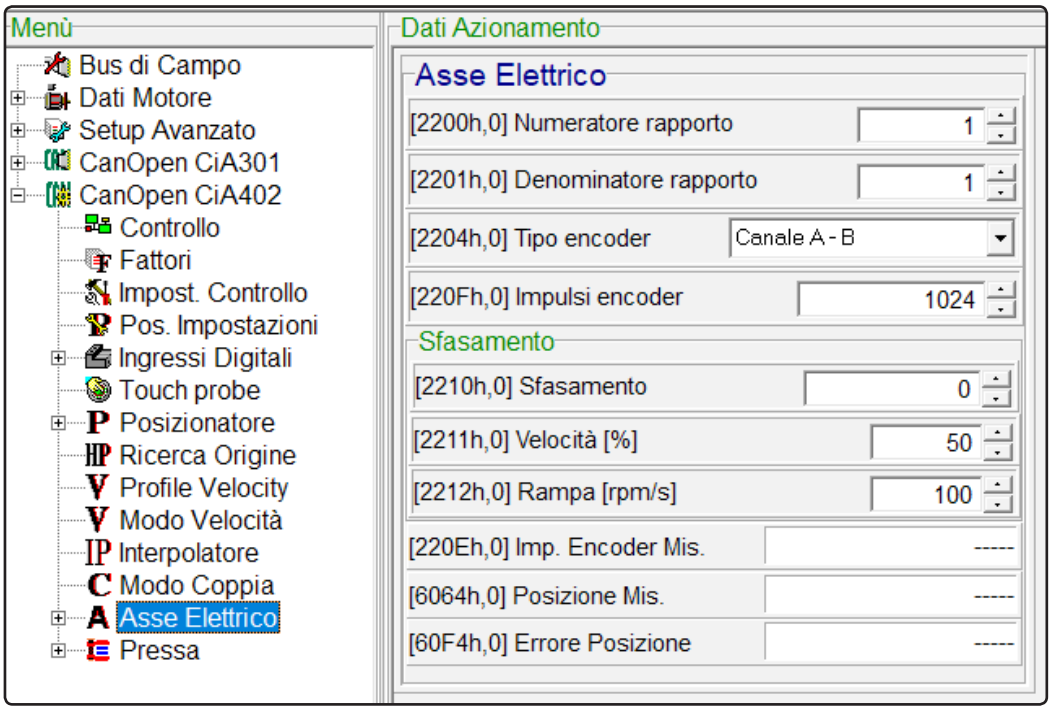
To select the "***Gearbox mode***" the "***Modes of operation***" (***index 6060h***) parameter must be set to -1. The Gearbox function is enabled by switching the ***controlword*** bit 4 to 1. After its activation the engaging phase starts, during which the motor is carried to the same speed of the reference position. When the engaging is completed the Drive follows the reference position obtained by the external encoder, and the parameters "***Pos. error window***" (***index 6065h***) and "***Timeout error***" (***index 6066h***) are enabled for the control of the position error. The Gearbox function is disabled by switching the ***controlword*** bit 4 to 0; this command starts the disengaging function, which has the task to stop the motor in the selected way.

The ***statusword*** bit 12 reports the activation state of the Gearbox function, when is set to 1 it means that this function is active and the Drive is executing one of these 3 phases: engaging phase, position tracking phase or disengaging phase.

PRELIMINARY

Index	Name	Type	Map	Attributes
2200h	Numerator ratio	INT16	Yes	rw
2201h	Denominator ratio	UINT16	Yes	rw
2204h	Encoder type	UINT8	No	rw
220Eh	Meas. imp. Encoder	UINT32	Yes	ro
220Fh	Encoder impulses	UINT32	No	rw
2210h	Phase shift	INT32	No	rw
2211h	shift-Velocity	UINT16	No	rw
2212h	shift- ramp	UINT16	No	rw

The parameters required by this mode can be seen in the Caliper menu "***(EtherCat-)CanOpen CiA402- Gearbox mode***" in which the following screen appears:



4.13.1 Controlword bits setting- Gearbox mode

Bit	Description	
0	***Switch on***	
1	***Enable voltage***	
2	***Quick stop***	
3	***Enable operation***	
4	***Enable axis***	
	value	**Description**
	0 -> 1	The Gearbox engaging function is started
	1 -> 0	The Gearbox disengaging function is started
5	***Reset encoder***	
	value	**Description**
	0-> 1	When the gearbox function is disabled, switching this bit from 0 to 1 resets the counter "***Meas. Imp. encoder***" (***index 220Eh***) .
6	***Reset position***	
	value	**Description**
	0-> 1	When the Gearbox function is disabled, switching this bit from 0 to 1 resets the position value.
7	***Fault reset***	
8	***Halt***	
	value	**Description**
	0	The motor moves according with the Gearbox function
	1	The motor stops with a ramp
9	***Shift***	
	value	**Description**
	0->1	switching this bit from 0 to 1 starts a shifting of the motor axis with respect to the master's encoder, according with the "***Phase shift***" (***index 2210h***) parameter.
10	***Reserved***	
15..11	***Manufacturer specific***	

4.13.2 Meaning of the Statusword bits - Gearbox mode

<table>
<tr><th>Bit</th><th colspan="3">Description</th></tr>
<tr><td>0</td><td colspan="3">Ready to switch on</td></tr>
<tr><td>1</td><td colspan="3">Switched on</td></tr>
<tr><td>2</td><td colspan="3">Operation enable</td></tr>
<tr><td>3</td><td colspan="3">Fault</td></tr>
<tr><td>4</td><td colspan="3">Voltage enabled</td></tr>
<tr><td>5</td><td colspan="3">Quick stop</td></tr>
<tr><td>6</td><td colspan="3">Switch on disabled</td></tr>
<tr><td>7</td><td colspan="3">Warning</td></tr>
<tr><td>8</td><td colspan="3">Shift execution
<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Shift function disabled</td></tr><tr><td>1</td><td>The shift function is running</td></tr></table></td></tr>
<tr><td>9</td><td colspan="3">Remote</td></tr>
<tr><td>10</td><td colspan="3">Halt state
<table><tr><th>Value</th><th>Controlword Halt bit</th><th>Description</th></tr><tr><td rowspan="2">0</td><td>0</td><td>Halt command disabled</td></tr><tr><td>1</td><td>The motor is stationary</td></tr><tr><td rowspan="2">1</td><td>0</td><td>If the motor is still moving the deceleration phase is started</td></tr><tr><td>1</td><td>The motor is stationary</td></tr></table></td></tr>
<tr><td>11</td><td colspan="3">Internal limit active (enabled only when “Enable Position limits” -index 201Ah- is on)
<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>The absolute value of the target position is not limited by “Pos. limits” (index 607Dh)</td></tr><tr><td>1</td><td>The absolute value of the target position is limited by “Pos. limits” (index 607Dh)</td></tr></table></td></tr>
<tr><td>12</td><td colspan="3">Gearbox enabled
<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Gearbox function disabled, the Drive is stopped</td></tr><tr><td>1</td><td>Gearbox function enabled, the Drive could be in one of the following phases: engaging phase, position tracking phase or disengaging phase.</td></tr></table></td></tr>
<tr><td>13</td><td colspan="3">Following Error
<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>No position error detected</td></tr><tr><td>1</td><td>Position error detected</td></tr></table></td></tr>
<tr><td>14</td><td colspan="3">Manufacturer specific</td></tr>
<tr><td>15</td><td colspan="3">Position measuring (enabled by “Input setting” – index 201Bh)
<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>• (index 201Bh bit7=0) continuous mode: the position between the rising and falling edge of the chosen input has been calculated.
• (index 201Bh bit7=1) one shot mode: the two values required to calculates the position between the edges of the chosen input have not yet been acquired.</td></tr><tr><td>1</td><td>• (index 201Bh bit7=0) continuous mode: the position between the rising and falling edge of the chosen input has not yet been calculated.
• (index 201Bh bit7=1) one shot mode: the two values required to calculates the position between the edges of the chosen input have been acquired.</td></tr></table></td></tr>
</table>

4.13.3 Object 2200 h- Numerator ratio

Index	Name	Type	Range	Default	Unit	Map	Attributes
2200h	Numerator ratio	INT16	±7FFF	1		Yes	rw

This parameter sets the numerator of the Gearbox ratio, it's used with the "***Denominator ratio***" (***index 2201h***) parameter to set the speed ratio between the master axis and the slave axis. In this mode the position reference is obtained by multiplying the number of pulses received by the external encoder by the "***Numerator ratio***" (***index 2200h***) and then dividing the result by the "***Denominator ratio***" (***index 2201h***).

4.13.4 Object 2201h - Denominator ratio

Index	Name	Type	Range	Default	Unit	Map	Attributes
2201h	Denominator ratio	UINT16	1..65535	1		Yes	rw

This parameter sets the denominator of the Gearbox ratio (see *"4.13.3 Object 2200 h- Numerator ratio" pag. 108*).

4.13.5 Object 2204h – Encoder type

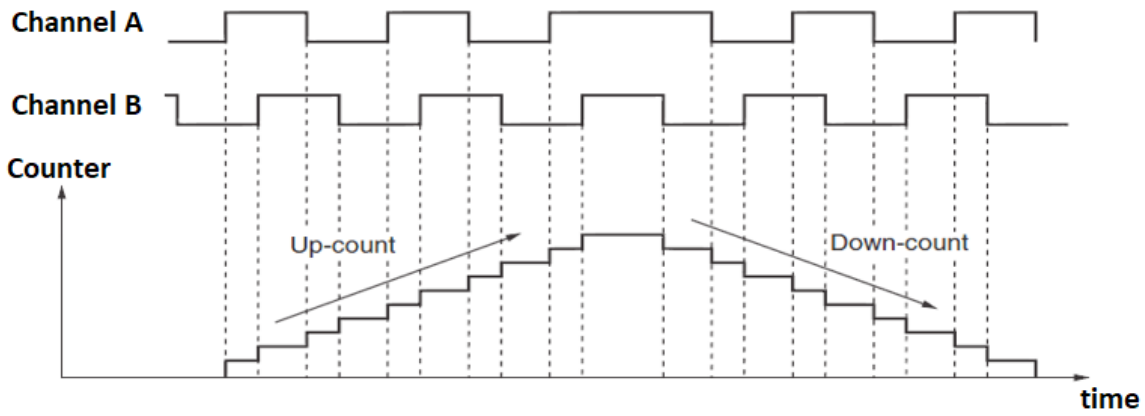
Index	Name	Type	Range	Default	Unit	Map	Attributes
2204h	Encoder type	UINT8	0..2	0		No	rw

This parameter sets how the pulses from the external encoder will be acquired.

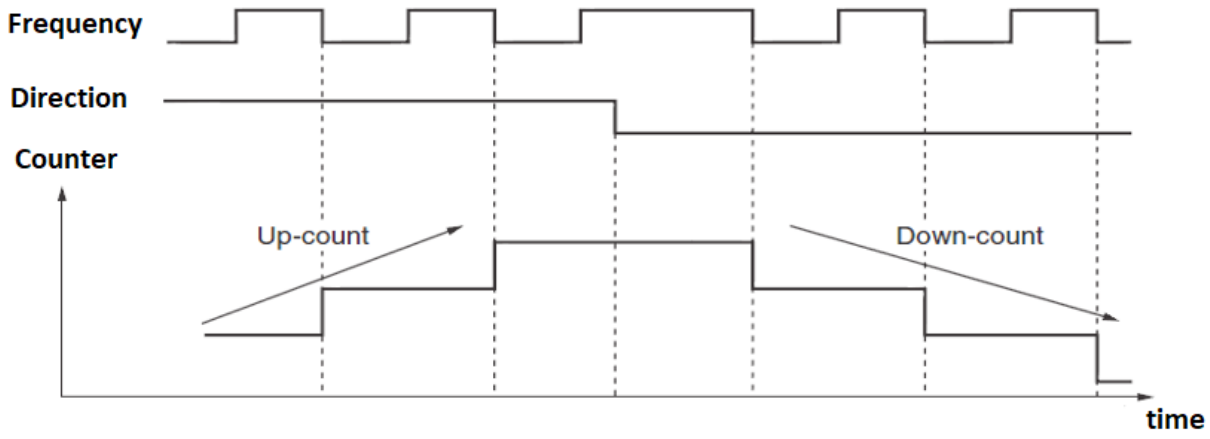
By choosing "***channel A-B***" mode, the Drive is able to count all the pulses of the two encoder channels, therefore the resolution is 4 times higher than the declared encoder resolution. For example, if the encoder emits 1024 pulses per turn, the Drive is able to count 4096 pulses per turn. This must be taken into account before setting the parameters "***Numerator ratio***" (***index 2200h***) and "***Denominator ratio***" (***index 2201h***).

Value	*Encoder type description*
0	***Channel A-B*** : the signal comes from the quadrature signals
1	***Freq-Direction***: One channel sends the direction, the other sends the speed information
2	***Impulses CW-CCW***: Depending on which channel sends the impulses, the motor rotates in one direction or the other

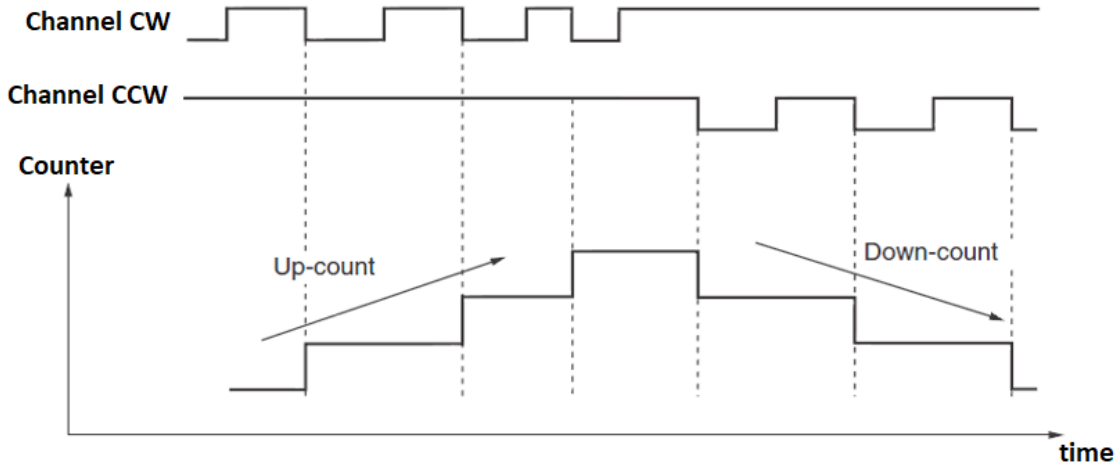
Encoder type "Channel A-B"



Encoder type "Frequency-Direction"



Encoder type "Impulses CW CCW"



4.13.6 Object 220Eh - Meas. Imp. Encoder

Index	Name	Type	Range	Default	Unit	Map	Attributes
220Eh	Meas Imp. Encoder	INT32	±2147483647			Yes	ro

This parameter reports the number of external encoder pulses counted. This value can be reset to 0 by switching bit 5 of the ***controlword*** from 0 to 1, when the gearbox function is disabled.

4.13.7 Object 220Fh - Encoder impulses

Index	Name	Type	Range	Default	Unit	Map	Attributes
220Fh	Encoder impulses	UINT32	100.. 65536	1024		No	rw

The number of pulses per turn of the encoder must be set in this parameter. If a "***channel A-B***" type encoder is used, the value of this parameter is internally multiplied by 4, since with this type the Drive can acquire the encoder pulses with a resolution 4 times greater.

4.13.8 Object 2210h - Phase Shift

Index	Name	Type	Range	Default	Unit	Map	Attributes
2210h	Phase Shift	INT32	±16777215	0		No	rw

This parameter sets the number of pulses used to shift the Slave axis position with respect to the reference of the master encoder. This phase (position) shift is executed when the ***controlword*** bit 9 switches from 0 to 1; the ***statusword*** bit 8 remains high while the motor is shifting.

The phase shifting function makes one revolution every 65536 pulses.

4.13.9 Object 2211h – Shift- velocity

Index	Name	Type	Range	Default	Unit	Map	Attributes
2211h	Shift-velocity	UINT16	1..200	50	%	No	rw

This parameters sets the speed used for the phase shift, expressed as a percentage of the measured speed.

4.13.10 Object 2212h – Shift -Ramp

Index	Name	Type	Range	Default	Unit	Map	Attributes
2212h	Accelerazione sfasamento	UINT16	1..60000	100	rpm/s	No	rw

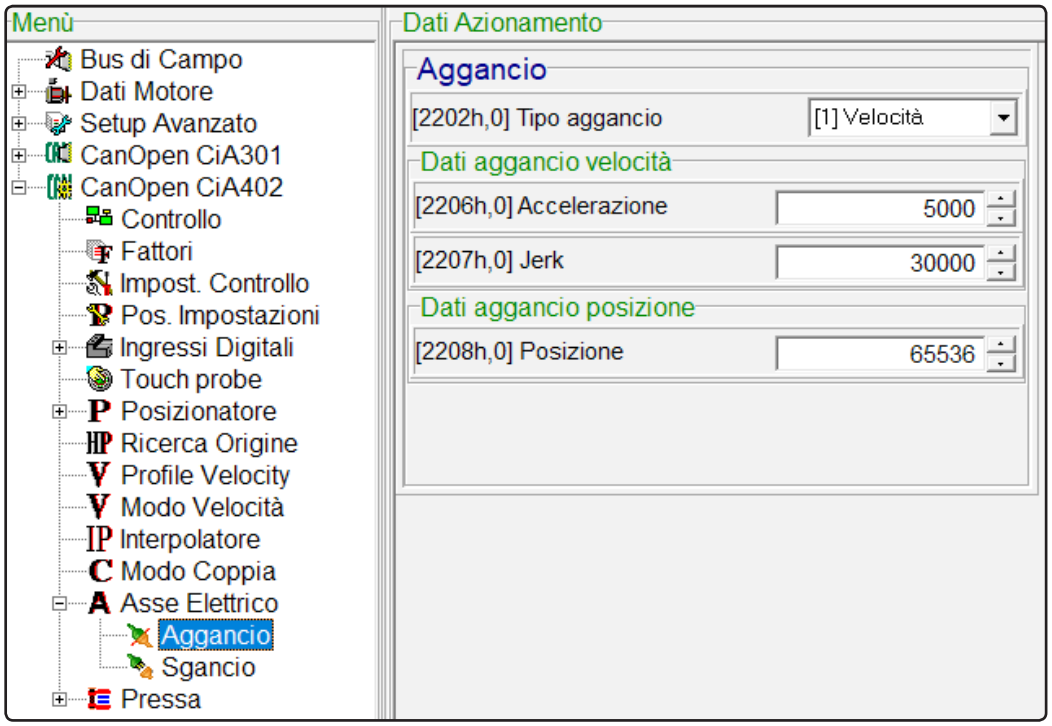
This parameter sets the acceleration applied at the beginning and end of the phase shifting.

4.13.11 Engagement parameters - Gearbox

In this section are reported the parameters used by the engagement function used by the ***Gearbox*** mode. When the ***controlword*** bit 4 switches from 0 to 1 this function starts, and the motor is carried to the same speed of the reference position obtained by multiplying the pulses of thr external encoder by the gearbox ratio. When the engaging phase is terminated the Drive rigidly follows the position reference coming from the external encoder.

Index	Name	Type	Map	Attributes
2202h	Engagement type	UINT8	No	rw
2206h	Engagement - acceleration	UINT32	No	rw
2207h	Engagement - Jerk	UINT32	No	rw
2208h	Engagement - position	UINT32	No	rw

The previous parameters can be seen in the Caliper menu "***(EtherCat-)CanOpen CiA402- Gearbox mode-engaging***" in which the following screen appears:



PRELIMINARY

4.13.11.a Object 2202h – Engagement type

Index	Name	Type	Range	Default	Unit	Map	Attributes
2202h	Engagement type	UINT8	0..2	1		No	rw

This parameter is used to select how the motor will pass from stand still to follow the reference position obtained by multiplying the pulses of the external encoder by the gearbox ratio.

Value	Engagement type	Description
0	Immediate	with this mode the position control is immediately enabled in order to follow the reference position obtained from the external encoder. This choice is suggested if the pulses of the external encoder are null when the gearbox function is enabled (i.e. on the rising edge of bit 4 of the ***controlword***).
1	Velocity	with this mode the motor is carried to the same speed of the reference position with a speed ramp, starting from standstill. This engagement is done according to the parameters "***Engagement - acceleration***" (***index 2206h***) and "***Engagement - jerk***" (***index 2207h***).
2	Position	with this mode, when the ***Gearbox*** function is enabled the motor reaches the same speed of the reference position covering the space set on the parameter "***Engagement - position***" (***index 2208h***). The position profile during this engagement is calculated using a 5th order polynomial curve.

4.13.11.b Object 2206h – Engagement - acceleration

Index	Name	Type	Range	Default	Unit	Map	Attributes
2206h	Engagement - acceleration	UINT32	1..6000000	5000	rpm/s	No	rw

This parameter sets the acceleration of the speed ramp executed during the engaging phase.

4.13.11.c Object 2207h – Engagement - Jerk

Index	Name	Type	Range	Default	Unit	Map	Attributes
2207h	Engagement - Jerk	UINT32	1..6000000	30000	rpm/s^2	No	rw

This parameter sets the acceleration variation of the speed ramp executed during the engaging phase.

4.13.11.d Object 2208h – Engagement - position

Index	Name	Type	Range	Default	Unit	Map	Attributes
2208h	Engagement - position	UINT32	0..2147483647	65536		No	rw

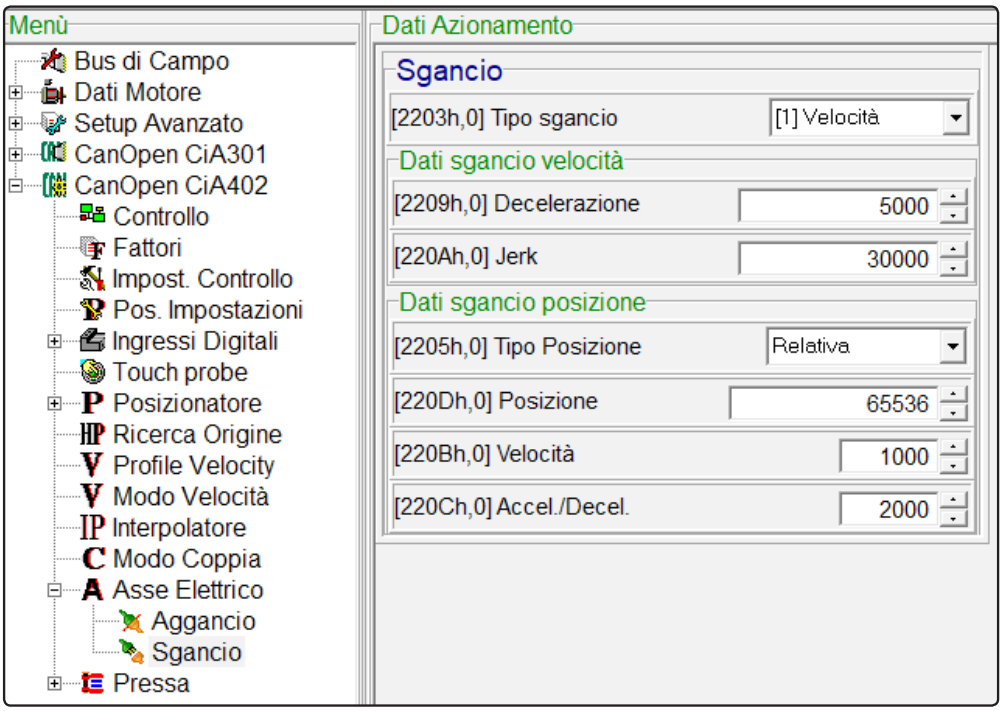
When the "***Engagement type***" (***index 2202h***) is set on "***position***", this parameter sets the space that the axis will cover as it reaches the speed of the position reference. This space corresponds to the one covered by the external encoder during the engagement, multiplied by the gearbox ratio.

4.13.12 Disengagement parameters - Gearbox

In this section are reported the parameters used by the disengagement function used by the ***Gearbox*** mode. When the ***controlword*** bit 4 switches from 1 to 0 this function starts, and the motor is stopped in the selected mode. When the disengaging phase is terminated the bit 6 of the ***Statusword*** is put to 0 and the motor remains stationary with the power on.

Index	Name	Type	Map	Attributes
2203h	Disengagement type	UINT8	No	rw
2205h	Position type	UINT8	No	rw
2209h	disengagement-Deceleration	UINT32	No	rw
220Ah	disengagement-Jerk	UINT32	No	rw
220Bh	disengagement-Velocity	UINT16	No	rw
220Ch	disengagement-Accel./Decel.	UINT16	No	rw
220Dh	disengagement-position	INT32	No	rw

The previous parameters can be seen in the Caliper menu "***(EtherCat-)CanOpen CiA402- Gearbox mode-disengaging***" in which the following screen appears:



4.13.13 Object 2203h – Disengagement type

Index	Name	Type	Range	Default	Unit	Map	Attributes
2203h	Disengagement type	UINT8	0..2	1		No	rw

The "***Disengagement type***" is used to select how the Drive will go from following the position reference to stopping the motor. This passage is done when the ***Gearbox*** function is disabled (i.e. when the bit 4 of the ***controlword*** is switched from 1 to 0).

Value	Type	Description
0	Immediate	with this mode, when the ***Gearbox*** function is disabled the axis stops immediately. *Note: this stop is very abrupt, the motor stops using all this torque,causing high mechanical stress.*
1	velocity	with this mode the motor stops with a speed ramp. The disengagement is done according with the parameters "***disengagement-deceleration***" (***index 2209h***) and "***disengagement-jerk***" (***index 220Ah***).
2	position	With this mode the motor stops following a position profile obtained by a 4th order polynomial function.The disengagement is done according with the parameters "***Position type***" (***index 2205h***), "***disengagement-velocity***" (***index 220Bh***), "***disengagement-accel./decel.***" (***index 220Ch***) and "**disengagement-position**" (***index 220Dh***). If the "***disengagement-position***" is an absolute value, it sets the final position where the motor stops; if it's a relative value, it sets the space covered by the motor during the disengagement phase.

PRELIMINARY

4.13.13.a Object 2205h – Position type

Index	Name	Type	Range	Default	Unit	Map	Attributes
2205h	Position type	UINT8	0..1	1		No	rw

This parameter sets the type of the "***disengagement-position***": if it's an absolute value, it sets the final position where the motor stops; if it's a relative value, it sets the space covered by the motor during the disengagement phase.

4.13.13.b Object 2209h - Disengagement-Deceleration

Index	Name	Type	Range	Default	Unit	Map	Attributes
2209h	Disengagement-Deceleration	UINT32	1..6000000	5000	rpm/s	No	rw

This parameter sets the deceleration of the speed ramp used to stop the motor during a "***velocity***" type disengagement.

4.13.13.c Object 220Ah - Disengagement-Jerk

Index	Name	Type	Range	Default	Unit	Map	Attributes
220Ah	Disengagement-Jerk	UINT32	1..6000000	30000	rpm/s^2	No	rw

This parameter sets the deceleration variation of the speed ramp used to stop the motor during a "***velocity***" type disengagement.

4.13.13.d Object 220Bh - Disengagement-Velocity

Index	Name	Type	Range	Default	Unit	Map	Attributes
220Bh	Disengagement-Velocity	UINT16	1..10000	1000	rpm	No	rw

This parameter sets the maximum speed that could be reached during a "***position***" type disengagement.

4.13.13.e Object 220C - disengagement-Accel./Decel.

Index	Name	Type	Range	Default	Unit	Map	Attributes
220Ch	Disengagement-Accel./Decel.	UINT16	1..60000	2000	rpm/s	No	rw

This parameter sets the maximum deceleration that could be reached by the motor during a "***position***" type disengagement.

4.13.13.f Object 220D - disengagement-position

Index	Name	Type	Range	Default	Unit	Map	Attributes
220Dh	Disengagement-position	INT32	±2147483647	65536		No	rw

This parameter sets a position value that can be considered absolute or relative, depending on the mode selected with "***Position type***" (***index 2205h***). If it is considered an absolute value, the motor will stop in this exact position at the end of the disengagement phase. If it is considered a relative value, the motor will stop in the position obtained by adding this value with the position measured at the start of the disengagement.

Note: the measured position value can be reset to 0 by switching the bit 6 of the controlword from 0 to 1, when the gearbox function is disabled.

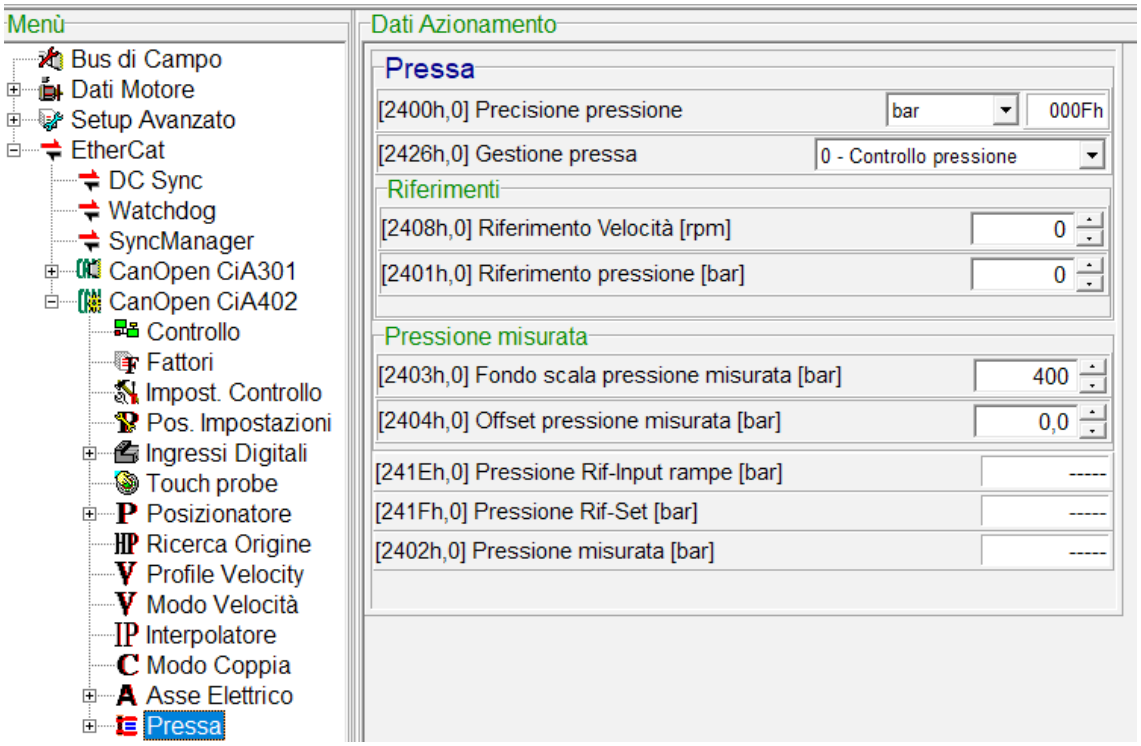
PRELIMINARY

4.14 Press mode

The ***Press*** mode is selected when the parameter "***Modes of operation***" (index 6060h) is set to -2. With this mode, in its ordinary functioning, the Drive regulates the speed of a servopump in order to bring its pressure to the reference value.This is its normal usage, wich is applied when the bit 5 of the ***Controlword*** is set on 0. However, if the bit 5 of the ***Controlword*** is set on 1, then the Drive will actually switch to a Speed control, without changing the parameter "***Modes of operation***" (index 6060h). This particular usage could be useful if you want to control with one Drive both the servopump of an hydraulic unit in ***Press*** mode and another motor in ***Speed*** mode, connecting the Drive first with one, and then with the other (you can't control both motors with the two previously descripted modes at the same time).

Index	Name	Type	Map	Attributes
2400h	Press - flag	UINT16	No	rw
2401h	Pressure reference	UINT16	Yes	rw
2402h	Measured pressure	UINT16	Yes	ro
2403h	Maximum measured pressure	UINT16	No	rw
2404h	Offset measured pressure	UINT16	No	rw
2408h	Speed reference	UINT16	Yes	rw
241Eh	Pressure ref-input ramps	UINT16	Yes	ro
241Fh	Pressione ref-set	UINT16	Yes	ro

The previous parameters can be seen in the Caliper menu "***(EtherCat-)CanOpen CiA402-Press***" in which the following screen appears:



PRELIMINARY

4.14.1 Controlword bits setting- Press mode

Bit	Description
0	***Switch on***
1	***Enable Voltage***
2	***Quick stop***
3	***Enable operation***
4	***Reserved***
5	***Set mode Press/Vel***
6	***Enable setpoint***
7	***Fault reset***
8	***Halt***
9	***Ctrl-Press ON***
10	***Reserved***
15 ..11	***Manufacturer specific***

4.14.2 Meaning of the Statusword bits - Press mode

bit	description
0	***Ready to Switch on***
1	***Switched on***
2	***Operation enable***
3	***Fault***
4	***Voltage enabled***
5	***Quick stop***
6	***Switch on disabled***
7	***Warning***
8	***Manufacturer specific***
9	***Remote***
10	***Pressure reached***
11	***Underpressure***
12	***Overpressure***
13	***Power-limit***
14	***Mode Press-Velocity***
15	***Manufacturer specific***

PRELIMINARY

4.14.2.a Object 2400h - Press-flag

Index	Name	Type	Range	Default	Unit	Map	Attributes
2400h	Press-flag	UINT16	0..65535			No	rw

This object contains some flags used when the the control type is set on ***Press***. The list of this flags is reported in the table below:

bit	description	
0	***Enables speed ramp***	
	Value	**Description**
	0	Speed ramps disabled
	1	Speed ramps enabled
1	***Enables pressure ramp***	
	Value	**Description**
	0	Pressure ramps disabled
	1	Pressure ramps enabled
2	***Overpressure alarm***	
	Value	**Description**
	0	Overpressure alarm disabled
	1	Overpressure alarm enabled
3	***Underpressure alarm***	
	Value	**Description**
	0	Underpressure alarm disabled
	1	Underpressure alarm enabled
4	***Pressure accuracy***	
	Value	**Description**
	0	precision of 1 bar
	1	precision of 1/10 of bar
5 ...15	***Manufacturer specific***	

4.14.2.b Object 2401h - Pressure reference

Index	Name	Type	Range	Default	Unit	Map	Attributes
2401h	Pressure reference	UINT16	0... 1000	0	bar	Yes	rw

This object is used to set the pressure reference, in [Bar], that the servopump must reach.

4.14.2.c Object 2402h - Measured pressure

Index	Name	Type	Range	Default	Unit	Map	Attributes
2402h	Measured pressure	INT16	-32768...32767		bar	Yes	ro

This object contains the value of the pressure measured by the transducer. The transducer converts the pressure in a voltage proportional with its value, which is read by the main analogical input of the Drive. The Drive will then calculate the value in [Bar] of the measured pressure using a proportion with the parameter "***Maximum measured pressure***" (***index 2403h***).

4.14.2.d Object 2403h - Maximum measured pressure

Index	Name	Type	Range	Default	Unit	Map	Attributes
2403h	Maximum measured pressure	UINT16	0... 6000	400	bar	No	rw

This object contains the full scale of pressure, in [Bar]. The transducer converts the measured pressure in a voltage proportional with its value, which is read by the main analogical input of the Drive.The Drive will then calculate the value in [Bar] of the measured pressure using a proportion,assuming that a voltage of 10 [V] corresponds to the pressure set on this object.

The formula used to convert the measured pressure in [Bar] is shown below:

$$measured\_pressure = \frac{Maximum\_pressure}{10} * volt\_input$$

Note: we assume that the pressure of 0 [bar] corresponds to a voltage of 0 [V], therefore we are using only half of the measurable range of voltage, which ranges from -10 [V] to + 10 [V]

4.14.2.e Object 2404h - Offset measured pressure

Index	Name	Type	Range	Default	Unit	Map	Attributes
2404h	Offset measured pressure	INT16	-3000... 3000	0	bar	No	rw

This object contains a constant value which is added to (or subtracted from) the value of the measured pressure. It's used to eliminate the potential offset of the transducer.

4.14.2.f Object 2408h - Speed reference

Index	Name	Type	Range	Default	Unit	Map	Attributes
2408h	Speed reference	INT16	-3000... 3000	0	rpm	Si	rw

This object contains a value of speed expressed in [rpm]. When the bit 5 of the ***Controlword*** is set to 0,this object represents the maximum limit that the speed produced by the pressure regulator cannot exceed. When the bit 5 of the ***Controlword*** is set to 1,this object represents the speed reference that the Drive is trying to reach; in this case the control is changed in fact from ***Press*** mode to ***Speed*** mode.

4.14.2.g Object 241Eh - Pressure ref-input ramps

Index	Name	Type	Range	Default	Unit	Map	Attributes
241Eh	Pressure ref- input ramps	INT16	-32768... 32767		bar	Yes	ro

This object contains the value of the pressure reference, in [Bar], that the Drive is trying to reach. If the Drive is controlled with a Fieldbus, then this parameter has the same value of "***Pressure reference***" (***index 2401h***). Instead,if the Drive is controlled with the Input/Output remote control, then the value of this parameter will corresponds to the signal received by the analogical input. If the pressure ramps are enabled, this pressure reference will be "filtered" prior to being passed to the pressure regulator, in order to prevent it from any sharp variation.

4.14.2.h Object 241Fh - Pressure ref-set

Index	Name	Type	Range	Default	Unit	Map	Attributes
241Fh	Pressure ref-set	INT16	-32768... 32767		bar	Yes	ro

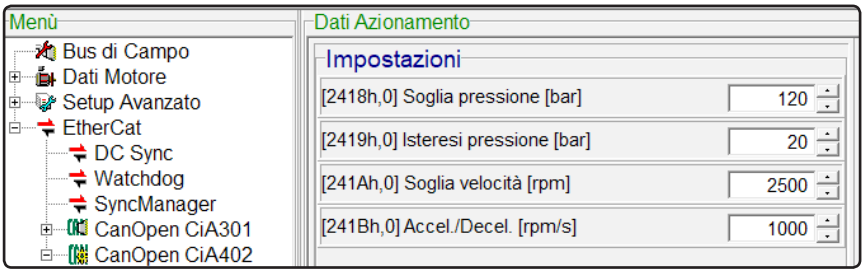
This obejct contains the pressure reference value after it has been "filtered" by the pressure ramps. These ramps are used to avoid the speed reference from changing abruptly when it's enabled or disabled .

4.14.3 Setting

When the Drive is working in ***Press*** mode it's possible to limit the maximum power that the servopump could absorb. In oder to set this limit of power you must use the parameters listed in the table below. Note that the power is never explicitly mentioned, because the Drive works with a value proportional with it obtained by the product of pressure and speed, considering that the latter is proportional with the flow rate of the pump.

Index	Name	Type	Map	Attributes
2418h	Pressure threshold	UINT16	No	rw
2419h	Pressure hysteresis	UINT16	No	rw
241Ah	Speed threshold	UINT16	No	rw
241Bh	Accel./ decel.	UINT16	No	rw

These parameters can be set in the Caliper menu "***(EtherCat-)CanOpen CiA402-Press-setting***" in which the following screen appears:



4.14.3.a Object 2418h - Pressure threshold

Index	Name	Type	Range	Default	Unit	Map	Attributes
2418h	Pressure threshold	UINT16	0.. 6000	120	bar	No	rw

This object contains a pressure threshold. If the measured pressure surpasses the ***Pressure threshold*** set here, and the measured speed surpasses the "***Speed threshold***" (***index 241Ah***) at the same time, then the Power limit will be activated. When this limit is enabled the maximum speed that the motor cannot surpass is the one set on "***Speed threshold***".

4.14.3.b Object 2419h - Pressure hysteresis

Index	Name	Type	Range	Default	Unit	Map	Attributes
2419h	Pressure Hysteresis	UINT16	1.. 50	20	bar	No	rw

This object contains a pressure value, named "***Pressure Hysteresis***". If the Power limit for the ***Press*** mode is enabled,in order to disable it is necessary that the measured pressure or the measured speed or both decrease under a certain threshold. The measured speed must go under the "***Speed threshold***" (***index 241Ah***); the measured pressure must go under a value lower than the "***Pressure threshold***" (***index 2418h***), obtained by decreasing the "***Pressure threshold***" by the "***Pressure Hysteresis***" value.

4.14.3.c Object 241Ah - Speed threshold

Index	Name	Type	Range	Default	Unit	Map	Attributes
241Ah	Speed threshold	UINT16	0.. 30000	2500	rpm	No	rw

This object contains a speed threshold. If the measured pressure surpasses the "***Pressure threshold***" (***index 2418h***) and the measured speed surpasses the "***Speed threshold***" set here at the same time, then the Power limit will be activated. When this limit is enabled the motor maximum speed cannot surpass the value set here.

4.14.3.d Object 241Bh - Accel./decel.

Index	Name	Type	Range	Default	Unit	Map	Attributes
241Bh	Accel./decel.	UINT16	1.. 60000	1000	rpm/s	No	rw

The function that manages the speed ramps keeps working even when the Drive has reached the power limi, in order to avoid sudden changes of the speed limit at the output of the pressure regulator. This object sets the acceleration and deceleration applied during the speed ramps.

4.14.4 Limits

This paragraph lists the parameters used to set additional limits for the Drive when it's working on ***Press*** mode.

Index	Name	Type	Map	Attributes
2006h	Current limit	UINT16	No	rw
2409h	Negative speed limit	INT16	Yes	rw

These parameters can be set in the Caliper menu "***(EtherCat-)CanOpen CiA402-Press-Limit***" in which the following screen appears:

4.14.4.a Object 2006h – Current limit

Index	Name	Type	Range	Default	Unit	Map	Attributes
2006h	Current limit	UINT16	0.. 1000	200	%	No	rw

This object is used to set the upper limit of the absolute value of the current, expressed as parts per thousand (‰) of the nominal current of the motor;for example, if we have a nominal current of 2,00 [A] and the current limit is set on 2000, then the maximum current allowed will be equal to 4,00 [A].

4.14.4.b Object 2409h - Negative speed limit

Index	Name	Type	Range	Default	Unit	Map	Attributes
2409h	Negative speed limit	INT16	-30000.. 0	-50	rpm	Yes	rw

This object contains the maximum limit of speed at which the servopump can rotate in the opposite direction.

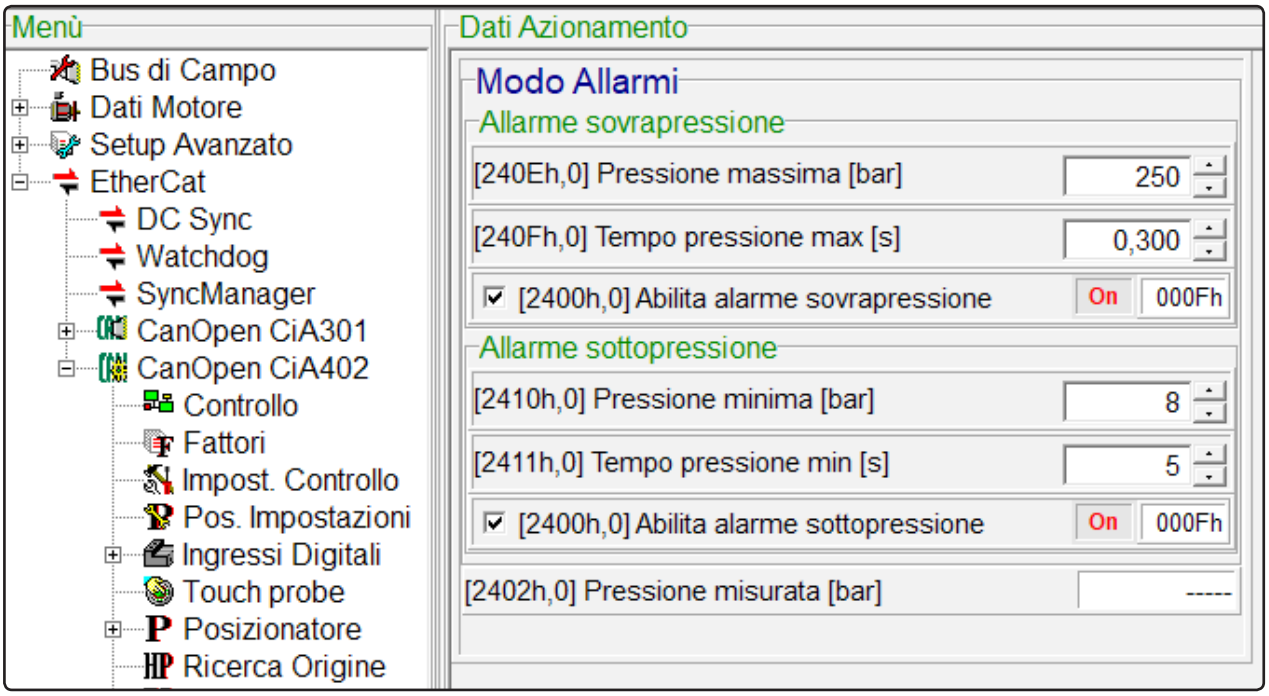
PRELIMINARY

4.14.5 Alarm mode

This paragraph lists the parameters used to set the "***Underpressure alarm***" and the "***Overpressure alarm***" when the Drive is working with the ***Press*** mode.

Index	Name	Type	Map	Attributes
240Eh	**Maximum pressure**	UINT16	Yes	rw
240Fh	**Time pressure max**	UINT16	Yes	rw
2410h	**Minimum pressure**	UINT16	Yes	rw
2411h	**Time pressure min**	UINT16	Yes	rw

These parameters can be set in the Caliper menu "***(EtherCat-)CanOpen CiA402-Press-Alarm mode***" in which the following screen appears:



4.14.5.a Object 240Eh - Maximum pressure

Index	Name	Type	Range	Default	Unit	Map	Attributes
240Eh	Maximum pressure	UINT16	1.. 6000	250	bar	Yes	rw

This object contains the upper limit of the pressure, expressed in [Bar]. If the measured pressure surpass this limit for a period of time longer than the one set on the parameter "***Time pressure max***" (***index 240Fh***), then Drive will signal the "***Overpressure Alarm-FA30***".

4.14.5.b Object 240Fh - Time pressure max.

Index	Name	Type	Range	Default	Unit	Map	Attributes
240Fh	Time pressure max	UINT16	0,001.. 60	0,3	s	Yes	rw

This object contains a time period, expressed in [ms]. If the measured pressure surpass the "***Maximum pressure***" limit (***index 240Eh***) for a period of time longer than the one set on this parameter, then the Drive will signal the "***Overpressure Alarm-FA30***".

4.14.5.c Object 2410h - Minimum pressure

Index	Name	Type	Range	Default	Unit	Map	Attributes
2410h	Minimum pressure	UINT16	1.. 6000	8	bar	Yes	rw

This object contains the lower limit of the pressure, expressed in [Bar]. If the measured pressure drops below this limit for a period of time longer than the one set on the parameter "***Time pressure min***" (***index 2411h***),then Drive will signal the "***Underpressure Alarm-FA31***".

4.14.5.d Object 2411h - Time minimum pressure

Index	Name	Type	Range	Default	Unit	Map	Attributes
2411h	Time pressure min.	UINT16	1.. 60000	5	s	Yes	rw

This object contains a time period, expressed in [ms]. If the measured pressure drops below the "***Minimum pressure***" limit (***index 2410h***) for a period of time longer than the one set on this parameter, then the Drive will signal the "***Underpressure Alarm-FA31***".

4.14.6 Pressure regulation

The pressure is regulated using a PID. The proportional, integral and derivative gains of this PID are calculated as shown in the following formula:

$$Gain_x = K_x 2^{Exponent}$$

There is a constant value ***Kx*** and an ***Exponent*** for each of the PID's gains. We opted for this solution because it allows to make a coarse regulation by changing the exponents, and a fine regulation by changing the constant values.

Index	Name	Type	Map	Attributes
2412h	**Press-Kp**	UINT16	No	rw
2413h	**Press-Ki**	UINT16	No	rw
2414h	**Press-Kd**	UINT16	No	rw
2415h	**Esp Kp**	INT16	No	rw
2416h	**Esp Ki**	INT16	No	rw
2417h	**Esp Kd**	INT16	No	rw

The constant values ***Kx*** and the ***Exponents*** used to calculate this 3 gains can be set in the Caliper menu "***(EtherCat-) CanOpen CiA402-Press - Pressure regulator***" in which the following screen appears:

4.14.6.a Object 2412h- Press-Kp

Index	Name	Type	Range	Default	Unit	Map	Attributes
2412h	Press-Kp	UINT16	1 ... 32767	450		No	rw

This object contains the constant value ***Kx*** of the proportional gain of the Pressure regulator, see formula on paragraph *"4.14.6 Pressure regulation" pag. 121*

4.14.6.b Object 2413h - Press-Ki

Index	Name	Type	Range	Default	Unit	Map	Attributes
2413h	Press-Ki	UINT16	0 ... 32767	5		No	rw

This object contains the constant value ***Kx*** of the integral gain of the Pressure regulator, see formula on paragraph *"4.14.6 Pressure regulation" pag. 121*

4.14.6.c Object 2414h - Press-Kd

Index	Name	Type	Range	Default	Unit	Map	Attributes
2414h	Press-Kd	UINT16	0 ... 32767	0		No	rw

This object contains the constant value ***Kx*** of the derivative gain of the Pressure regulator, see formula on paragraph *"4.14.6 Pressure regulation" pag. 121*

4.14.6.d Object 2415h - Esp Kp

Index	Name	Type	Range	Default	Unit	Map	Attributes
2415h	Esp Kp	UINT16	-16 ... 16	9		No	rw

This object contains the ***Exponent*** of the proportional gain of the Pressure regulator, see formula on paragraph *"4.14.6 Pressure regulation" pag. 121*

4.14.6.e Object 2416h - Esp Ki

Index	Name	Type	Range	Default	Unit	Map	Attributes
2416h	Esp Ki	UINT16	-16 ... 16	5		No	rw

This object contains the ***Exponent*** of the integral gain of the Pressure regulator, see formula on paragraph *"4.14.6 Pressure regulation" pag. 121*

4.14.6.f Object 2417h - Esp Kd

Index	Name	Type	Range	Default	Unit	Map	Attributes
2417h	Esp Kd	UINT16	-16 ... 16	4		No	rw

This object contains the ***Exponent*** of the derivative gain of the Pressure regulator, see formula on paragraph *"4.14.6 Pressure regulation" pag. 121*

PRELIMINARY

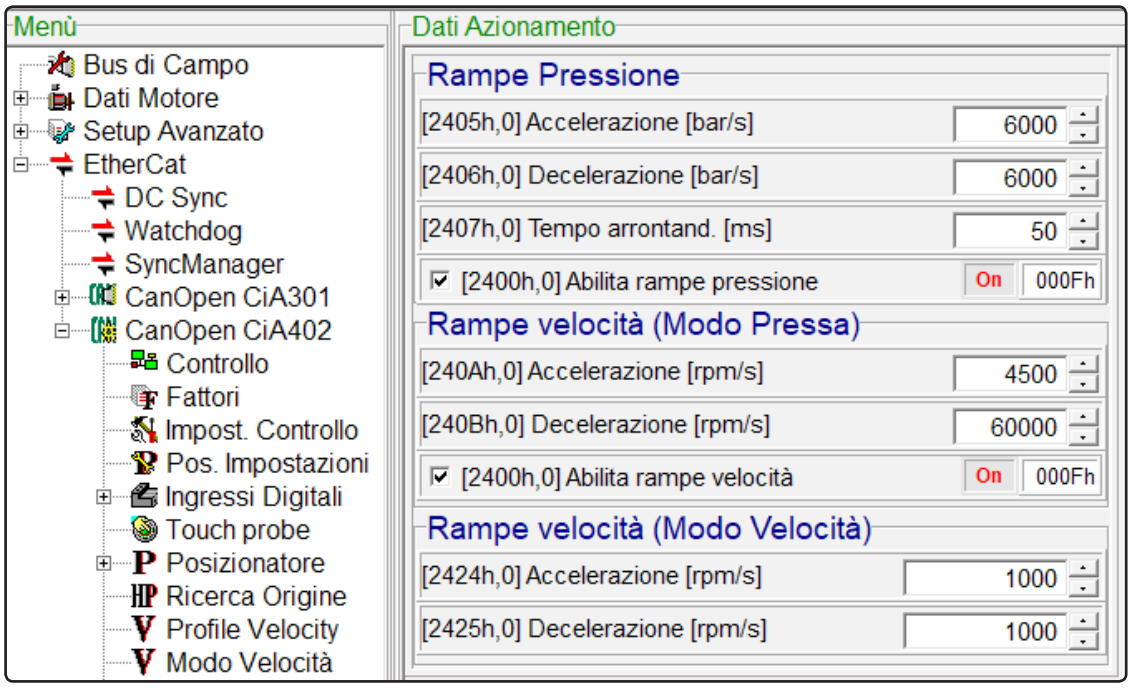
4.14.7 Ramps

This paragraph lists the parameters used to set the pressure ramp and the speed ramp. These ramps are used to smooth the abrupt variations in the pressure reference and in the upper limit of speed at the exit of the pressure regulator , which could occur when these values are inserted, removed or modified.

Index	Name	Type	Map	Attributes
2405h	**Pressure ramps - acceleration**	UINT16	Yes	rw
2406h	**Pressure ramps - deceleration**	UINT16	Yes	rw
2407h	**Pressure ramps - Rounding time**	UINT16	Yes	rw
240Ah	**Speed ramps - acceleration**	UINT16	Yes	rw
240Bh	**Speed ramps - deceleration**	UINT16	Yes	rw
2424h	**Speed mode - acceleration**	UINT32	No	rw
2425h	**Speed mode - deceleration**	UINT32	No	rw

The objects 2424h and 2425h are used when the bit 5 of the ***Controlword*** is set to 1, their task is to smooth the speed reference of the Drive when the latter switched from the ***Press*** mode to the ***Speed*** mode (see *"4.14 Press mode" pag. 114*).

These parameters can be set in the Caliper menu "***(EtherCat-)CanOpen CiA402-Press-Ramps***" in which the following screen appears:



4.14.7.a Object 2405h - Pressure ramps - acceleration

Index	Name	Type	Range	Default	Unit	Map	Attributes
2405h	Pressure ramps - acceleration	UINT16	1 ... 60000	6000	bar/s	Yes	rw

This object is used to set the gradient of the rising phase of the pressure ramp. Using this ramp, when the pressure reference is increased (or entered) it will not rise immediately, instead it will rise with the gradient set here. This value is expressed in [bar/s].

4.14.7.b Object 2406h- Pressure ramps - deceleration

Index	Name	Type	Range	Default	Unit	Map	Attributes
2406h	Pressure ramps-deceleration	UINT16	1 ... 60000	6000	bar/s	Yes	rw

This object is used to set the gradient of the descending phase of the pressure ramp. Using this ramp, when the pressure reference is decreased (or removed) it will not drop immediately, instead it will decrease with the gradient set here. This value is expressed in [bar/s].

4.14.7.c Object 2407h - Pressure ramps - Rounding time

Index	Name	Type	Range	Default	Unit	Map	Attributes
2407h	Pressure ramps - Rounding time	UINT16	0 ... 10000	50	ms	Yes	rw

This object is used to set a period of time, expressed in [ms]. At the end of the rising phase or the descending phase of the pressure ramp the pressure reference is smoothed to cancel its discontinuity; this smoothing curve will last for the period of time set on this parameter.

4.14.7.d Object 240Ah - Speed ramps - acceleration

Index	Name	Type	Range	Default	Unit	Map	Attributes
240Ah	Speed ramps-acceleration	UINT16	1 ... 60000	4500	rpm/s	Yes	rw

This object contains the acceleration applied on the rising phase of the speed ramp; when the speed reference is increased (or inserted) it will not rise immediately, instead it will rise with the acceleration set here.

4.14.7.e Object 240Bh - Speed ramps - deceleration

Index	Name	Type	Range	Default	Unit	Map	Attributes
240Bh	Speed ramps - deceleration	UINT16	1 ... 60000	60000	rpm/s	Yes	rw

This object contains the deceleration applied on the descending phase of the speed ramp; when the speed reference is decreased (or removed) it will not drop immediately, instead it will decrease with the deceleration set here.

4.14.7.f Object 2424h - Speed mode - acceleration

Index	Name	Type	Range	Default	Unit	Map	Attributes
2424h	Speed mode - acceleration	UINT32	0 ... 7FFFFFFFh	1000	rpm/s	No	rw

This parameter sets the acceleration applyed when the Drive is set on Speed mode and the ramps are enabled. This value is converted to the internal unit of measure of the Drive (pulses/s) by using the "***Acceleration factor***" (***index 6097h***) as described in *"4.05.10 Object 6097h - Acceleration factor" pag. 63*. By using the default values of the "***acceleration factor***", i.e. ***Numerator***=65536 and ***Divisor***=60, this acceleration is expressed in [rpm/s].

4.14.7.g Object 2425h - Speed mode - deceleration

Index	Name	Type	Range	Default	Unit	Map	Attributes
2425h	Speed mode - deceleration	UINT32	0 ... 7FFFFFFFh	1000	rpm/s	No	rw

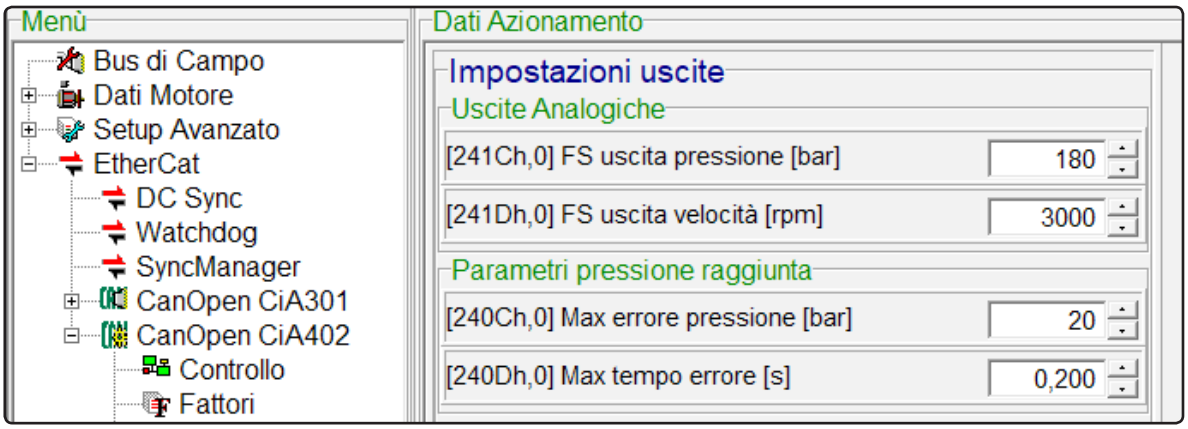
This parameter sets the deceleration applyed when the Drive is set on Speed mode and the ramps are enabled. This value is converted to the internal unit of measure of the Drive (pulses/s) by using the "***Acceleration factor***" (***index 6097h***) as described in *"4.05.10 Object 6097h - Acceleration factor" pag. 63*. By using the default values of the "***acceleration factor***", i.e. ***Numerator***=65536 and ***Divisor***=60, this deceleration is expressed in [rpm/s].

4.14.8 Outputs

The parameters listed below are used to set the full scale of the analogic outputs of speed and pressure. They're also used to signal through the ***Statusword*** if the pressure reference has been reached.

Index	Name	Type	Map	Attributes
240Ch	**Max pressure error**	UINT16	Yes	rw
240Dh	**Max time error**	UINT16	Yes	rw
241Ch	**FS pressure out**	UINT16	No	rw
241Dh	**FS speed out**	UINT16	No	rw

These parameters can be set in the Caliper menu "***(EtherCat-)CanOpen CiA402-Press-Outputs***" in which the following screen appears:



PRELIMINARY

4.14.8.a Object 240Ch - Max pressure error

Index	Name	Type	Range	Default	Unit	Map	Attributes
240Ch	Max pressure error	UINT16	1 ... 6000	20	bar	Yes	rw

This object contains a pressure value,named "***Max pressure error***". If the difference between the measured pressure and the pressure reference is smaller than this value for a period of time longer than the one set on "***Max time error***" (***index 240Dh***), the bit 5 of the ***Statusword*** is switched to 1; otherwise is switched to 0.

4.14.8.b Object 240Dh - Max time error

Index	Name	Type	Range	Default	Unit	Map	Attributes
240Dh	Max time error	UINT16	1 ... 60000	200	s/ 1000	Yes	rw

This object contains a time period,named "***Max time error***". If the difference between the measured pressure and the pressure reference is smaller than "***Max pressure error***" (***index 240Ch***) for a period of time longer than "***Max time error***", the bit 5 of the ***Statusword*** is switched to 1; otherwise is switched to 0.

4.14.8.c Object 241Ch - FS pressure out

Index	Name	Type	Range	Default	Unit	Map	Attributes
241Ch	FS pressure out	UINT16	0 ... 6000	180	bar	No	rw

This object contains the full scale of pressure (expressed in [Bar]) applied to the analog output which sends the measured pressure. This value is matched with the maximum voltage of the analog output, i.e. 10 [V]. The other values of pressure are converted to voltage using a proportion.

4.14.8.d Object 241Dh - FS speed out

Index	Name	Type	Range	Default	Unit	Map	Attributes
241Dh	FS speed out	UINT16	0 ... 30000	3000	rpm	No	rw

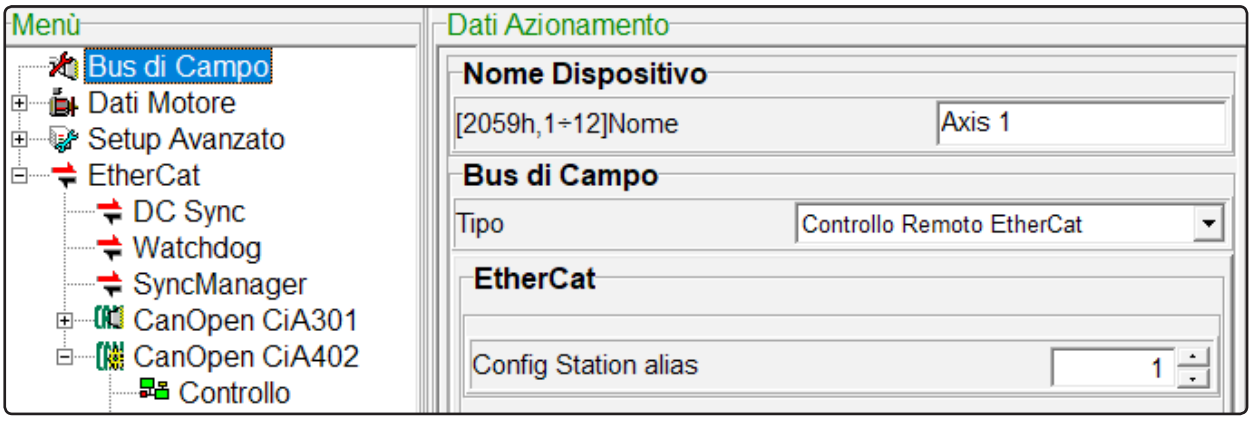
This object contains the full scale of speed (expressed in [rpm]) applied to the analog output which sends the measured speed. This value is matched with the maximum voltage of the analog output, i.e. 10 [V]. The other values of speed are converted to voltage using a proportion.

Ch. 5 INTERNAL PARAMETERS OF THE DRIVE

This chapters lists some specific parameters concerning the Drive. These parameters are not involved with the type of application that you want to use, instead they describe how the Drive operates, how it is identified, how its regulators are tuned, what kind of feedback it use, when it will go in alarm, etc.

5.01 Drive's name

The parameter "***Drive's name***" can be set with the Caliper in the menù "***Field Bus***", where the following screen appears:



Index	Sub-index	Name	Type	Range	Default	Unit	Map	Attributes
259Ch	0...11	Drive's name	UINT16	0 ... 65536			No	rw

This parameter allows you to change the name of a specific Drive. When you're connected to a group of Drives via a HUB USB this feature allows you to name each of them as you like, making it easier to distinguish them. The size of this parameter cannot exceed 12 Bytes.

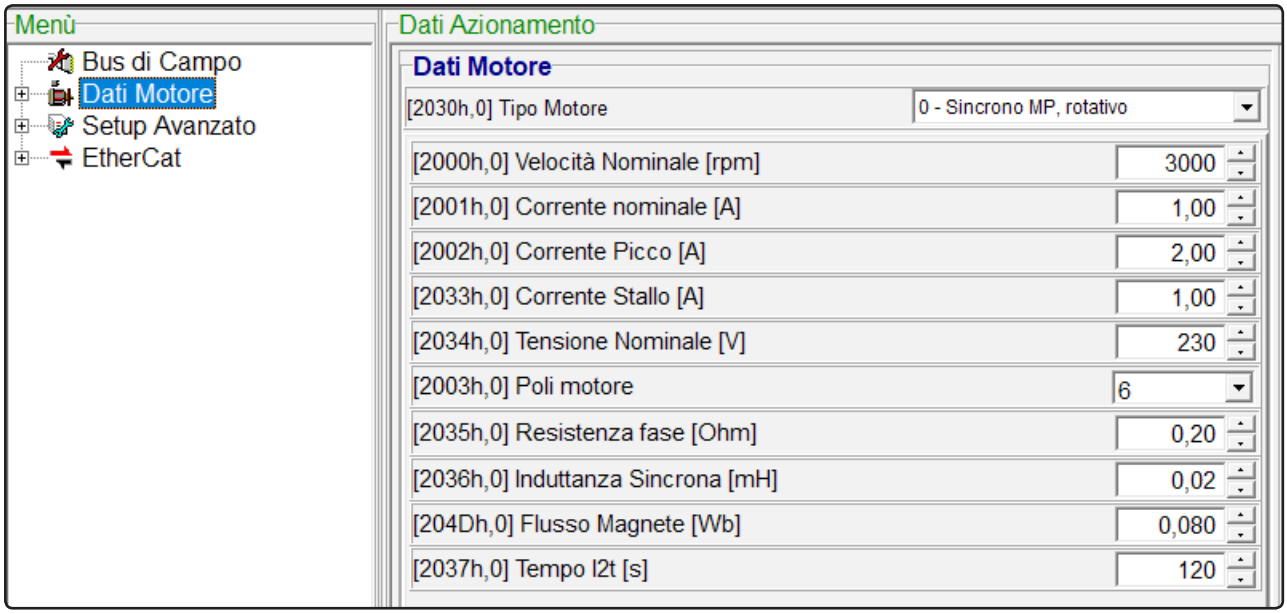
5.02 Motor data

This paragraph lists the parameters used to describe the motor controlled by the Drive.

Index	Name	Type	Map	Attributes
2000h	**Nominal speed**	UINT16	No	rw
2001h	**Nominal current**	UINT16	No	rw
2002h	**Peak current**	UINT16	No	rw
2003h	**Motor poles**	UINT16	No	rw
2030h	**Type motor**	UINT16	No	rw
2033h	**Stall current**	UINT16	No	rw
2034h	**Nominal voltage**	UINT16	No	rw
2035h	**Phase resistor**	UINT16	No	rw
2036h	**Synchrony Inductance**	UINT16	No	rw
2037h	**I2t Time**	UINT16	No	rw
2043h	**Pole pitch**	UINT32	No	rw
204Dh	**Magnet flow**	UINT16	No	rw
209Ah	**Nominal frequency**	UINT16	No	rw
209Bh	**Nominal power factor**	UINT16	No	rw
209Dh	**Frequency jumps 1..4**	Array[UINT16]	No	rw

Index	Name	Type	Map	Attributes
209Eh	Delta freq. 1..4	Array[UINT16]	No	rw
209Fh	Phase resistor (FOC)	UINT32	No	rw
20A0h	Magnetizing current (FOC)	UINT32	No	rw
20A1h	Magnetizing inductance (FOC)	UINT32	No	rw
20A2h	Leakage inductance (FOC)	UINT32	No	rw
20A7h	Freq. 1..5	Array[UINT16]	No	rw
20A8h	Volt. 1..5	Array[UINT16]	No	rw
20ADh	Current limit Kp	UINT16	No	rw
20AEh	Current limit Ki	UINT16	No	rw
20B0h	flag (FOC)	UINT16	No	rw
20B1h	Rotor time costant (FOC)	UINT32	No	rw
20B4h	G mix (FOC)	UINT16	No	rw
20C1h	Kp vel. (FOC)	UINT16	No	rw
20C2h	Ki vel. (FOC)	UINT16	No	rw
20C3h	Current covariance (FOC)	UINT16	No	rw
20C4h	Flux covariance (FOC)	UINT16	No	rw
20C5h	Speed covariance (FOC)	UINT16	No	rw
20C6h	Current noise (FOC)	UINT16	No	rw
20C7h	Gmix (FOC-sensorless)	UINT16	No	rw

These parameters can be set in the Caliper menu "***Motor data***" in which the following screen appears:



5.02.1 Object 2000h – Nominal speed

Index	Name	Type	Range	Default	Unit	Map	Attributes
2000h	Nominal speed	UINT16	0 ... 65000	3000	Rpm o mm/s	No	rw

This object contains the nominal speed of the motor expressed in "***rpm***" for the rotative motors or in "***mm/s***" for the linear motors.

5.02.2 Object 2001h – Nominal current

Index	Name	Type	Range	Default	Unit	Map	Attributes
2001h	Nominal current	UINT16	0...60000	100	A/100	No	rw

This object contains the nominal current of the motor expressed in hundredths of Ampere. The least significant decimal places are intended as decimal fractions of Ampere; for example, the value 100 is read as 1,00 [A].

5.02.3 Object 2002h – Peak current

Index	Name	Type	Range	Default	Unit	Map	Attributes
2002h	Peak current	UINT16	0...60000	200	A/100	No	rw

This parameter contains the peak current of the motor expressed in hundredths of Ampere. The least significant decimal places are intended as decimal fractions of Ampere; for example, the value 200 is read as 2,00 [A].

5.02.4 Object 2003h – Motor poles

Index	Name	Type	Range	Default	Unit	Map	Attributes
2003h	Motor poles	UINT16	2...50	6		No	rw

This parameter is used to set the number of poles of the motor. If the controlled motor has an encoder with a number of pulses per turn that is set correctly in the Drive, then the Drive is able to automatically count the number of poles of the motor during the phasing procedure.

5.02.5 Object 2030h – Motor type

Index	Name	Type	Range	Default	Unit	Map	Attributes
2030h	Motor type	UINT16	0...4	0		No	rw

This object is used to select what kind of motor is used

Value	Motor types	Description
0	PM Rotary motor	This type is used for the SPM brushless motors. When this type is selected, the values reported in the menu "***Data monitor***" of the Caliper will adopt "***rpm***" as unit of measure of the speed and "***increments***" as unit of measure of the position. In addition, the parameters used by the "***Conversion Factors***" menu (see *"4.05 Conversion factors" pag. 60*) are set up for a rotary motor.
1	PM Linear motor	When this type is selected, the values reported in the menu "***Data monitor***" of the Caliper will adopt "***mm/s***" as unit of measure of the speed and "***mm***" as unit of measure of the position. In addition, the parameters used by the "***Conversion Factors***" menu (see*"4.05 Conversion factors" pag. 60*) are set up for a linear motor. In the Motor data menu the parameter "***pole pitch***" (***index 2043h***) is included and the number of poles of the motor is set to 2.
2	Brushed DC motor	When this type is selected, the brushed DC motor control is enabled; the unit of measures of speed and position are the same used with the rotary SPM motor. In this mode the positive terminal of the motor must be connected with the ***U*** phase and the negative terminal must be connected with the ***W*** phase of the Drive.

Value	Motor types	Description
3	Induction V-Hz	This type is used for the asynchronous motors. The Drive controls directly the output voltage in order to create the rotating magnetic field necessary to bring the motor to the reference speed. There are no current nor speed feedback; the torque is not controlled
4	Induction Foc	This type is used for the asynchronous motors. Using a model of the motor and the current and speed feedbacks, the ***Field Oriented Control*** (***FOC***) allows you to control the torque of the induction motor, thus achieving better performance than the V-Hz control.

5.02.6 Object 2033h – Stall current

Index	Name	Type	Range	Default	Unit	Map	Attributes
2033h	Stall current	UINT16	0...60000	100	A/100	No	rw

This object contains the value of the current that the motor can continuously absorb when the rotor is blocked, expressed in hundredths of Ampere. The least significant decimal places are intended as decimal fractions of Ampere; for example, the value 100 is read as 1,00 [A].

5.02.7 Object 2034h – Nominal voltage

Index	Name	Type	Range	Default	Unit	Map	Attributes
2034h	Nominal voltage	UINT16	10...500	230	V	No	rw

This object contains the value of the nominal motor voltage, expressed in [Volt].

5.02.8 Object 2035h – Phase resistance

Index	Name	Type	Range	Default	Unit	Map	Attributes
2035h	phase resistance	UINT16	1...65000	20	Ohm/100	No	rw

This object contains the resistance of a single phase of the motor, expressed in hundredths of [Ohm].

5.02.9 Object 2036h – Synchronous inductance

Index	Name	Type	Range	Default	Unit	Map	Attributes
2036h	Synchronous inductance	UINT16	1...65000	2	mH/ 100	No	rw

This object contains the synchronous inductance of a single phase of the motor, expressed in hundredths of [mH].

5.02.10 Object 2037h – I2T time

Index	Name	Type	Range	Default	Unit	Map	Attributes
2037h	I2T time	UINT16	1...3000	120	sec	No	rw

This object is used to set the maximum time interval, expressed in seconds, in which the motor could absorb twice the current set on "***Nominal current***" (***index 2001h***). If this current is absorbed for a time interval longer than the one set on this parameter,the Drive goes into alarm and the current will be automatically limited to its nominal value.

5.02.11 Object 2043h – Pole pitch

Index	Nome	Tipo	Range	Default	Unit	Map	Attributes
2043h	Pole pitch	UINT32	1 ... 1000000	10000	mm/ 1000	No	rw

Su questo parametro si imposta il passo polare del motore, misurato in millesimi di millimetro. Questo parametro appare solamente quando si sceglie un motore lineare (oggetto 2030h portato a [1])

5.02.12 Object 204Dh – Magnetic flux

Index	Name	Type	Range	Default	Unit	Map	Attributes
2043h	Magnetic flux	UINT16	1...65000	80	Wb/ 1000	No	rw

This object is used only when the Drive is controlling a brushless motor (rotative or linear). It contains the magnetic flux linked with a motor phase. It's expressed in hundredths of [Weber].

5.02.13 Object 209Ah – Nominal frequency

Index	Name	Type	Range	Default	Unit	Map	Attributes
209Ah	Nominal frequency	UINT16	100...65000	530	Hz/ 10	No	rw

This object is used only when the Drive is controlling an induction motor. It contains the nominal frequency of the motor. It's expressed in tenths of [Hertz].

5.02.14 Object 209Bh – Nominal power factor

Index	Name	Type	Range	Default	Unit	Map	Attributes
209Bh	Nominal power factor	UINT16	500... 1000	850	1/ 1000	No	rw

This object is used only when the Drive is controlling an induction motor. It contains the nominal power factor - cos(phi) - of the motor. It's expressed in tenths of thousandths.

5.02.15 Obect 209Dh – Frequency jumps

Index	Sub-index	Name	Type	Range	Default	Unit	Map	Attributes
209Dh	**1**	**Freq. 1**	UINT16	100... 3000	100	Hz/ 10	No	rw
	2	**Freq. 2**	UINT16	100... 3000	200	Hz/ 10	No	rw
	3	**Freq. 3**	UINT16	100... 3000	400	Hz/ 10	No	rw
	4	**Freq. 4**	UINT16	100... 3000	600	Hz/ 10	No	rw

This object is used only when the Drive is controlling an induction motor with the V-Hz control. It's used to set 4 frequency values that the Drive must avoid, because they would create dangerous resonances on the motor. When the Drive is changing the output voltage frequency it will "skip" these values, keeping the frequency to at least a minimum difference from them which could be set on the parameter "***frequency delta***" (***index 209Eh***).

5.02.16 Object 209Eh – Frequency delta

Index	Sub-index	Name	Type	Range	Default	Unit	Map	Attributes
209Eh	**1**	**D.freq. 1**	UINT16	0... 100	0	Hz/ 10	No	rw
	2	**D.freq. 2**	UINT16	0... 100	0	Hz/ 10	No	rw
	3	**D.freq. 3**	UINT16	0... 100	0	Hz/ 10	No	rw
	4	**D.freq. 4**	UINT16	0... 100	0	Hz/ 10	No	rw

This object is used only when the Drive is controlling an induction motor with the V-Hz control. It's used to set 4 frequency values,named "***frequency delta***". there is one frequency delta assigned for every resonance frequencies set on index 209Dh. When the Drive has to skip one of the chosen resonance frequencies, it will bring the voltage frequency to a value with a minimum difference from the resonance frequency equals to its frequency delta

5.02.17 Objects 20A7h and 20A8h - V/Hz ramp

Index	Sub-index	Name	Type	Range	Default	Unit	Map	Attributes
20A7h	**1**	**Freq. 1**	UINT16	0...249	0	%/10	No	rw
	2	**Freq. 2**	UINT16	1...499	250	%/10	No	rw
	3	**Freq. 3**	UINT16	251...749	500	%/10	No	rw
	4	**Freq. 4**	UINT16	501...999	750	%/10	No	rw
	5	**Freq. 5**	UINT16	751...1000	1000	%/10	No	rw
20A8h	**1**	**Volt. 1**	UINT16	0...1000	0	%/10	No	rw
	2	**Volt. 2**	UINT16	0...1000	250	%/10	No	rw
	3	**Volt. 3**	UINT16	0...1000	500	%/10	No	rw
	4	**Volt. 4**	UINT16	0...1000	750	%/10	No	rw
	5	**Volt. 5**	UINT16	0...1000	1000	%/10	No	rw

These objects are used only when the Drive is controlling an induction motor via the "***V-Hz***" control. They constitute the x and y coordinates of 5 points, (Freq.1, Volt.1) ... (Freq.5, Volt.5). Interpolating these points the Drive calculates a function that reduces the voltage supplied to the motor with the decrease of the frequency. With the default values, the Drive will reduce the voltage proportionally to the frequency, to keep the rotor flux costant when the motor is rotating below the nominal speed.

5.02.18 Object 20ADh - Current limit Kp

Index	Name	Type	Range	Default	Unit	Map	Attributes
20ADh	Current limit Kp	UINT16	0...65535	1200		No	rw

This object is used only when the Drive is controlling an induction motor with the "***V-Hz***" control. It's the proportional gain value of a PID regulator, used to bring the current to its nominal value.

5.02.19 Object 20AEh - Current limit Ki

Index	Name	Type	Range	Default	Unit	Map	Attributes
20AEh	Current limit Ki	UINT16	0...65535	100		No	rw

This object is used only when the Drive is controlling an induction motor with the "***V-Hz***" control. It's the integral gain value of a PID regulator, used to bring the current to its nominal value.

PRELIMINARY

5.02.20 Field Oriented Control (FOC)

The "***Field Oriented Control***" (***FOC***) is used to command the induction motors. It's a closed loop regulator, i.e. it uses the current and speed feedbacks (when a sensor is installed) to achieve better performance than "***V-Hz***" control. The key of the ***FOC*** functioning is the estimation of the rotor flux intensity and direction. If the correct direction of the magnetic flux is known the motor torque can be controlled more effectively by adjusting the current in quadrature with the flux. The magnetic flux cannot be directly measured though, so it needs to be estimated with an ***Observer***. Our Drives support three differents flux observer you can choose from:

- "***Voltage/Current***" Observer;
- "***Luenberger***" Observer;
- "***Indirect***" Observer*.

To improve the estimation of these Observers it's necessary to set all the parameters of the induction motors; if their values are not avaiable it's possible to measure them on the spot using an autotuning procedure.

If you want to control an induction motor in sensorless you have to use another kind of observers, which can also estimate the motor speed. Our Drives support three different flux and speed observers that you can choose from:

- "***Luenberger***" Observer;
- "***Kalman***" Observer;
- "***Voltage/Current***" Observer.

** Note: actually the "**Indirect**" Observer does not estimate the magnetic flux of the rotor, bu is able to indirectly bring the torque current into quadrature with it.*

PRELIMINARY

5.02.20.a Object 209Fh – Phase resistance (FOC)

Index	Name	Type	Range	Default	Unit	Map	Attributes
209Fh	Phase resistance (FOC)	UINT32	1... 10000000	2000	Ohm/10000	No	rw

This parameter is used only when the Drive is controlling an induction motor with the FOC. It represents the phase resistance of the motor stator.

5.02.20.b Object 20A0h – Magnetizing current (FOC)

Index	Name	Type	Range	Default	Unit	Map	Attributes
20A0h	Magnetizing current (FOC)	UINT32	0... 60000	200	A/100	No	rw

This parameter is used only when the Drive is controlling an induction motor with the FOC. It represents the magnetizing current required to create the nominal magnetic flux.

5.02.20.c Object 20A1h – Magnetizing inductance (FOC)

Index	Name	Type	Range	Default	Unit	Map	Attributes
20A1h	Magnetizing inductance (FOC)	UINT32	1... 10000000	30000	H/100000	No	rw

This parameter is used only when the Drive is controlling an induction motor with the FOC. It represents the mutual inductance between the stator and rotor windings.

5.02.20.d Object 20A2h – Leakage inductance (FOC)

Index	Name	Type	Range	Default	Unit	Map	Attributes
20A2h	Leakage inductance	UINT32	1... 10000000	350000	mH/10000	No	rw

This parameter is used only when the Drive is controlling an induction motor with the FOC. It represents the total leakage inductance of the motor, which takes into account both the stator and rotor windings.

5.02.20.e Object 20B0h – Flag (FOC)

Index	Name	Type	Range	Default	Unit	Map	Attributes
20B0h	Flag (FOC)	UINT16	0...65536	1		No	rw

This object contains several flags used only when the Drive is controlling an induction motor with the FOC. In the following table is described their tasks.

PRELIMINARY

Flag FOC (index 20B0h)	
Bit	***description***
0	• Bit 0=0: In addition to the flux weakening PID, a PID for the flux control is used • Bit 0=1: Only the flux weakening PID is used
1	• Bit 1=0: The Observers use the parameters calculated by the motor plate • Bit 1=1: The Observers use the parameters calculated by the autotuning procedure
2,3	• Bit 2,3=0: The "***Voltage/current***" Observer is used for the flux estimation (a speed sensor is installed) • Bit 2,3=1: The "***Luenberger***" Observer is used for the flux estimation (a speed sensor is installed) • Bit 2,3=2: The "***Indirect***" Observer is used for the FOC functioning (a speed sensor is installed)
4	• Bit 4=0: the Magnetizing current is set manually • Bit 4=1: the magnetizing current is obtained by the "no load test" of the autotuning procedure
5	This bit select what value to assign to the magnetizing inductance, a costant value or a value taken from a curve measured by the autotuning procedure, which depends on the current.
6,7	• Bit 6,7=0: The "***Luenberger***" Observer is used for the flux and speed estimation (with the sensorless mode) • Bit 6,7=1: The "***Kalman filter***" Observer is used for the flux and speed estimation (with the sensorless mode) • Bit 6,7=2: The "***Voltage/current***" Observer is used for the flux and speed estimation (with the sensorless mode)
8,15	Not used at the moment

5.02.20.f Object 20B1h – Rotor time costant (FOC)

Index	Name	Type	Range	Default	Unit	Map	Attributes
20B1h	Rotor time costant	UINT32	1... 10000000	70000	ms/1000	No	rw

This parameter is used only when the Drive is controlling an induction motor with the FOC. it represents the time costant of the equivalent rotor winding (rotor inductance/rotor resistance).

5.02.20.g Object 20B4h – Gmix (FOC)

Index	Name	Type	Range	Default	Unit	Map	Attributes
20B4h	G mix	UINT16	1... 65535	10000		No	rw

"***Gmix***" is a parameter used by the "***Voltage/Current***" observer. This Observer in turn is made up of two different observers:

- the first one estimates the magnetic flux using both the current and the voltage of the stator winding;
- the second one estimates the magnetic flux using only the stator current and the rotor cage parameters.

The first Observer is more accurate, but it has a problem: The flux is obtained by the integral of the voltage, and this integral can diverge if there is an offset at its input. To solve this problem the Drive makes a "weighted average" of the fluxes obtained by the two obeserver at each iteration; with this solution the flus estimated by the first observer is corrected step by step, preventing the risk of divergence. The parameter "***Gmix***" is used to decide how much the flux of second observer affects this average:

- an higher ***Gmix*** means that the averaged flux is more dependent on the estimation of the second observer, with the risk of obtaining a less accurate value;
- a lower ***Gmix*** means that the averaged flux is more dependent on the estimation of the first observer, with the risk of a diverging value.

5.02.20.h Object 20C1h – Kp vel. (FOC)

Index	Name	Type	Range	Default	Unit	Map	Attributes
20C1h	Kp vel.	UINT16	1... 65535	2000		No	rw

This object sets the proportional gain of a PID, used by the "***Luenberger***" observer used to calculate the motor speed when it's working in sensorless. This PID takes into account the estimated fluxes of the observer and the difference between the estimated currents and the measured ones, and calculates the speed that should cancel this error.

5.02.20.i Object 20C2h – Ki vel. (FOC)

Index	Name	Type	Range	Default	Unit	Map	Attributes
20C2h	Ki vel.	UINT16	0... 65535	100		No	rw

This object sets the integral gain of a PID, used by the "***Luenberger***" observer used to calculate the motor speed when it's working in sensorless. This PID takes into account the estimated fluxes of the observer and the difference between the estimated currents and the measured ones, and calculates the speed that should cancel this error.

5.02.20.j Object 20C3h – Current covariance (FOC)

Index	Name	Type	Range	Default	Unit	Map	Attributes
20C3h	Current covariance (FOC)	UINT16	0,01... 650	10		No	rw

The "***current covariance***" represents the error on the current estimate that we assume will occur due to an inaccurate model of the induction motor (es. incorrect value of resistance and inductance). An high covariance means that a not very precise current's estimate is expected, a low covariance means that an accurate current's estimate is expected. The "***Kalman filter***" takes into account these covariances in order to make the best possible estimation of the observed values.

5.02.20.k Object 20C4h – Flux covariance (FOC)

Index	Name	Type	Range	Default	Unit	Map	Attributes
20C4h	Flux covariance (FOC)	UINT16	0,01... 650	1		No	rw

The "***flux covariance***" represents the error on the flux estimate that we assume will occur due to an inaccurate model of the induction motor (es. incorrect value of resistance and inductance). An high covariance means that a not very precise flux estimate is expected, a low covariance means that an accurate flux estimate is expected. The "***Kalman filter***" takes into account these covariances in order to make the best possible estimation of the observed values.

5.02.20.l Object 20C5h – Speed covariance (FOC)

Index	Name	Type	Range	Default	Unit	Map	Attributes
20C5h	Speed covariance (FOC)	UINT16	0,01... 650	10		No	rw

The "***speed covariance***" represents the error on the speed estimate that we assume will occur due to an inaccurate model of the induction motor (es. incorrect value of resistance and inductance). An high covariance means that a not very precise speed estimate is expected, a low covariance means that an accurate speed estimate is expected. The "***Kalman filter***" takes into account these covariances in order to make the best possible estimation of the observed values.

5.02.20.m Object 20C6h – Current noise (FOC)

Index	Name	Type	Range	Default	Unit	Map	Attributes
20C6h	Current noise (FOC)	UINT16	0,01... 650	50		No	rw

The "***current noise***" represents the error on the current measurement that is assumed to occur due to the limitations of the current sensors. A high noise means that a not very precise current measurement is expected, a low noise means that an accurate current' measurement is expected. The "***Kalman filter***" takes into account the covariances of the observed values and the current noise in order to make the best possible estimates.

5.02.20.n Object 20C7h – Gmix-sensorless (FOC)

Index	Name	Type	Range	Default	Unit	Map	Attributes
20C7h	Gmix-sensorless	UINT16	1... 65535	100		No	rw

The parameter "***Gmix-sensorless***" is used by a "***Voltage/Current***" observer similar to the one described in the paragraph *"5.02.20.g Object 20B4h – Gmix (FOC)" pag. 135*. This Observer in turn is made up of two different observers:

- the first one estimates the magnetic flux using both the current and the voltage of the stator winding;
- the second one estimates the magnetic flux using only the stator current and the rotor cage parameters.

The first Observer is more accurate, but it has a problem: The flux is obtained by the integral of the voltage, and this integral can diverge if there is an offset at its input. To solve this problem the Drive makes a "weighted average" of the fluxes obtained by the two obeserver at each iteration; with this solution the flus estimated by the first observer is corrected step by step, preventing the risk of divergence. The parameter "***Gmix-sensorless***" is used to decide how much the flux of second observer affects this average:

- an higher ***Gmix*** means that the averaged flux is more dependent on the estimation of the second observer, with the risk of obtaining a less accurate value;

a lower ***Gmix*** means that the averaged flux is more dependent on the estimation of the first observer, with the risk of a diverging value.

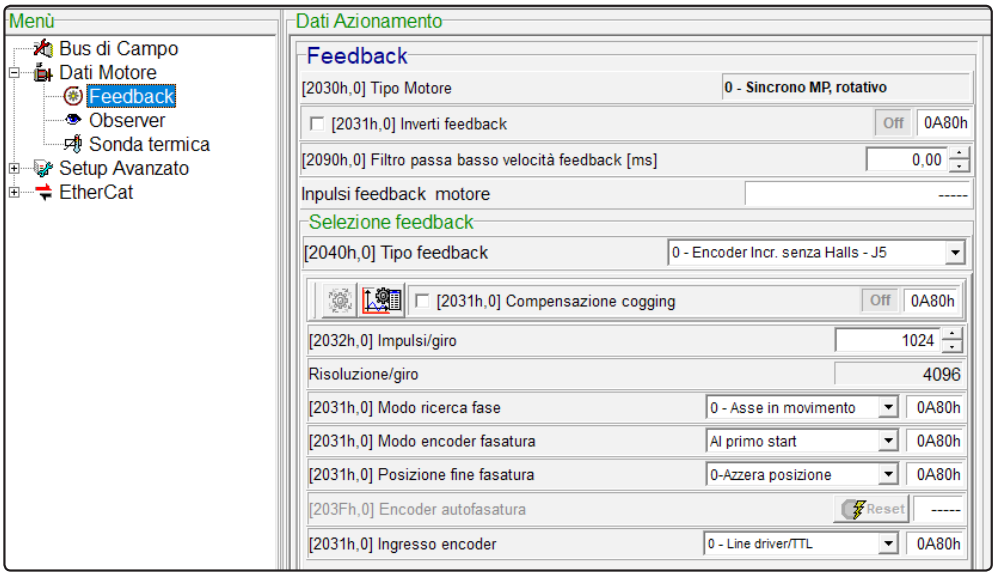
This observer, unlike the one described above, also estimates the rotor speed, by subtracting the speed of the rotating magnetic flux with the estimated slip.

5.03 Feedback parameters

This paragraph lists the parameters related to the position sensor of the motor.

Index	Name	Type	Map	Attributes
2004h	Feedback Offset	UINT16	No	rw
2031h	Flags	UINT16	No	rw
2032h	Pulses/revolution	UINT32	No	rw
203Fh	Encoder autophasing	UINT16	No	rw
2040h	Feedback type	UINT16	No	rw
2041h	Bit single-turn	UINT16	No	rw
2042h	Bit multi-turn	UINT16	No	rw
2044h	Application Offset	INT32	No	rw
2045h	Encoder position	INT32	No	ro
204Eh	Resolver phase	INT16	Yes	ro
2060h	Current covariance	UINT16	No	rw
2061h	Speed covariance	UINT16	No	rw
2062h	Phase covariance	UINT16	No	rw
2063h	Current noise	UINT16	No	rw
2064h	Zero crossing	UINT16	No	rw
2065h	Starting current	UINT16	No	rw
2066h	Current ramp	UINT16	No	rw
2067h	Stationing current	UINT16	No	rw
2068h	Initial speed max.	UINT16	No	rw
2069h	Speed hysteresis	UINT16	No	rw
206Ah	Speed ramp	UINT16	No	rw
206Bh	Fault measured speed	UINT16	No	rw
206Ch	Initial delay	UINT16	No	rw
207Dh	Resolver state	INT16	No	ro
207Eh	Resolver poles	UINT16	No	rw
20C0h	Serial frequency encoder	UINT32	No	ro
20CDh	Application Offset (MSB)	INT64	No	rw
20CEh	Application Offset (LSB)			
20CFh	Encoder position (MSB)	INT64	No	ro
20D0h	Encoder position (LSB)			

These parameters can be set in the Caliper menu "***Motor data-Feedback***" in which the following screen appears:



5.03.1 Object 2004h – Feedback Offset

Index	Name	Type	Range	Default	Unit	Map	Attributes
2004h	Feedback Offset	UINT16	0...36000	0	°/100	No	rw

This object contains the offset of the position sensor, expressed in degrees[°]. The least significant digits are interpreted as decimal fractions of a degree, for example the value 18000 is intended as 180,00°. The position sensor offset can be automatically calculated by pushing the button "***sensor pos autophasing*** ", (it's in the toolbar above the screen), which starts the phasing procedure of the position sensor and counts the number of poles.

5.03.2 Object 2031h - Flags

Index	Name	Type	Range	Default	Unit	Map	Attributes
2031h	Flags	UINT16	0..65536		Bit	No	rw

This object contains various flags, most of which are used to set the motor feedback. These flags are listed in the table below:

Bit	Name	2031h- Bit description
0	***Connection type***	0: star
		1: triangle
1	***Reverse Feedback***	0: normal feedback
		1: reversed feedback
2	***Cogging compensation***	0: no cogging compensation
		1: cogging compensation
3	***Encoder input***	0: Line driver/TTL
		1: Open collector/Push Pull
4,5	***Encoder phasing mode (without Hall)***	00: At the first start 01: At every start 10: After each reset
6,7	***Resolver precision***	00: 10 Bit 01: 12 Bit 10: 14 Bit 11: 16 Bit

Bit	Name	2031h- Bit description
8,9	***Encoder precision***	00: 256 01: 1024 10: 4096 11: 16384
10	***Reverse output encoder***	0: normal simulated encoder
		1: reversed simulated encoder
11	***Check electrical angle***	0: Off
		1: On
12	***mode phase search (without Hall)***	0: Axis in motion
		1: Standstill axis
13...15	***reserved***	

- The parameter "***Resolver precision***" is used to adjust the resolver resolution from 10 bits up to 16. The default value is 14 bit, which works fine for most applications. For application where the nominal speed could reach 5000 [rpm] is possible to set the resolution to 16 bit to improve the performance of the position control, at the expense of the bandwidth of the speed loop, which is 4 times smaller than the bandwidth with a 14 bit resolution.
- The parameter "***encoder precision***" allows you to set the number of pulses per (mechanical) rotation (ppr) produced by the output of the simulated encoder with index pulse. It's possible to set a number of pulses per rotation less than or equal to the number chosen for the feedback.

5.03.3 Object 2032h - Pulses/revolution

Index	Name	Type	Range	Default	Unit	Map	Attributes
2032h	Impulsi per giro	UINT32	100 ... 16777216	1024 (dec)		No	rw

This object is used to set the number of pulses produced by the encoder when the motor complete one revolution or, in case of a linear motor, when it covers a length equal to the pole pitch.

5.03.4 Object 203Fh – Encoder autophasing

Index	Name	Type	Range	Default	Unit	Map	Attributes
203Fh	Encoder Autophasing	UINT16	0...1	0		No	rw

This object is used only when the Feedback type is set on "***Incremental Encoder without Halls***". At the end of the phasing procedure this object is set to 1. Bringing it back to 0 the previuous phasing of the motor will be resetted; the Drive will excecute another phasing procedure at the next start command.

PRELIMINARY

5.03.5 Object 2040h – Feedback type

Index	Name	Type	Range	Default	Unit	Map	Attributes
2040h	Feedback type	UINT16	0...12	0		No	rw

This object is used to set what type of position sensor is used as feedback by the motor.

Feedback type
0 - Incremental encoder without Halls
Using this type you have to set the number of pulses of the encoder on the parameter "***Pulses per revolution***" (***index 2032h***). At the first start command the phasing procedure will be executed for few seconds; this procedure is necessary to align the rotor magnet in the correct position in order to obtain the best torque control. This procedure involves a rotation of a certain angle (max 180 °/ n° of poles pair) or a shift if the motor is linear, so there must be no obstacles limiting the motor movement because they would cause an error in the calculation of the optimal position for the motor control. At the end of the procedure the parameter "***Encoder autophasing***" (***index 203Fh***) is set to 1; to repeat the procedure at the next power on just reset this last parameter.
1 - Incremental encoder with Halls
Using this type you have to set the number of pulses of the encoder on the parameter "***Pulses per revolution***" (***index 2032h***). The rotor magnet phase must be set in degrees [°] in the parameter "***Offset***" (***index 2004h***) or it could be automatically calculated by pushing the button "***sensor pos autophasing***  ", which starts the phasing procedure of the position sensor, counts the number of poles and calculates the offset of the magnetic field of the rotor.
2 - Halls only
This type means that the position feedback consists of hall sensors only. In this situation the performance of the control are very limited because without the encoder the motor must be controlled with a Trapezoidal current instead of a Sinusoidal one.
3 - Encoder SSI
With this type of sensor the position is acquired in a serial mode. You have to set on the Drive the number of bits of the position resolution provided by the sensor manufacturer, which have to be set in the parameters "***Bit single turn***" (***index 2041h***), "***Bit multi turn***" (***index 2042h***). The rotor magnet phase could be manually set, in degrees [°], on parameter "***Offset***" (***index 2004h***) or it could be automatically calculated by pushing the button "***sensor pos. autophasing***  ". The absolute position measured by the encoder is reported on the parameter "***Encoder position***" (***index 2045h***); this value is reduced by the value of "***Application offset***" (***index 2044h***), and the obtained result represents the value of the measured position used as feedback for the positioner applications. by acting on the "***Application offset***" is possible to move the position 0 of the application.
4 - Sensorless
This type is used when the motor has no position sensor. The Drive will use the ***Kalman Filter*** in order to estimate the rotor position. The functioning of this Filter and the parameters required to set it will be explained apart in paragraph *"5.03.16 Feedback parameters - Kalman filter" pag. 143*
5 - Resolver
With this type you have to set the "***Resolver precision***" (***index 2031h***). The rotor magnet phase must be set in degrees [°] in the parameter "***Offset***" (***index 2004h***) or it could be automatically calculated by pushing the button "***sensor pos. autophasing***  ", which starts the phasing procedure of the position sensor, counts the number of poles and calculates the offset of the magnetic field of the rotor.
6- Encoder Biss
This feedback is an absolute encoder which comunicate using the ***Biss*** protocol. There is no need to set the parameters "***Bit single turn***" and "***Bit multi turn***" , because they're automatically read when the Drive turn on. The rotor magnet phase could be manually set, in degrees [°], on parameter "***Offset***" (***index 2004h***) or it could be automatically calculated by pushing the button "***sensor pos autophasing***  ".
7-Encoder Endat
This feedback is an absolute encoder which comunicate using the ***Endat*** protocol. There is no need to set the parameters "***Bit single turn***" and "***Bit multi turn***" , because they're automatically read when the Drive turn on. The rotor magnet phase could be manually set, in degrees [°], on parameter "***Offset***" (***index 2004h***) or it could be automatically calculated by pushing the button "***sensor pos autophasing***  ".

Feedback type
8-Encoder Biss Sin/Cos
This feedback converts the measured position in two analog sinusoidal signals, the "***Sin/Cos***", instead of converting it in two digital signals. This change increase the position resolution. Using only these two analogical signals you can't obtain the absolute position. To get the latter the Drive will read the absolute position from the encoder ***Biss*** when it turns on, and next it will start using the "***Sin/Cos***" : summing the relative position read by the "***Sin/Cos***" with the absolute position initially read by the encoder ***Biss*** the Drive obtains the current absolute position of the motor
9-Encoder Endat Sin/Cos
This feedback converts the measured position in two analog sinusoidal signals, the "***Sin/Cos***", instead of converting it in two digital signals. This change increase the position resolution. Using only these two analogical signals you can't obtain the absolute position. To get the latter the Drive will read the absolute position from the encoder ***Endat*** when it turns on, and next it will start using the "***Sin/Cos***" : summing the relative position read by the "***Sin/Cos***" with the absolute position initially read by the encoder ***Endat*** the Drive obtains the current absolute position of the motor
10-Incr. Encoder Sin/Cos
This feedback converts the measured position in two analog sinusoidal signals, the "***Sin/Cos***", instead of converting it in two digital signals. This change increase the position resolution. Using only these two analogical signals you can't obtain the absolute position. Therefore the Drive has to execute the phasing procedure every times it turns on, as the one described when we use an incremental encoder without Hall as feedback.
11-Encoder SSI Sin/cos
This feedback converts the measured position in two analog sinusoidal signals, the "***Sin/Cos***", instead of converting it in two digital signals. This change increase the position resolution. Using only these two analogical signals you can't obtain the absolute position. To get the latter the Drive will read the absolute position from the encoder ***SSI*** when it turns on, and next it will start using the "***Sin/Cos***" : summing the relative position read by the "***Sin/Cos***" with the absolute position initially read by the encoder **SSI** the Drive obtains the current absolute position of the motor
12-Encoder Renishaw
This Feedback is a ***Biss unidirectional*** absolute encoder. This means that the parameters "***Bit single turn***" and "***Bit multi turn***" are not automatically read, so they must be set manually. The rotor magnet phase could be manually set, in degrees [°], on parameter "***Offset***" (***index 2004h***) or it could be automatically calculated by pushing the button "***sensor pos autophasing***  ".

PRELIMINARY

5.03.6 Object 2041h – Bit single-turn

Index	Name	Type	Range	Default	Unit	Map	Attributes
2041h	Bit single turn	UINT16	4...31	12	Count	No	rw

This object contains the number of bits of resolution of the absolute encoder for a single turn of the motor,i.e. a single turn is splitted in a number of sectors equal to the value set on this parameter, the obtained sectors are the minimum "variation" that the encoder can detect .

5.03.7 Object 2042h – Bit Multi-turn

Index	Name	Type	Range	Default	Unit	Map	Attributes
2042h	Bit multi-turn	UINT16	0...31	0	Count	No	rw

This object contains the number of bits of resolution of the absolute encoder for many turns of the motor,i.e. it sets the maximum value of motor turns that the Drive will be able to count.

5.03.8 Object 2044h – Application offset

Index	Name	Type	Range	Default	Unit	Map	Attributes
2044h	Application offset	INT32	± 7FFFFFFFh	0		No	rw

When an absolute encoder is used, by using the "***application offset***" parameter it is possible to shift the "position 0" of the application. When you're working with a position control, in fact, the value of the measured position is obtained by subtracting the value of the "***Encoder position***" (***index 2045h***) with this offset:

$$Measured\_position = Encoder\_position - Application\_Offset$$

Pushing the button "***Home***" on the menù "***Feedback***", the actual position will be reset depending on the value read from the parameter "***Encoder position***" (***index 2045h***).

PRELIMINARY

5.03.9 Object 2045h – Encoder position

Index	Name	Type	Range	Default	Unit	Map	Attributes
2045h	Encoder position	INT32	± 7FFFFFFFh	0		No	ro

This read-only parameter shows the absolute value of the measured position read by the serial encoder SSI.

5.03.10 Object 204Eh– Resolver phase

Index	Name	Type	Range	Default	Unit	Map	Attributes
204Eh	Resolver phase	UINT16	0 ... 65535	0		Yes	ro

This object contains the current position angle (expressed in increments) of the resolver.

5.03.11 Object 207Eh – Resolver poles

Index	Name	Type	Range	Default	Unit	Map	Attributes
207Eh	Resolver poles	UINT16	2...16	2		No	rw

This object contains the number of pole couples of the resolver.

5.03.12 Object 207Dh – Resolver state

Index	Name	Type	Range	Default	Unit	Map	Attributes
207Dh	Resolver state	UINT16	0 ... 65535	0		No	ro

This read-only parameter shows the current state of the resolver.

Bit	Description
0	Configuration parity error
1	Phase error lock range
2	Velocity max tracking rate
3	Tracking error threshold
4	Sin/cos mismatch threshold
5	Sin/cos ovverrange threshold
6	Sin/cos below threshold
7	Sin/cos clipped

5.03.13 Object 20C0h - Serial frequency encoder

Index	Name	Type	Range	Default	Unit	Map	Attributes
20C0h	Serial frequency encoder	UINT32	0 .... FFFFFFFFh	0		No	ro

This parameter contains the frequency of the clock used by the encoder to communicate with the Drive.

5.03.14 Objects 20CDh, 20CEh – Application offset

Index	Name	Type	Range	Default	Unit	Map	Attributes
20CDh 20CEh	Application offset	INT64		0		No	rw

This parameter is used only with the Biss,Endat, SSi sin/cos or Renishaw encoder types. it's an offset used to shift the "position 0" of the application. When you're working with a position control, in fact, the value of the measured position is obtained by subtracting the value of the "***Encoder position***" (***index 20CFh e 20D0h***) with this value:

$$Measured\_position = Encoder\_position - Application\_Offset$$

Pushing the button "***Home***" on the menù "***Feedback***", the actual position will be reset depending on the value read from the parameter "***Encoder position***" (***index 20CFh e 20D0h***). This parameter differs from the object on index ***2044h*** because it has a resolution of 64 bits.

5.03.15 Objects 20CF,20D0h – Encoder position

Index	Name	Type	Range	Default	Unit	Map	Attributes
20CFh 20D0h	Encoder position	INT64		0		No	ro

This read-only parameter shows the absolute value of the measured position read by a Biss,Endat, SSi sin/cos or Renishaw encoder. It differs from the object on index ***2045h*** because it has a resolution of 64 bits.

5.03.16 Feedback parameters - Kalman filter

When the Drive has to control the motor in "***sensorless***" mode (there aren't any position feedbacks), in order to estimate the speed and the position it uses the ***Kalman filter***.

The Kalman filter is an ***Observer*** which use a model of the motor to estimate its state variables from its inputs. In our case the state variables are the currents, the speed and the position, while the inputs are the supplyed voltages. The estimation of this filter is corrected at each step using a sort of "feedback" based on the estimation errors between the measured currents and the estimated currents (you can use as feedback only the measurable state variables).

In order to make this filter works you need an accurate model of the motor, therefore you have to set correctly its parameters (resistance, inductance, magnetic flux, number of pole pairs).

The kalman filter resembles the Luenberger observer, but it has three major differences:

» The Kalman filter allows you to take into account the "***measurement errors***". In this application they refer only to the noise in the current measurement. On the object ***2063h*** you have to write the error that you expect to find when you're measuring the current.

» The Kalman filter allows you to take into account the "***model errors***", which inevitably appear when the real parameters of the motor differ form the ones set Via Caliper. These errors will affect the estimated values of currents, speed and position. On the objects ***2060h***, ***2061h***, ***2062h*** you have to write the model errors that you expect will appear estimating the state variables.

» The "gains" that multiplies with the "feedbacks" of the Kalman filter change over time, in order to make the best possible estimation. The expected model's and measurement's errors that you set via Caliper represent the reliability of our estimations and measurements. Depending on their value, the Kalman filter will change its gains to achieve the best possible estimation.

At very low speed the estimation made by the Kalman Filter is not very reliable. Therefore when the speed drops below a certain threshold the Drive will switch to a "***V-Hz***" control, i.e. it will directly produce the voltage necessary to bring (at regime) the motor to the selected reference speed, without estimating it's position.

5.03.16.a Object 2060h – Current covariance

Index	Name	Type	Range	Default	Unit	Map	Attributes
2060h	Current covariance	UINT16	1 ... 65000	1000	1/100	No	rw

This object is used by the Kalman filter, i.e. when the Drive works in sensorless mode (see *"5.03.16 Feedback parameters - Kalman filter" pag. 143*). The "***current covariance***" represents the current estimation error that is expected to appear due to of an approximate model of the motor (for example the set resistance and the inductance could be inaccurate). An high covariance means that you expect a rough estimation of the current, on the contrary a low covariance means that you expect an accurate estimation.

5.03.16.b Object 2061h – Speed covariance

Index	Name	Type	Range	Default	Unit	Map	Attributes
2061h	Speed covariance	UINT16	1 ... 65000	100	1/100	No	rw

This object is used by the Kalman filter, i.e. when the Drive works in sensorless mode (see *"5.03.16 Feedback parameters - Kalman filter" pag. 143*). The "***speed covariance***" represents the speed estimation error that is expected to appear due to an approximate model of the motor (for example the set resistance and the inductance could be inaccurate). An high covariance means that you expect a rough estimation of the speed, on the contrary a low covariance means that you expect an accurate estimation.

5.03.16.c Object 2062h – Covarianza fase (posizione)

Index	Name	Type	Range	Default	Unit	Map	Attributes
2062h	Covarianza di fase (posizione)	UINT16	1 ... 65000	80	1/10000	No	rw

This object is used by the Kalman filter, i.e. when the Drive works in sensorless mode (see *"5.03.16 Feedback parameters - Kalman filter" pag. 143*). The "***phase covariance***" represents the position estimation error that is expected to appear due to an approximate model of the motor (for example the set resistance and the inductance could be inaccurate). An high covariance means that you expect a rough estimation of the position, on the contrary a low covariance means that you expect an accurate estimation.

5.03.16.d Object 2063h – Current noise

Index	Name	Type	Range	Default	Unit	Map	Attributes
2063h	Current noise	UINT16	1 ... 65000	100	1/100	No	rw

This object is used by the Kalman filter, i.e. when the Drive works in sensorless mode (see *"5.03.16 Feedback parameters - Kalman filter" pag. 143*). The "***current noise***" represents the value of disturbance that is expected to affect your current measurement. An high covariance means that you expect an innacurate measurement of the current, on the contrary a low covariance means that you expect an accurate measurement.

5.03.16.e Object 2064h – Zero crossing

Index	Name	Type	Range	Default	Unit	Map	Attributes
2064h	Zero crossing	UINT16	0 ... 1	0		No	rw

This object is used by the Kalman filter, i.e. when the Drive works in sensorless mode (see *"5.03.16 Feedback parameters - Kalman filter" pag. 143*). It's used to set how the Drive will switch from the Kalman filter control to the "***V-Hz***" control.

Bit	description
[0] = Stop/Start	If the absolute value of the reference speed is lower than the difference between the "***initial speed max***" (***index 2068h***) and the "***speed hysteresis***" (***index 2069h***) then the Drive switches immediately to the ***V/Hz*** control.
[1] = Continuous	If the absolute value of the reference speed is lower than the difference between the "***initial speed max***" (***index 2068h***) and the "***speed hysteresis***" (***index 2069h***) then a countdown starts; when it terminates the Drive switches to the ***V/Hz*** control.

5.03.16.f Object 2065h– Starting current

Index	Name	Type	Range	Default	Unit	Map	Attributes
2065h	Starting current	UINT16	1 ... 2000	1000	%/10	No	rw

This object is used by the Kalman filter, i.e. when the Drive works in sensorless mode (see *"5.03.16 Feedback parameters - Kalman filter" pag. 143*). When a motor starts moving in sensorless mode, the Drive uses the "***V/Hz***" control instead the Kalman filter because the latter is not reliable at low speed. This parameter contains the value of the current (expressed in % with respect to the nominal motor current) that the "***V/Hz***" control will apply during the startup.

5.03.16.g Object 2066h – Current ramp

Index	Name	Type	Range	Default	Unit	Map	Attributes
2066h	Current ramp	UINT16	1 ... 60000	400	s/1000	No	rw

This object is used by the Kalman filter, i.e. when the Drive works in sensorless mode (see *"5.03.16 Feedback parameters - Kalman filter" pag. 143*). When a motor starts moving in sensorless mode, the current applied doesn't switch immediately to the value set on the object ***2065h***, but it rises with a linear ramp. The value of object ***2066h***, expressed in seconds, represents how long this ramp will last.

5.03.16.h Object 2067h – Stationing current

Index	Name	Type	Range	Default	Unit	Map	Attributes
2067h	Stationing current	UINT16	0 ... 2000	500	%/10	No	rw

This object is used by the Kalman filter, i.e. when the Drive works in sensorless mode (see *"5.03.16 Feedback parameters - Kalman filter" pag. 143*). It contains the current absorbed by the motor when the latter is standstill and the Drive is at ***SWITCH ON*** state. It's expressed as a percentage of the nominal current.

5.03.16.i Object 2068h – Initial speed max

Index	Name	Type	Range	Default	Unit	Map	Attributes
2068h	Initial speed max.	UINT16	1 ... 10000	100	rpm	No	rw

This object is used by the Kalman filter, i.e. when the Drive works in sensorless mode (see *"5.03.16 Feedback parameters - Kalman filter" pag. 143*). When a motor starts moving in sensorless mode, the Drive uses the "***V/Hz***" control instead of the Kalman filter because the latter is not reliable at low speed. This object contains the speed threshold (expressed in [rpm]) that the motor must reach before switching from the "***V/Hz***" control to the Kalman filter control.

5.03.16.j Object 2069h – Speed hysteresis

Index	Name	Type	Range	Default	Unit	Map	Attributes
2069h	Speed hysteresis	UINT16	1 ... 1000	20	rpm	No	rw

This object is used by the Kalman filter, i.e. when the Drive works in sensorless mode (see *"5.03.16 Feedback parameters - Kalman filter" pag. 143*). The Drive doesn't change immediately from the Kalman filter control to the ***V/Hz*** control when the motor speed drops below the threshold set on object ***2068h***, but at a lower speed. Therefore there is an hysteresis cycle when the Drive switches from one control type to the other. The object ***2069h*** represents indeed the width of this hysteresis cycle, i.e. the difference between "***Initial speed max.***" (***index 2068h***) and the reduced speed below which the Drive change to the ***V/Hz*** control.

5.03.16.k Object 206Ah– Speed ramp

Index	Name	Type	Range	Default	Unit	Map	Attributes
206Ah	Rampa velocità	UINT16	1...60000	1000	rpm/s	No	rw

This object is used by the Kalman filter, i.e. when the Drive works in sensorless mode (see *"5.03.16 Feedback parameters - Kalman filter" pag. 143*). When a motor starts moving in sensorless mode, the reference speed doesn't switch immediately to the value set on ***index 2068h***, but it rises with a linear ramp. The value of object ***206Ah,*** expressed in [rpm/s], represents the acceleration of this ramp.

5.03.16.l Object 206Bh– Fault measured speed

Index	Name	Type	Range	Default	Unit	Map	Attributes
206Bh	Fault velocità misurata	UINT16	1...65000	2000	rpm	No	rw

This object is used by the Kalman filter, i.e. when the Drive works in sensorless mode (see *"5.03.16 Feedback parameters - Kalman filter" pag. 143*). It represents the maximum allowed speed error when the Drive uses the Kalman Filter, above which the Drive switches directly to the ***V/Hz*** control.

5.03.16.m Object 206Ch – Initial delay

Index	Name	Type	Range	Default	Unit	Map	Attributes
206Ch	Ritardo iniziale	UINT16	1...10000	200	s/1000	No	rw

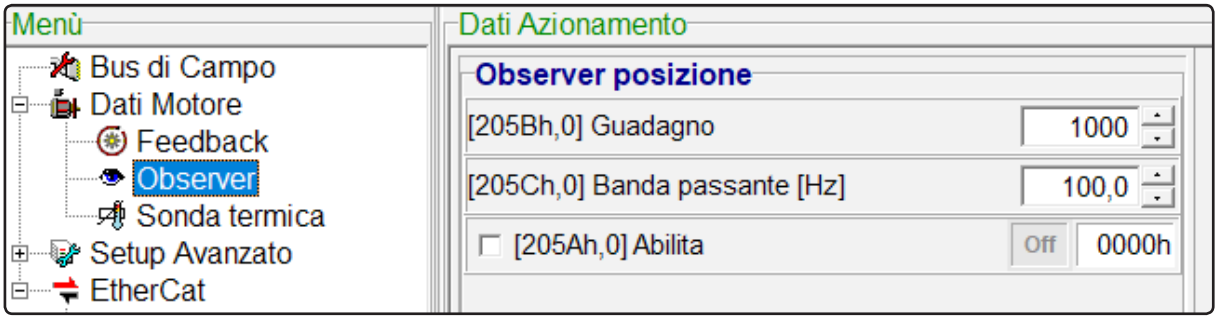
This object is used by the Kalman filter, i.e. when the Drive works in sensorless mode (see *"5.03.16 Feedback parameters - Kalman filter" pag. 143*). It represents the period of time the Drive has to wait before starting to follow the speed reference. This delay is necessary to permit to the motor to get in the initial orientation.

5.04 Observer parameters

In this section is described an Observer used by our Drives to improves the resolution of the position measured by the feedback.

Index	Name	Type	Map	Attributes
205Ah	**Enable Observer**	UINT16	No	rw
205Bh	**Gain**	UINT16	No	rw
205Ch	**Bandwidth**	UINT16	No	rw

The parameters related to the Observer can be set in the Caliper menu "***Motor data-Observer***" in which the following screen appears:



The Observer can be used with every kind of feedback, and it can increment the position resolution on a single turn up to 16 bit. This system could be useful to reduce the motor noise caused by the speed regulator in certain applications, while maintaining good static and dynamic performances.

We recommend its use when:

- the position feedback is provided by an encoder with a resolution below 1000 ppr or 12 bit;
- the position feedback is provided by hall sensors only

5.04.1 Object 205Ah – Enable Observer

Index	Name	Type	Range	Default	Unit	Map	Attributes
205Ah	Enable Observer	UINT16	0...65535	0		No	rw

This is a flag used to enable or disable the Observer.

5.04.2 Object 205Bh – Gain

Index	Name	Type	Range	Default	Unit	Map	Attributes
205Bh	Gain	UINT16	1...65535	1000		No	rw

This object contains the value of the Observer gain.

5.04.3 Object 205Ch – Bandwidth

Index	Name	Type	Range	Default	Unit	Map	Attributes
205Ch	Bandwidth	UINT16	1...60000	1000	Hz/ 10	No	rw

This object is used to set the Observer bandwidth, a sort of filter used to eliminate the truncation error from the measured position, which could be remarkable when a sensor with low resolution is used.

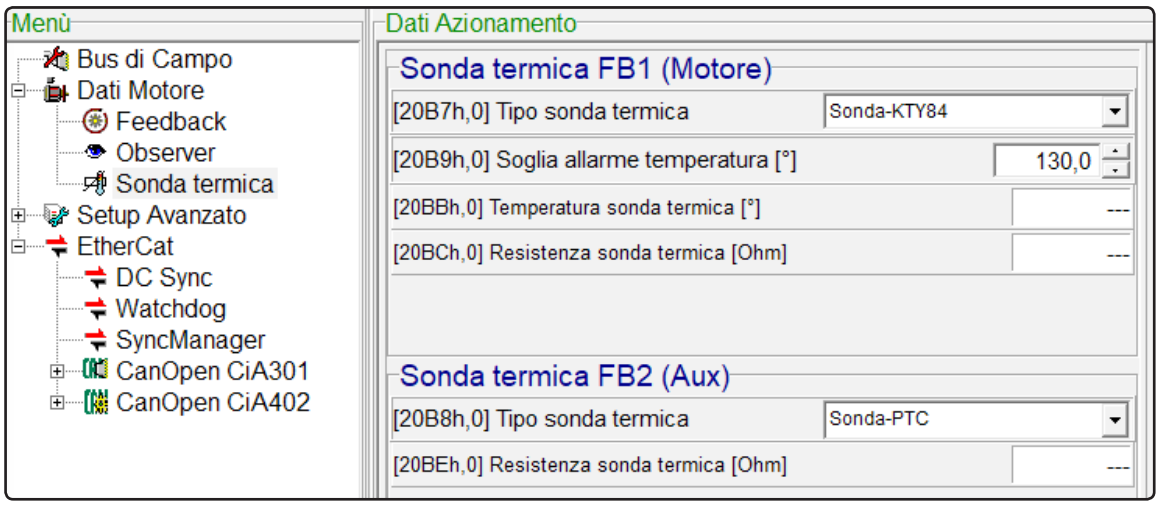
N.B: excessively decreasing the bandwidth of the Observer could cause instability in the speed regulator with the loss of control of the motor, with the risk of damaging the application where it's mounted.

5.05 Thermistor parameters

This paragraph lists the parameters related to the thermistor.

Index	Name	Type	Map	Attributes
20B7h	**Type thermistor FB1 (motor)**	UINT16	No	rw
20B8h	**Type thermistor FB2 (Aux)**	UINT16	No	rw
20B9h	**Temperature alarm threshold FB1 (motor)**	UINT16	No	rw
20BAh	**Temperature alarm resistance FB1 (motor)**	UINT16	No	rw
20BBh	**Thermistor temperature FB1 (motor)**	INT16	No	ro
20BCh	**Thermistor resistance FB1 (motor)**	UINT16	No	ro
20BDh	**Thermistor temperature FB2 (Aux)**	INT16	No	ro
20BEh	**Thermistor resistance FB2 (Aux)**	UINT16	No	ro

These parameters can be set in the Caliper menu "***Motor data-Thermistor***" in which the following screen appears:



5.05.1 Object 20B7h – Type thermistor FB1 (motor)

Index	Name	Type	Range	Default	Unit	Map	Attributes
20B7h	Type thermistor FB1 (motor)	UINT16	0 ... 2	0		No	rw

This object is used to set what type of thermistor is mounted on the motor, if present.

Type -thermistor FB1 (motor): description	
0	No thermal sensor
1	Sensor PTC
2	Sensor KTY84

5.05.2 Object 20B8h – Type thermistor FB2 (Aux)

Index	Name	Type	Range	Default	Unit	Map	Attributes
20B8h	Type thermistor FB2 (Aux)	UINT16	0 ... 2	0		No	rw

This object is used to set what type of auxiliary thermistor is used, if present.

Type -thermistor FB2 (aux): description	
0	No thermal sensor
1	Sensor PTC
2	Sensor KTY84

5.05.3 Object 20B9h – Temperature alarm threshold FB1 (Motor)

Index	Name	Type	Range	Default	Unit	Map	Attributes
20B9h	Temperature alarm threshold FB1 (motor)	UINT16	1...10000	1300	°/10	No	rw

This object is used to set a temperature threshold for the KTY84 sensor mounted on the motor, if the measured temperature exceeds this threshold an alarm appears.

5.05.4 Object 20BAh – Temperature alarm resistance FB1 (Motor)

Index	Name	Type	Range	Default	Unit	Map	Attributes
20BAh	Temperature alarm resistance FB1 (motor)	UINT16	1...65535	1322	Ohm	No	rw

This object is used to set the maximum resistance allowed for the PTC sensor mounted on the motor, if its resistance exceeds this value an alarm appears.

5.05.5 Object 20BBh – Thermistor temperature FB1 (Motor)

Index	Name	Type	Range	Default	Unit	Map	Attributes
20BBh	Thermistor temperature FB1 (motor)	INT16	-32768 ... 32767		°	No	ro

This object reports the temperature measured by the Kty84 sensor mounted on the motor.

PRELIMINARY

5.05.6 Object 20BCh – Thermistor resistance FB1 (Motor)

Index	Name	Type	Range	Default	Unit	Map	Attributes
20BCh	Thermistor resistance FB1 (motor)	UINT16	0...65535	---	Ohm	No	ro

This object reports the measured resistance of the PTC sensor mounted on the motor.

5.05.7 Object 20BDh – Thermistor temperature FB2 (Aux)

Index	Name	Type	Range	Default	Unit	Map	Attributes
20BDh	Thermistor temperature FB2 (Aux)	INT16	-32768 ... 32767		°	No	ro

This object reports the temperature measured by the auxiliary Kty84 sensor.

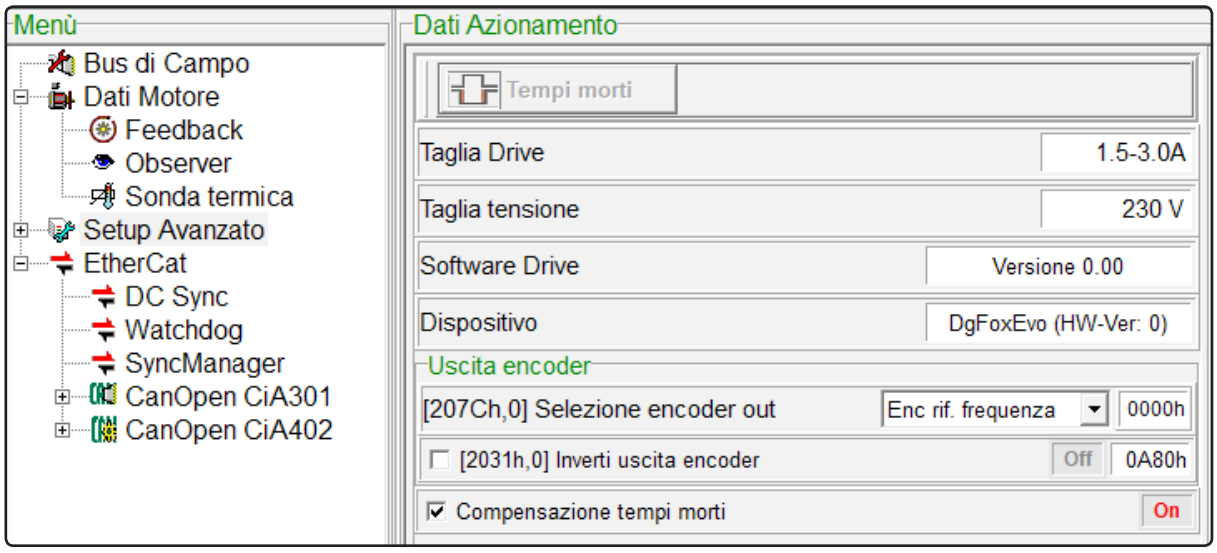
5.05.8 Object 20BEh – Thermistor resistance FB2 (Aux)

Index	Name	Type	Range	Default	Unit	Map	Attributes
20BEh	Thermistor resistance FB2 (Aux)	UINT16	0...65535	---	Ohm	No	ro

This object reports the measured resistance of the auxiliary PTC sensor.

5.06 Advanced Setup

Selecting the Caliper menù "***Advanced setup***" the following screen appears:



In this screen you can read some useful information regarding the used Drive, such as its rated voltage and current,its model and the installed firmware.

You can also start an automatic procedure for measuring the inverter dead times. When this measurement is completed,you can choose to enable or disable the dead times compensation; it is recommended to enable it because it improves the performance of the current regulator and the sensorless algorithm.

Finally, from this screen you can also set two flags contained on the objects ***2031h*** and ***207Ch***. These flags will be described below.

5.06.1 Object 2031h, bit 10 – Reverse encoder output

Index	Name	Type	Range	Default	Unit	Map	Attributes
2031h	Reverse encoder output	UINT16	0..65535		Bit	No	rw

This flag is used to reverse the encoder signal that will be transmitted in output.

value	description
0	normal simulated encoder
1	reversed simulated encoder

5.06.2 Object 207Ch, bit 4,5 – Encoder output selection

Index	Name	Type	Range	Default	Unit	Map	Attributes
207Ch	Encoder output selection	UINT16	0..65535		Bit	No	rw

This flag is used to select what "encoder signal" you want to transmit on the output:

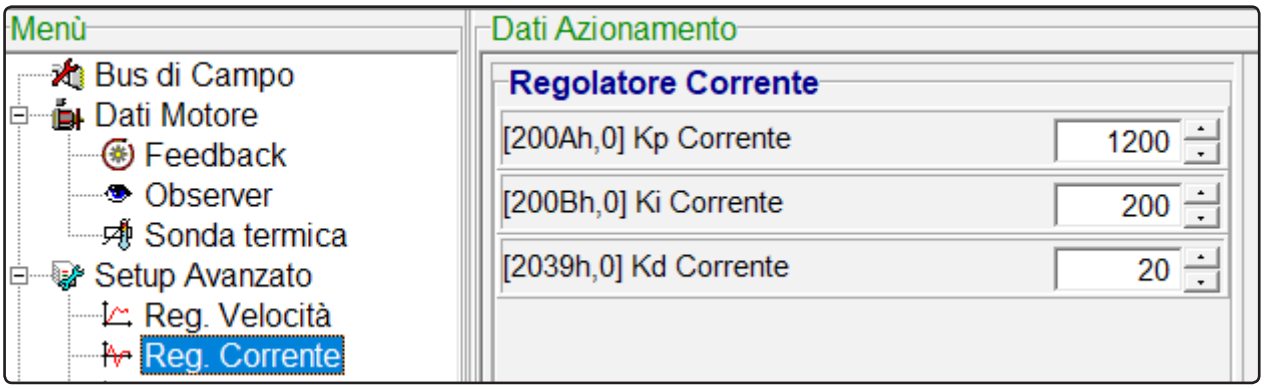
Object 207Ch- description of bit 4 and 5	
value	**Description**
0 (00b)	***Enc. rif. frequency***: the Drive transmits on the output a copy of the position reference signal
1 (01b)	***Enc. sim. resolver***: the Drive transmits on the output the signal of a simulated resolver
2 (10b)	***Enc. increment+Hall***: the Drive transmits on the output a copy of the measured position signal

5.07 Speed's and current's regulators

This paragraph describes the parameters used for the tuning of the current's and speed's regulators. The regulators currently used are PI (proportional-integral) regulators. The derivative gains are not used at the moment; they were included because they will be implemented in the future.

Index	Name	Type	Map	Attributes
200Ah	**Kp current regulator**	UINT16	No	rw
200Bh	**Ki current regulator**	UINT16	No	rw
2039h	**Kd current regulator**	UINT16	No	rw
2008h	**Kp speed regulator**	UINT16	No	rw
2009h	**Ki speed regulator**	UINT16	No	rw
2038h	**Kd speed regulator**	UINT16	No	rw

These parameters can be set in the Caliper menus "***Advanced setup-Current regulator***" and "***Advanced setup-Speed regulator***" in which the following screens appear:



5.07.1 Object 200Ah – Kp current regulator

Index	Name	Type	Range	Default	Unit	Map	Attributes
200Ah	Kp current regulator	UINT16	1... 65535	1200		No	rw

This object contains the proportional gain value of the current regulator.

5.07.2 Object 200Bh – Ki current regulator

Index	Name	Type	Range	Default	Unit	Map	Attributes
200Bh	Ki current regulator	UINT16	0... 65535	200		No	rw

This object contains the integral gain value of the current regulator.

5.07.3 Object 2039h – Kd current regulator

Index	Name	Type	Range	Default	Unit	Map	Attributes
2039h	Kd current regulator	UINT16	0... 65535	20		No	rw

This object contains the derivative gain value of the current regulator. Currently it's not used by the firmware.

5.07.4 Object 2008h – Kp speed regulator

Index	Name	Type	Range	Default	Unit	Map	Attributes
2008h	Kp speed regulator	UINT16	1... 65535	1000		No	rw

This object contains the proportional gain value of the speed regulator.

5.07.5 Object 2009h – Ki speed regulator

Index	Name	Type	Range	Default	Unit	Map	Attributes
2009h	Ki speed regulator	UINT16	0... 65535	300		No	rw

This object contains the integral gain value of the speed regulator.

5.07.6 Object 2038h – Kd speed regulator

Index	Name	Type	Range	Default	Unit	Map	Attributes
2038h	Kd speed regulator	UINT16	0... 65535	20		No	rw

This object contains the derivative gain value of the speed regulator. Currently it's not used by the firmware.

PRELIMINARY

5.08 Flux regulator

In this section are described three PIDs used to control the magnetic flux of the induction motors. These regulators are used for flux weakening (reduction of the magnetic flux in the air gap), which is the only choice when the motor has to go faster than the nominal speed. If the flux was kept constant, the emf of the motor would continue to increase proportionally to the speed, up to the point where the Drive would no longer be able to supply the voltage required by the motor. The pids mentioned above are:

- the "***Flux PID***";
- the "***Field weakening PID***";
- the "***desaturation Vd PID***".

The "***Flux PID***" is optional. When enabled, it takes as error the difference between the maximum quadrature voltage ***Vq*** that the Drive can supply (reduced with the costant value reported on index ***20B5h*** for safety) and the ***Vq*** actually produced. On the output it returns the reference flux that the Drive must reach.

The "***Field weakening PID*** " has the major role for the reduction of the magnetic flux, since it controls the ***Id*** current which magnetizes the motor. This PID can take as error one of the two following values:

- if the "***Flux PID***" is enabled the error corresponds to the difference between the flux calculated by this PID and the flux estimated by the chosen Observer;
- if the "***Flux PID***" is enabled the error corresponds to the difference between the maximum quadrature voltage ***Vq*** that the Drive can supply (reduced with the costant value reported on index ***20B5h*** for safety) and the ***Vq*** actually produced.

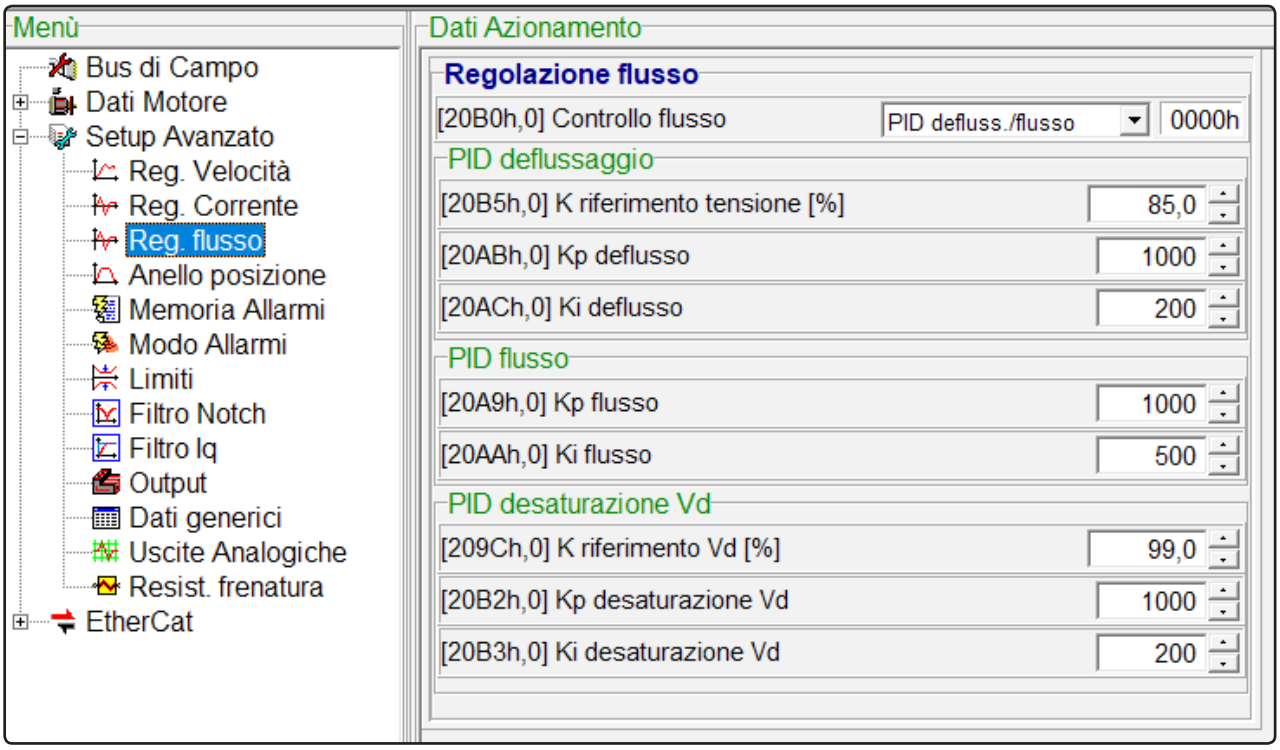
The "***Flux PID***" improves the performance of the control, but it cannot be used in all cases since the "***indirect***" Observer does not estimate the flux value.

When you have to reach a very high speed the flux weakening alone is not sufficient to reduce the voltage required by the motor: the torque current ***Iq*** must be limited to reduce the voltage drop across the windings. This is the task of the "***desaturation Vd PID*** ", that controls the maximum limit of the current ***Iq***. This PID takes as error the difference between the maximum direct voltage ***Vd*** that the Drive can supply (reduced with the costant value reported on index ***209Ch*** for safety) and the ***Vd*** actually produced.

PRELIMINARY

Index	Name	Type	Map	Attributes
209Ch	**K reference Vd [%]**	UINT16	No	rw
20A9h	**Flux Kp**	UINT16	No	rw
20AAh	**Flux Ki**	UINT16	No	rw
20ABh	**Field weakening Kp**	UINT16	No	rw
20ACh	**Field weakening Ki**	UINT16	No	rw
20B0h	**Flux Control**	UINT16	No	rw
20B2h	**Kp desaturation Vd**	UINT16	No	rw
20B3h	**Ki desaturation Vd**	UINT16	No	rw
20B5h	**K voltage reference [%]**	UINT16	No	rw

These parameters can be set in the Caliper menu "***Advanced setup-Flux regulator***" in which the following screen appears:



5.08.1 Object 209Ch – K reference Vd [%]

Index	Name	Type	Range	Default	Unit	Map	Attributes
209Ch	K reference Vd [%]	UINT16	0,1... 99,9	99	%	No	rw

This object sets a reduction factor (in percentage) of the Maximum direct voltage ***Vd*** that the Drive can supply. It's used with the "***desaturation Vd PID***" (see *"5.08 Flux regulator" pag. 152*). Limiting the voltage Vd reduces the torque that can be achieved by an induction motor with weakened flux, but makes the control more stable.

5.08.2 Object 20A9h – Flux Kp

Index	Name	Type	Range	Default	Unit	Map	Attributes
20A9h	Flux Kp	UINT16	0... 65535	1000		No	rw

This object sets the proportional gain of the "***Flux PID***" (see *"5.08 Flux regulator" pag. 152*).

5.08.3 Object 20AAh – Flux Ki

Index	Name	Type	Range	Default	Unit	Map	Attributes
20AAh	Flux Ki	UINT16	0... 65535	200		No	rw

This object sets the integral gain of the "***Flux PID***" (see *"5.08 Flux regulator" pag. 152*).

5.08.4 Object 20ABh – Field weakening Kp

Index	Name	Type	Range	Default	Unit	Map	Attributes
20ABh	Field weakening Kp	UINT16	0... 65535	1000		No	rw

This object sets the proportional gain of the "***Field weakening PID***" (see *"5.08 Flux regulator" pag. 152*).

5.08.5 Object 20ACh – Field weakening Ki

Index	Name	Type	Range	Default	Unit	Map	Attributes
20ACh	Field weakening Ki	UINT16	0... 65535	500		No	rw

This object sets the integral gain of the "***Field weakening PID***" (see *"5.08 Flux regulator" pag. 152*).

5.08.6 Object 20B0h bit 1 – Flux Control

Index	Name	Type	Range	Default	Unit	Map	Attributes
20B0h	Flux Control	UINT16	0... 65535	0		No	rw

The bit 1 of this object is used as a flag to enable/disable the "***Flux PID***" (see *"5.08 Flux regulator" pag. 152*).

Bit 1	*description*
0	the "***Flux PID***" and the "***Field weakening PID***" work together
1	the "***Flux PID***" is disabled, the "***Field weakening PID***" works alone

PRELIMINARY

5.08.7 Object 20B2h – Kp desaturation Vd

Index	Name	Type	Range	Default	Unit	Map	Attributes
20B2h	Kp desaturation Vd	UINT16	0... 65535	1000		No	rw

This object sets the proportional gain of the "***Desaturation Vd PID***" (see *"5.08 Flux regulator" pag. 152*).

5.08.8 Object 20B3h – Ki desaturation Vd

Index	Name	Type	Range	Default	Unit	Map	Attributes
20B3h	Ki desaturation Vd	UINT16	0... 65535	200		No	rw

This object sets the integral gain of the "***Desaturation Vd PID***" (see *"5.08 Flux regulator" pag. 152*).

5.08.9 Object 20B5h – K voltage reference [%]

Index	Name	Type	Range	Default	Unit	Map	Attributes
20B5h	K voltage reference [%]	UINT16	1... 100	85	%	No	rw

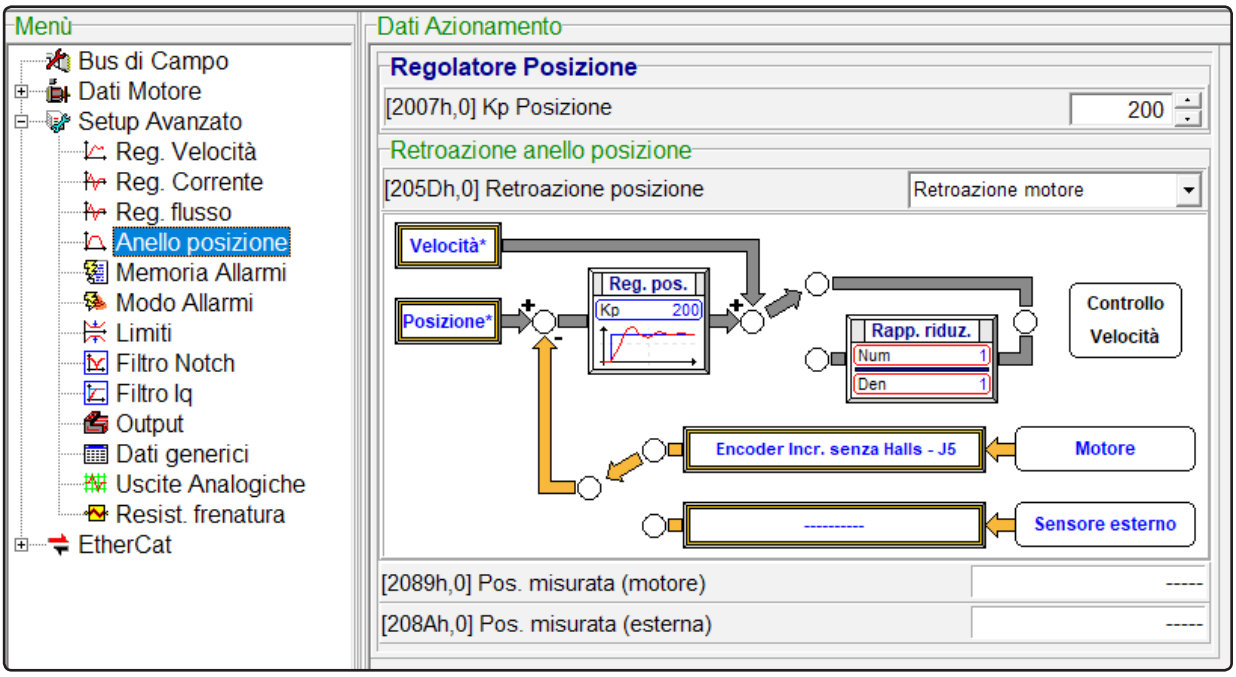
This object sets a reduction factor (in percentage) of the Maximum quadrature voltage ***Vq*** that the Drive can supply. It's used with the "***flux PID***" and the "***field weakening PID***" (see *"5.08 Flux regulator" pag. 152*). Limiting the voltage ***Vq*** reduces the magnetic flux that can be produced by an induction motor, but improves the performance of the control.

5.09 Position loop

This paragraph lists the parameter used for the tuning of the position regulator.

Index	Name	Type	Map	Attributes
2007h	**Position Kp**	UINT16	No	rw
205Dh	**Position feedback**	UINT16	No	rw
205Eh	**Pulses/revolution**	INT32	No	rw
2087h	**Gear ratio - num**	UINT32	No	rw
2088h	**Gear ratio - den**	UINT32	No	rw
2089h	**Pos. measured (motor)**	UINT32	Yes	ro
208Ah	**Pos. measured (extern)**	UINT32	Yes	ro

These parameters can be set in the Caliper menu "***Advanced setup-position loop***" in which the following screen appears:



5.09.1 Object 2007h – Kp position regulator

Index	Name	Type	Range	Default	Unit	Map	Attributes
2007h	Kp position regulator	UINT16	0...4000	200		No	rw

This object contains the proportional gain value of the position regulator.

5.09.2 Object 205Dh – Position feedback

Index	Name	Type	Range	Default	Unit	Map	Attributes
205Dh	Position feedback	UINT16	0...2	0		No	rw

This object is used to select what kind of position feedback you want to use

Position feedback		
0	motor feedback	The Drive uses the measured position of the motor as feedback
1	Encoder reference-J7	The Drive uses as feedback the position measured by an external sensor, received on input J7
2	Encoder feedback -J5	The Drive uses as feedback the position measured by an external sensor, received on input J5

PRELIMINARY

5.09.3 Object 205Eh– Pulses/revolution

Index	Name	Type	Range	Default	Unit	Map	Attributes
205Eh	Pulses/revolution	UINT32	1...65536	1024		No	rw

This object contains the number of pulses per revolution received by the external encoder. If an encoder oftype "***Channel A-B***" is used, the value of "***pulses per revolution***" will be internally multiplied by 4, because in this mode the Drive can read the encoder pulses with a resolution 4 times higher.

5.09.4 Object 2087h– Gear ratio - num

Index	Name	Type	Range	Default	Unit	Map	Attributes
2087h	Gear ratio - num	UINT32	0 ... 7FFFFFFh	1		No	rw

This object sets the numerator of a reduction ratio which multiplies with the speed produced by the position regulator, which will later become the speed reference of the Drive. This speed is calculated as follow:

$$Reference\_speed = \frac{gear\_ratio\_numerator}{gear\_ratio\_denominator} * Output\_speed$$

5.09.5 Object 2088h– Gear ratio - den

Index	Name	Type	Range	Default	Unit	Map	Attributes
2088h	Gear ratio - den	UINT32	1 ... 7FFFFFFh	1		No	rw

This object sets the denominator of a reduction ratio which multiplies with the speed produced by the position regulator, which will later become the speed reference of the Drive. This speed is calculated using the formula on paragraph *"5.09.4 Object 2087h– Gear ratio - num" pag. 156*.

5.09.6 Object 2089h – Pos. measured (motor)

Index	Name	Type	Range	Default	Unit	Map	Attributes
2089h	Pos. measured (motor)	UINT32	±7FFFFFFFh			Si	ro

This object contains the value of the normalized measured position of the motor, with a resolution of 16 bit.

5.09.7 Object 208Ah – Pos. measured (extern)

Index	Name	Type	Range	Default	Unit	Map	Attributes
208Ah	Pos. measured (extern)	UINT32	±7FFFFFFFh			Si	ro

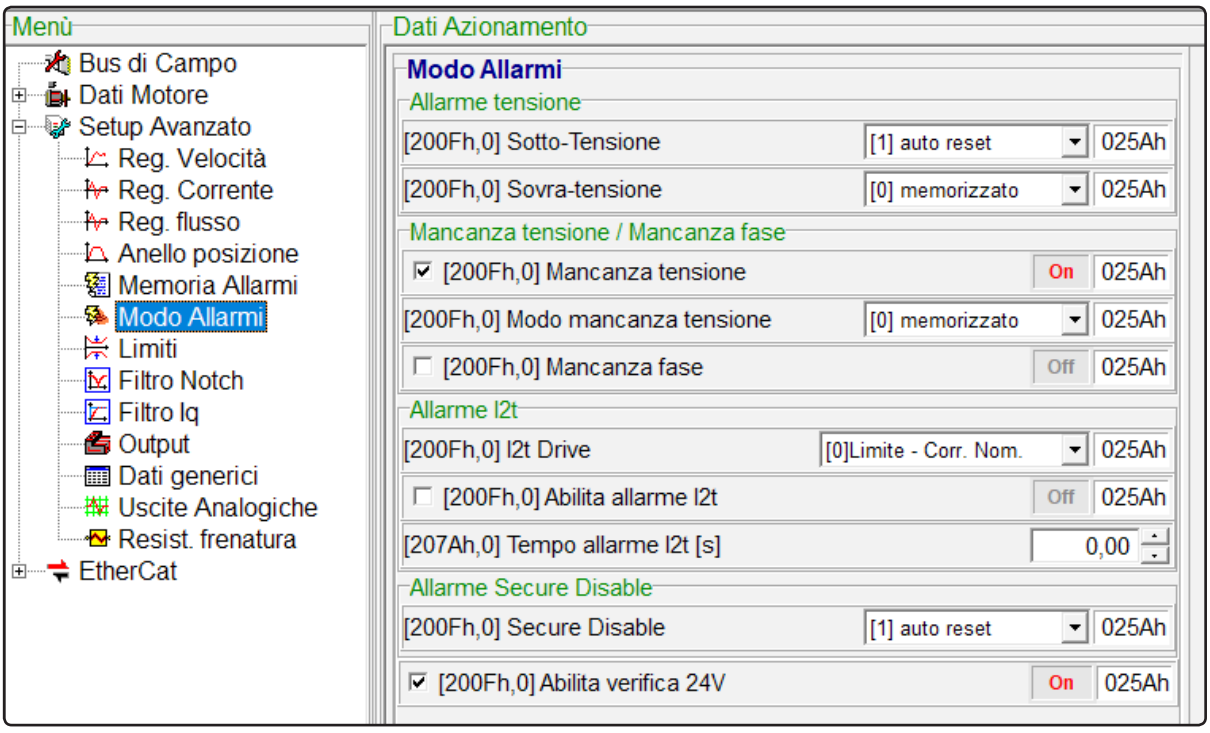
This object contains the value of the normalized measured position of an absolute external encoder, with a resolution of 16 bit.

5.10 Alarm mode

This paragraph lists the parameters used to manage the Drive alarms separately.

Index	Name	Type	Map	Attributes
200Fh	**Alarm mode**	UINT16	No	rw
207Ah	**Time alarm I2t**	UINT16	No	rw

These parameters can be set in the Caliper menu "***Advanced setup-Alarm mode***" in which the following screen appears:



PRELIMINARY

5.10.1 Object 200Fh – Alarms mode

Index	Name	Type	Range	Default	Unit	Map	Attributes
200Fh	Alarms mode	UINT16	0...65535	025A		No	rw

This object contains various flags used to set the Drive's alarms, as showed in the following table.

Bit	Description	
0	**Alarm mode over voltage**	
	Value	**Description**
	0	***Stored***: the "***over voltage***" alarm is signaled and the Drive goes in "***Fault***" state. You have to command a reset or restart the Drive if you want to restore its state.
	1	***Auto reset***: the "***over voltage***" alarm is signaled and the Drive goes in "***Fault***" state. When the voltage drops below the "***over voltage***" threshold, the Drive exits from the "***Fault***" state without the need of an external reset.
1	**Alarm mode undervoltage**	
	Value	**Description**
	0	***Stored***: the "***undervoltage***" alarm is signaled and the Drive goes in "***Fault***" state. You have to command a reset or restart the Drive if you want to restore its state.
	1	***Auto reset***: the "***undervoltage***" alarm is signaled and the Drive goes in "***Fault***" state. When the voltage rises over the "***undervoltage***" threshold, the Drive exits from the "***Fault***" state without the need of an external reset.
2	**Alarm mode I2t drive**	
	Value	**Description**
	0	***Limit to nominal current***: when the Drive "***I2t alarm***" occurs, the maximum current deliverable is limited to the value of the Drive's rated current, and remains there regardless of the evolution of the estimated temperature of the Drive.
	1	***Auto reset cyclic***: if the Drive "***I2t alarm***" occurs, it will be automatically cancelled as soon as the estimated temperature of the Drive drops below the fixed threshold.
3	**Alarm mode secure disable (Only TOMCAT)**	
	Value	**Description**
	0	***Stored***: the "***secure disable***" alarm is signaled and the Drive goes in "***Fault***" state. You have to command a reset or restart the Drive if you want to restore its state.
	1	***Auto reset***: the "***secure disable***" alarm is signaled and the Drive goes in "***Fault***" state. When the voltage is restored on the inputs assigned to this function, the Drive exits from the "***Fault***" state without the need of an external reset.
4	**Alarms save**	
	Value	**Description**
	0	***off***: The Drive doesn't store the alarms that have occurred
	1	***on***: The Drive stores the alarms that have occurred on a list
5	**Enable alarm I2t**	
	Value	**Description**
	0	***off***: alarm not enabled
	1	***on***: alarm enabled
6	**Enable check 24 V**	
	Value	**Description**
	0	***off***: alarm not enabled
	1	***on***: alarm enabled

Bit	Description	
7	**Alarm mode voltage fault**	
	Value	**Description**
	0	***Stored***: the "***voltage fault***" alarm is signaled and the Drive goes in "***Fault***" state. You have to command a reset or restart the Drive if you want to restore its state
	1	***Auto reset***: the "***voltage fault***" alarm is signaled and the Drive goes in "***Fault***" state. When the voltage is restored to its normal value, the Drive exits from the "***Fault***" state without the need of an external reset.
8	**Phase fault**	
	Value	**Description**
	0	***off***: alarm not enabled
	1	***on***: alarm enabled
9	**Voltage fault**	
	Value	**Description**
	0	***off***: alarm not enabled
	1	***on***: alarm enabled
10 ... 15	***Reserved***	

5.10.2 Object 207Ah – Time alarm I2t

Index	Nome	Tipo	Range	Default	Unit	Map	Attributi
207Ah	Time alarm I2t	UINT16	0...60000	0	s/100	No	rw

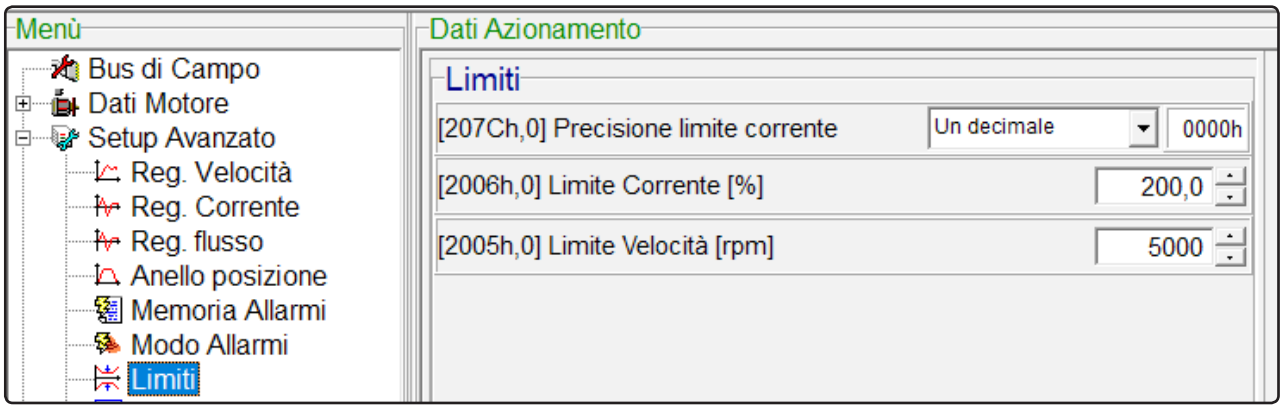
This object stores the reference time "***time alarm I2t***", used to control both the Drive's and the motor's current. If the drive current rises over twice its rated value for a period longer than the one set on this parameter, then the alarm 27 occurs. If the motor current rises over twice its rated value for a period longer than the one set here, then the alarm 28 occurs.

5.11 Limit

This paragraph lists the parameters used to set the speed's and current's maximum limits used by the Drive.

Index	Name	Type	Map	Attributes
2005h	**Speed limit**	UINT16	No	rw
2006h	**Current limit**	UINT16	Yes	rw
207Ch	**Various flags**	UINT16	No	rw

These parameters can be set in the Caliper menu "***Advanced setup-Limit***" in which the following screen appears:



5.11.1 Object 2005h – Speed limit

Index	Name	Type	Range	Default	Unit	Map	Attributes
2005h	Speed limit	UINT16	0...65000	5000	rpm	No	rw

This object contains the maximum limit, expressed in [rpm], of the speed reference's absolute value.

5.11.2 Object 2006h – Current limit

Index	Name	Type	Range	Default	Unit	Map	Attributes
2006h	Current limit	UINT16	0...10000	2000	‰	Yes	rw

This object contains the maximum limit of the current reference's absolute value, expressed in parts per thousands of the rated motor current. For example, if the rated motor current is of 2,00 [A] and the current's limit is set on 2000, then the maximum current allowed is equal to 4,00 [A].

5.11.3 Object 207Ch – Various Flags

Index	Name	Type	Range	Default	Unit	Map	Attributes
207Ch	Various Flags	UINT16	0...65535	0		No	rw

This object contains several flags, which are used in different part of the Firmware. In this menù only the "***Current limit accuracy***" flag is used, but for the sake of completeness the following table shows all of them:

Bit	Description	
0,1	**Frequency mode**	
	Value	**Description**
	0	Channel A-B
	1	Frequency direction
	2	Pulses CW/CCW
2	**Mode end position**	
	Value	**Description**
	0	End current quote using the position control
	1	End current quote using the speed control (reference speed=0)
3	**Compatibility PID (only DgFox/TomCat)**	
	Value	**Description**
	0	***Off***: The PID gains are taken as they are
	1	***On***: To mantain the compatibility with the tuning of PIDs made in older applications the PID gains are converted inside the Firmware
4,5	**Select encoder out (only NTT)**	
	Value	**Description**
	0	Enc. rif. frequency
	1	Enc. sim. resolver
	2	Enc. increment+Hall
6	**set_tmc_out2_analog2 (only Tomcat)**	
	Value	**Description**
	0	the analog input 2 is used as a digital input (n°7)
	1	the analog input 2 is used as a digital input (n°7)
7	**Current limit accuracy**	
	Value	**Description**
	0	A decimal
	1	No decimal
8 ... 15	***Reserved***	

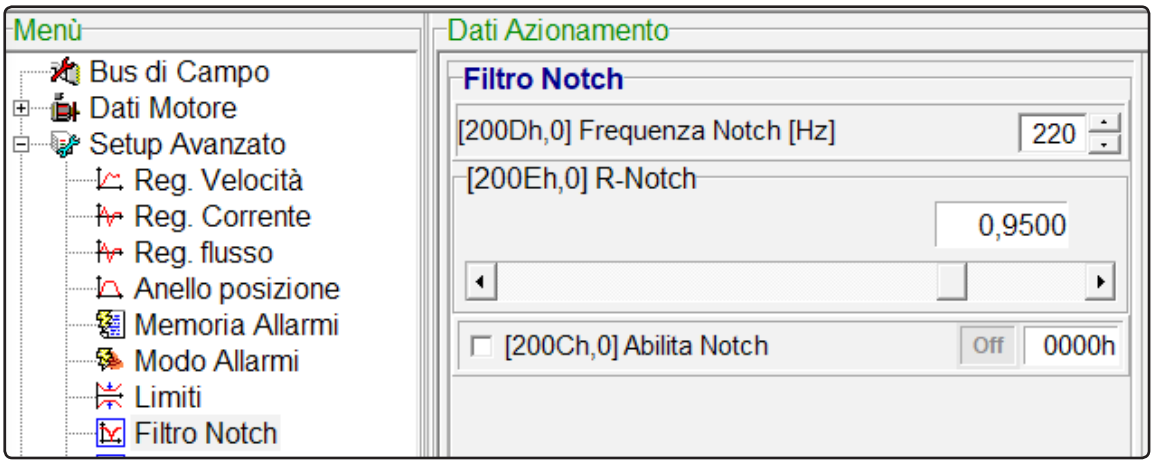
5.12 Filters

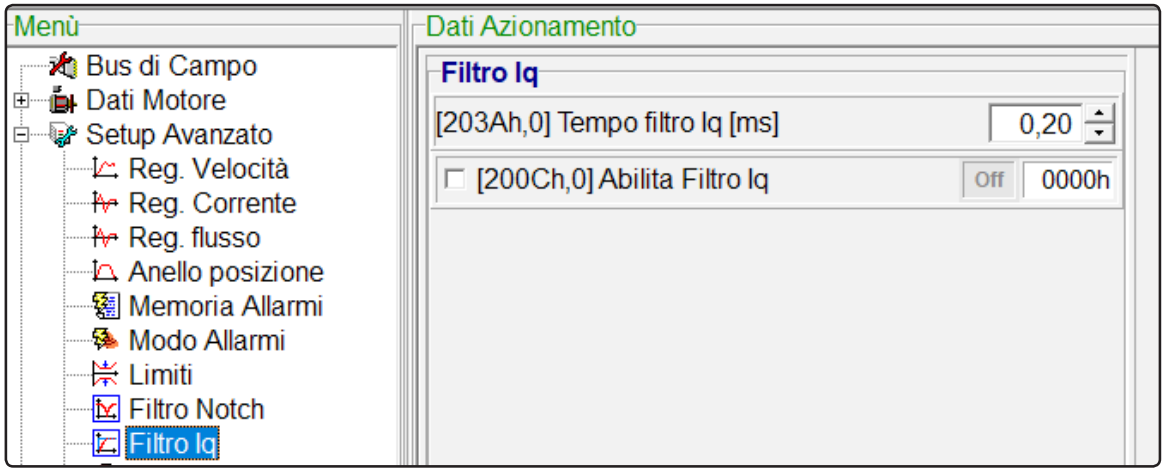
This paragraph lists the parameters used to tune the ***Notch filter*** and the ***Iq filter*** of the Drive. The Notch filter is used to eliminate a certain frequency from the Torque current (Iq) produced by the speed regulator. This Filter could be useful to avoid undesired resonances in the system composed by the motor and its load.On the other hand the Iq filter is used to filter the Iq reference used by the current regulator. This filter can be useful to avoid current's overshoot, but makes the current regulator less dynamic.

Index	Name	Type	Map	Attributes
200Ch	**Filters Flags**	UINT16	No	rw
200Dh	**Notch frequency**	UINT16	No	rw
200Eh	**R-Notch (bandwidth)**	UINT16	No	rw
203Ah	**Time filter Iq**	UINT16	No	rw

PRELIMINARY

These parameters can be set in the Caliper menus "***Advanced setup-Notch filter***" and "***Advanced setup-Iq filter***" in which the following screens appear:





5.12.1 Object 200Ch Bit 0,1 – Filters flags

Index	Name	Type	Range	Default	Unit	Map	Attributes
200Ch	Filters flags	UINT16	0...65535	0		No	rw

This object is referred in several paragraphs of this manual, because it contains various flags used in different points of the firmware. The bits 0 and 1 are two flags used to enable or disable the ***Notch filter*** or the ***iq filter***, as showed in the following table:

Bit	Description	
0	**Enable/disable Notch filter**	
	Value	**Description**
	0	Notch filter disabled
	1	Notch filter enabled
1	**Enable/disable Low-pass Iq filter**	
	Value	**Description**
	0	Low-pass Iq filter disabled
	1	Low-pass Iq filter enabled
2 ... 15	***Reserved***	

5.12.2 Object 200Dh – Notch frequency

Index	Name	Type	Range	Default	Unit	Map	Attributes
200Dh	Notch frequency	UINT16	2...400	220	Hz	No	rw

This object is used to set the frequency that the Notch filter should eliminate.

5.12.3 Object 200Eh – R-Notch (bandwidth)

Index	Name	Type	Range	Default	Unit	Map	Attributes
200Eh	R-Notch (bandwidth)	UINT16	0,8000... 0,9900	0,9500		No	rw

This object is used to set an adimensional parameter, R. R is used to define the bandwidth around the Notch frequency. All the frequencies inside this range will still be partially reduced by the Notch filter.

5.12.4 Object 203Ah – Time filter Iq

Index	Name	Type	Range	Default	Unit	Map	Attributes
203Ah	Time filter Iq	UINT16	1...3000	20	ms/100	No	rw

This parameter represents the time costant of the low-pass Iq filter and is expressed in hundredths of [ms] (the two least significant digits are interpreted as decimal fractions of [ms], for example the value 20 is intended as 0,20 [ms].

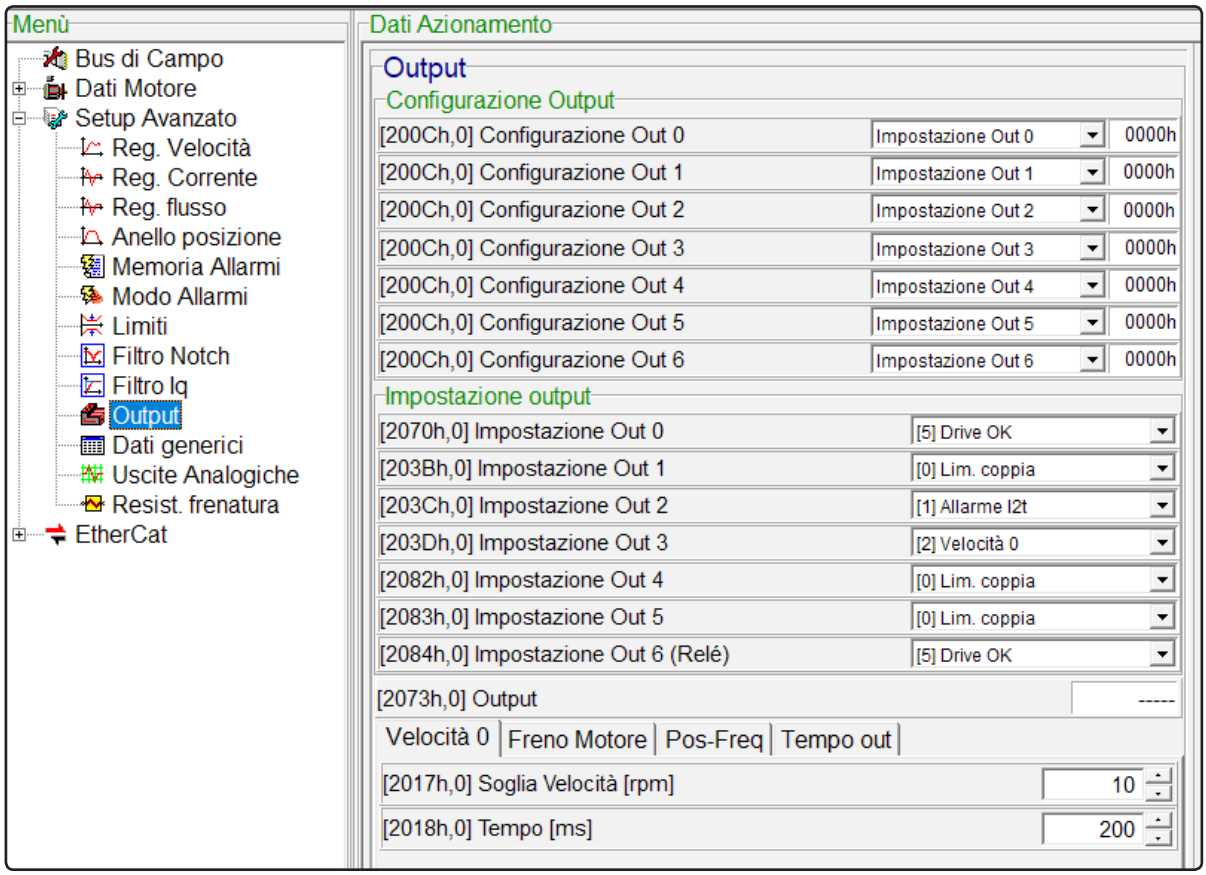
5.13 Output

This paragraph describes the parameters used to set the Drive's outputs.

Index	Name	Type	Map	Attributes
200Ch	**Configuration Outputs**	UINT16	No	rw
2013h	**Time brake enable**	UINT16	No	rw
2014h	**Time brake disable**	UINT16	No	rw
201Eh	**Current offset**	INT16	No	rw
2015h	**Deceleration**	UINT16	No	rw
2016h	**Speed brake enable**	UINT16	No	rw
2017h	**Speed threshold**	UINT16	No	rw
2018h	**Time**	UINT16	No	rw
203Bh	**Setting Out1**	UINT16	No	rw
203Ch	**Setting Out2**	UINT16	No	rw
203Dh	**Setting Out3 (NTT only)**	UINT16	No	rw
2070h	**Setting Out0**	UINT16	No	rw

Index	Name	Type	Map	Attributes
2071h	**Divisor count encoder**	UINT16	No	rw
2082h	**Setting Out4 (NTT only)**	UINT16	No	rw
2083h	**Setting Out5 (NTT only)**	UINT16	No	rw
2084h	**Setting Out6 (relay) (NTT only)**	UINT16	No	rw
2073h	**Output**	UINT16	Si	rw
208Bh	**Time output**	ARRAY(UINT16)	No	rw

These parameters can be set in the Caliper menu "***Advanced setup-Output***" in which the following screen appears:



PRELIMINARY

5.13.1 Object 200Ch, bit 4,5,6,7,8,9,10 – Configuration outputs

Index	Name	Type	Range	Default	Unit	Map	Attributes
200Ch	Configuration outputs	UINT16	0...65536	0		No	rw

This object is referred in several paragraphs of this manual, because it contains various flags used in different points of the firmware. The bits 4,5,6,7,8,9 and 10 of this parameter are used to set how the Drive's outputs will be controlled:

- When the bit assigned to an output is set to 0, the output in question will report the value chosen via Caliper, as described on paragraph *"5.13.12 Object 2070h – Setting Out0" pag. 166*
- When the bit assigned to an output is set to 1, the output in question will be controlled via Bus. This function allows you to use the Drive as an extension module for the outputs of a PLC to which it is connected

Bit	Name	Range	Default
4	Configuration Out0	0 = Setting Out 0	Setting Out 0
		1 = Bus control	
5	Configuration Out1	0 = Setting Out 1	Setting Out 1
		1 = Bus control	
6	Configuration Out2	0 = Setting Out 2	Setting Out 2
		1 = Bus control	
7	Configuration Out3	0 = Setting Out 3	Setting Out 3
		1 = Bus control	
8	Configuration Out4	0 = Setting Out 4	Setting Out 4
		1 = Bus control	
9	Configuration Out5	0 = Setting Out 5	Setting Out 5
		1 = Bus control	
10	Configuration Out6	0 = Setting Out6	Setting Out 6
		1 = Bus control	

PRELIMINARY

5.13.2 Object 2013h – Time brake enable

Index	Name	Type	Range	Default	Unit	Map	Attributes
2013h	Time brake enable	UINT16	10..2000	200	msec	No	rw

This object is used to set the time delay between the brake de-energizing command and the actual mechanical stop.

5.13.3 Object 2014h – Time brake disable

Index	Name	Type	Range	Default	Unit	Map	Attributes
2014h	Time brake disable	UINT16	10..2000	200	msec	No	rw

This object is used to set the time delay between the brake energizing command and its actual releasing.

5.13.4 Object 201Eh – Current offset

Index	Name	Type	Range	Default	Unit	Map	Attributes
201Eh	Current offset	INT16	-32000 ...32000	0	A/100	No	rw

In the vertical axis applications the gravity excert a continuous torque on the motor. As a result of this torque, when the brake is released the motor doesn't remain in place, but it makes a little movement before the PID starts working and bring it back to its initial position. To avoid this problem you can set a "***current offset***", which produces an ever-present feedforward torque which cancels the torque caused by the gravity. The value of this current has to be set manually. To find this value you have to follow these steps:

» Select "***Speed***" as control type, and give the ***Power on*** command without enabling the reference. This is equivalent to set the speed reference equal to 0;

» release the brake;

» Wait until the motor ends its transitory movement and stops moving;

» Read from Caliper the value of the current used by the Drive to blocks the motor;

» Set the current's value read in the previous step as the "***current offset***".

5.13.5 Object 2015h – Deceleration

Index	Name	Type	Range	Default	Unit	Map	Attributes
2015h	Deceleration	UINT16	1..60000	1000	rpm/s	No	rw

The Drive, before disabling the motor torque and closing the brake, checks whether the motor speed is lower than the threshold value set on the parameter "***Speed brake enable***" (***index 2016h***). If the motor speed is higher than this threshold, the Drive starts a deceleration ramp with the deceleration set on this parameter. Once the motor has reached the speed set in"***Speed brake enable***" the Drive closes the brake and removes the torque. However, If the Drive goes in the ***Fault*** state, the brake is closed immediately without checking the speed.

5.13.6 Object 2016h – Speed brake enable

Index	Name	Type	Range	Default	Unit	Map	Attributes
2016h	Speed brake enable	UINT16	1..500	4	rpm	No	rw

This object contains the minimum speed threshold below which the motor brake is closed during the stopping phase.

5.13.7 Object 2017h – Speed threshold

Index	Name	Type	Range	Default	Unit	Map	Attributes
2017h	Speed threshold	UINT16	1..1500	10	rpm	No	rw

This object is used to set a speed threshold. If the absolute value of the measured speed remains under the value set on this parameter for a time longer than the one set on the parameter "***Time*** (***index 2018h***)", than the output assigned to the function speed 0 rises to the logical value 1, i.e. it's signaled that the motor has reached the "***zero speed"***. Vice versa, in the opposite case (***measured speed*** > ***Speed threshold***),the same output shows the logical value 0.

5.13.8 Object 2018h – Time

Index	Name	Type	Range	Default	Unit	Map	Attributes
2018h	Time	UINT16	1..10000	200	msec	No	rw

This object is used to set a time period, named "***time***" on Caliper. If the absolute value of the measured speed remains under the value set on the parameter "***speed threshold*** (***index 2017h***)" for a time longer than the one set here, than the output assigned to the function "***speed 0***" rises to the logical value 1, i.e. it's signaled that the motor has reached the "zero speed".

5.13.9 Object 203Bh – Setting Out1

Index	Name	Type	Range	Default	Unit	Map	Attributes
203Bh	Setting Out1	UINT16	0...9	0		No	rw

This object contains the parameter "***Setting Out 1***". Depending on its value a certain function is assigned to the output 1. The possible assignable functions are the same for all the outputs; they're described in paragraph *"5.13.12 Object 2070h – Setting Out0" pag. 166*.

5.13.10 Object 203Ch – Setting Out2

Index	Name	Type	Range	Default	Unit	Map	Attributes
203Ch	Setting Out2	UINT16	0...9	1		No	rw

This object contains the parameter "***Setting Out 2***". Depending on its value a certain function is assigned to the output 2. The possible assignable functions are the same for all the outputs; they're described in paragraph *"5.13.12 Object 2070h – Setting Out0" pag. 166*.

5.13.11 Object 203Dh – Setting Out3 (NTT ONLY)

Index	Name	Type	Range	Default	Unit	Map	Attributes
203Dh	Setting Out3	UINT16	0...9	2		No	rw

This object contains the parameter "***Setting Out 3***". Depending on its value a certain function is assigned to the output 3. The possible assignable functions are the same for all the outputs; they're described in paragraph *"5.13.12 Object 2070h – Setting Out0" pag. 166*. This parameter is not supported by the DgFox and Tomcat Drives.

5.13.12 Object 2070h – Setting Out0

Index	Name	Type	Range	Default	Unit	Map	Attributes
2070h	Setting Out0	UINT16	0...9	5		No	rw

This object contains the parameter "***Setting Out 0***". Depending on the value of this parameter a certain function is assigned to the output 0. The description of the possible assignable functions are reported in the following table:

Value	Name	Description
0	***Torque limit***	The output turns on when the motor reaches the torque limit.
1	***I2t alarm***	The output turns on when there is an overcurrent
2	***Speed 0***	The output turns on when the motor has reached the "zero speed". This event is defined using the parameters "***Speed threshold***" (***index 2017h***) and "***Time***" (***index 2018h***)
3	***Target reached***	This function is used when the Drive works with a ***Positioner*** control. The output turns on when the motor has reached the target position.
4	***Motor brake***	The output turns on when the motor brakes is activated, which depends on the parameters "***Time brake enable***" (***index 2013h***), "***Time brake disable***" (***index 2014h***), "***Deceleration***" (***index 2015h***), "***Speed brake enable***" (***index 2016h***)
5	***Drive OK***	The output turns on when there is no alarms. The Drive can work
6	***Secure Disable***	The output turns on when the Drive is in the "***secure disable***" state
7	***Pos-freq out (only for DGFox and TomCat)***	The output produces a square waveform with a frequency that depends on a divider, which is a power of 2, and the position (in increments) of the motor on a single turn.
8	***Pos output***	This function is used when the Drive works with a "***Positioner Input-start***" control, the output produces a pulse of the set duration when the motor reach the current target position.
9	***Homing attained***	When any of the "***Homing***" procedure is performed, the output turns on and remains at 1 untill another "***Homing***" procedure is requested (or untill the Drive is turn off).

5.13.13 Object 2071h – Divisor count encoder

Index	Name	Type	Range	Default	Unit	Map	Attributes
2071h	Divisor count encoder	UINT16	1... 512	1		No	rw

This object contains the parameter "***divisor count encoder***", which is used by the function "***Pos-freq. out***" (see paragraph *"5.13.12 Object 2070h – Setting Out0" pag. 166*). "***Pos-Freq. Out***" produces a square waveform with a frequency which depends on a divider, which is a power of 2, and the position (in increments)of the motor on a single turn. To set the resolution of this output signal expressed in pulses per turn the Drive follows the following formula:

$$Enc_{out} = \frac{FB_{RESOLUTION}}{2\ DIV}$$

where:

» *Enc_out* is the resolution of the output square waveform;

» *FB_resolution* is the resolution of the motor feedback;

» *DIV* is the value of "***divisor count encoder***"

5.13.14 Object 2082h – Setting Out4 (NTT ONLY)

Index	Name	Type	Range	Default	Unit	Map	Attributes
2082h	Setting Out4	UINT16	0...9	0		No	rw

This object contains the parameter "***Setting Out 4***". Depending on its value a certain function is assigned to the output 4. The possible assignable functions are the same for all the outputs; they're described in paragraph *"5.13.12 Object 2070h – Setting Out0" pag. 166*. This parameter is not supported by the DgFox and Tomcat Drives.

5.13.15 Object 2083h – Setting Out5 (NTT ONLY)

Index	Name	Type	Range	Default	Unit	Map	Attributes
2083h	Setting Out5	UINT16	0...9	0		No	rw

This object contains the parameter "***Setting Out 5***". Depending on its value a certain function is assigned to the output 5. The possible assignable functions are the same for all the outputs; they're described in paragraph *"5.13.12 Object 2070h – Setting Out0" pag. 166*. This parameter is not supported by the DgFox and Tomcat Drives.

5.13.16 Object 2084h – Setting Out6 (relay) (NTT ONLY)

Index	Name	Type	Range	Default	Unit	Map	Attributes
2084h	Setting Out6	UINT16	0...9	5		No	rw

This object contains the parameter "***Setting Out 6***". Depending on its value a certain function is assigned to the output 6. The possible assignable functions are the same for all the outputs; they're described in paragraph *"5.13.12 Object 2070h – Setting Out0" pag. 166*. This parameter is not supported by the DgFox and Tomcat Drives.

5.13.17 Object 2073h – Output

Index	Name	Type	Range	Default	Unit	Map	Attributes
2073h	Output	INT16	0 ...65535			Si	rw

This object works as a buffer between the Field Bus and the 7 digital outputs of the Drive.

If one or more outputs are set on "***Bus control***" (see paragraph *"5.13.1 Object 200Ch, bit 4,5,6,7,8,9,10 – Configuration outputs" pag. 163*), this parameter will receives from the Field Bus the values that these outputs must show, and then it updates their values.

5.13.18 Object 208Bh – Time outputs

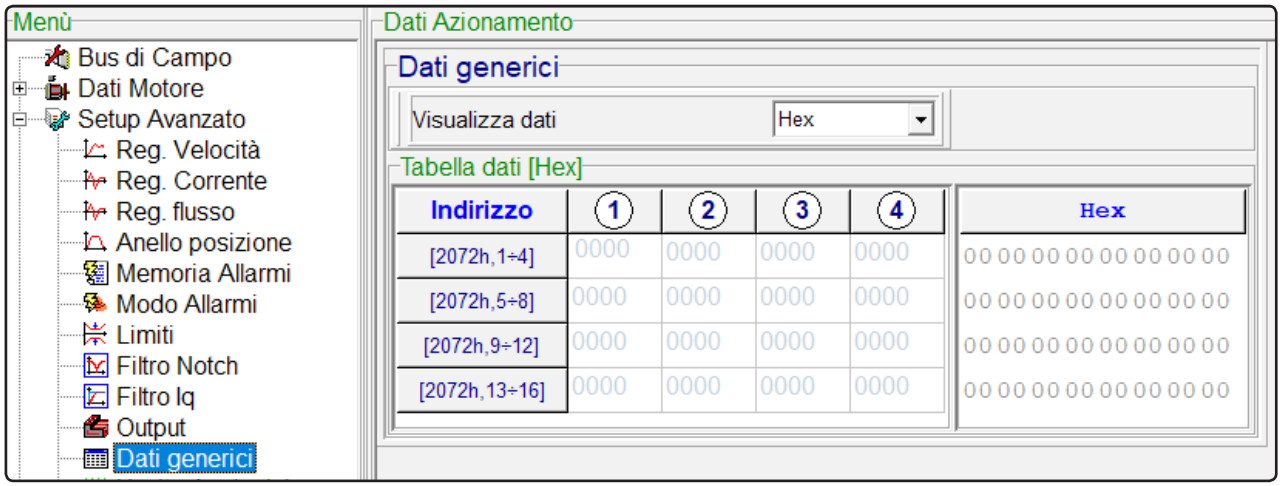
Index	Sub-index	Name	Type	Range	Default	Unit	Map	Attributes
208Bh	1	Time out0	UINT16	1... 6000	1	s/100	No	rw
	2	Time out1	UINT16	1... 6000	1	s/100	No	rw
	3	Time out2	UINT16	1... 6000	1	s/100	No	rw
	4	Time out3	UINT16	1... 6000	1	s/100	No	rw
	5	Time out4	UINT16	1... 6000	1	s/100	No	rw
	6	Time out5	UINT16	1... 6000	1	s/100	No	rw
	7	Time out6	UINT16	1... 6000	1	s/100	No	rw

This object contains the time intervals used by the function "***Pos output***" (see paragraph *"5.13.12 Object 2070h – Setting Out0" pag. 166*). "***Pos output***" makes the assigned outputs produce a pulse when the current target position has been reached; the width of this pulse is set on these sub-indexes.

5.14 Generic data

Clicking on the Caliper menu "***Advanced setup- Generic Data***" you can see the object 2072h.

Index	Name	Type	Map	Attributes
2072h	Generic data	ARRAY(UINT16)	No	ro

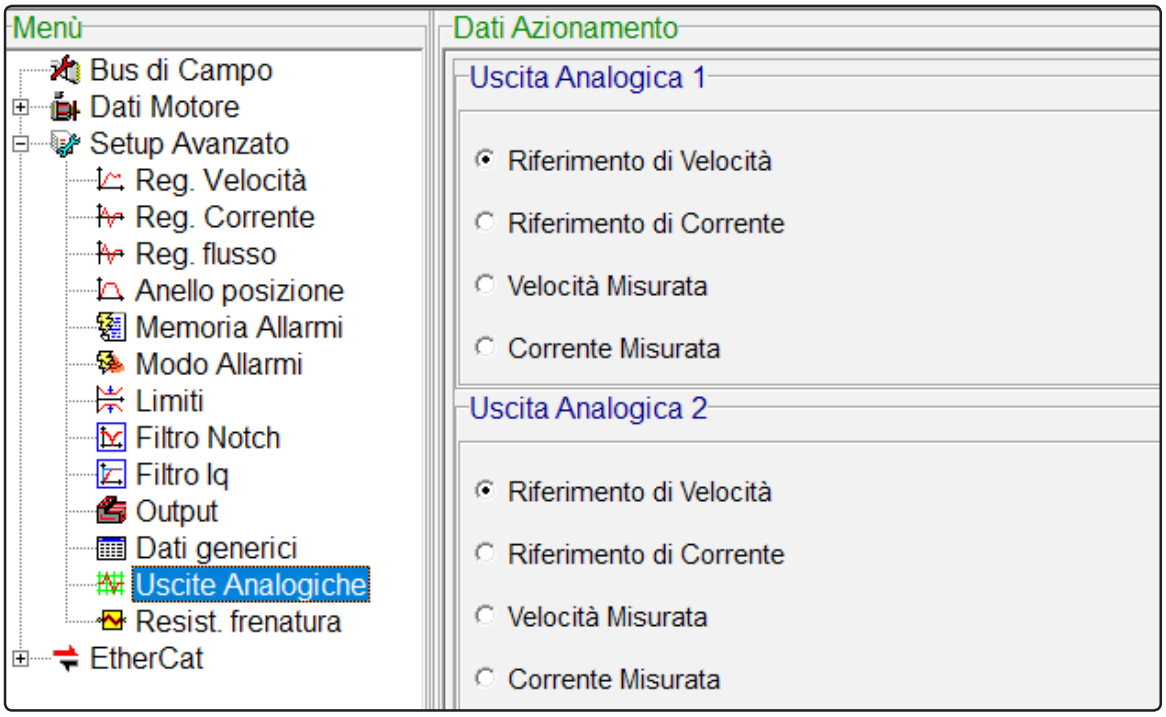


This object is stored on the non-volatile flash memory, and can only be accessed by the customer via Field Bus. The data stored on this object cannot be modified by the Drive nor be deleted if the firmware is updated. It can therefore be used to identify the Drive, by setting a unique value on it that works as an "electronic label".

PRELIMINARY

5.15 Analog outs

In this paragraph is described the menu "***Advanced setup -Analog outs***" in which the following screen appears:



From this menù you can select the variables to be sent via the Drive's analog outputs. The resolution of the outputs analog signals is 10 bits, so it's advised to use them for "monitoring" purposes only.

PRELIMINARY

5.15.1 Object 200Ch Bit 12,13,14,15 – Analog outputs

Index	Name	Type	Range	Default	Unit	Map	Attributes
200Ch	Analog outputs	UINT16	0...65536	0		No	rw

The NTT drives can produce 2 analog signals proportional to one of the variables listed in the table below, witha resolution of 10 Bit. The last 4 Bits of this object are used to select the variables that we want to observe:

Analog output 1		
Object 200Ch bits		**Observable variables**
12	**13**	
0	0	Speed reference
0	1	Current reference
1	0	Measured speed
1	1	Measured current

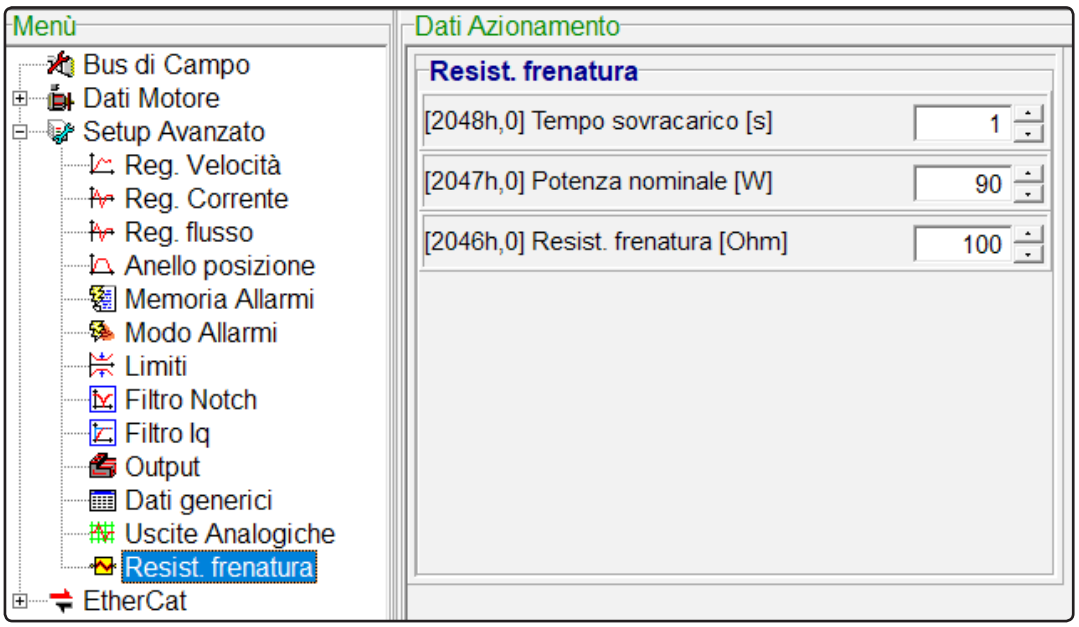
Analog output 2		
Object 200Ch bits		**Observable variables**
14	**15**	
0	0	Speed reference
0	1	Current reference
1	0	Measured speed
1	1	Measured current

5.16 Braking resistor

This paragraph describes the parameters used to set the characteristics of the braking resistor eventually connected to the Drive.

Index	Name	Type	Map	Attributes
2046h	**Braking resistor**	UINT16	No	rw
2047h	**Nominal power**	UINT16	No	rw
2048h	**Overload time**	UINT16	No	rw

These parameters can be set in the Caliper menu "***Advanced setup-Braking resistor***" in which the following screen appears:



5.16.1 Object 2046h – Braking resistor

Index	Name	Type	Range	Default	Unit	Map	Attributes
2046h	Braking resistor	UINT16	10..10000	100	ohm	No	rw

This object is used to set the resistance, expressed in [Ohm], of the Braking resistor connected to the Drive.

5.16.2 Object 2047h – Nominal power

Index	Name	Type	Range	Default	Unit	Map	Attributes
2047h	Nominal power	UINT16	30..10000	90	Watt	No	rw

This object is used to set the rated power, expressed in [Watt], of the braking resistor connected to the Drive.

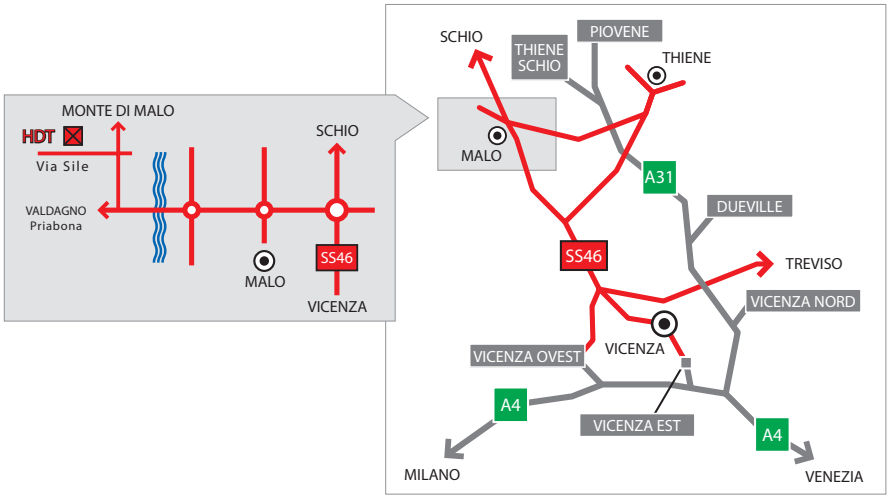
5.16.3 Object 2048h – Overload time

Index	Name	Type	Range	Default	Unit	Map	Attributes
2048h	Overload time	UINT16	1.. 255	1	s	No	rw

This object is used to set the overload time allowed by the braking resistor connected to the Drive.

PRELIMINARY

High Digital Technology



H.D.T. s.r.l.

www.hdtlovato.com

via Sile 8, 36030 Monte di Malo (VI)
Tel. +39. (0) 445.602744 r.a. – Fax +39. (0) 445.602668