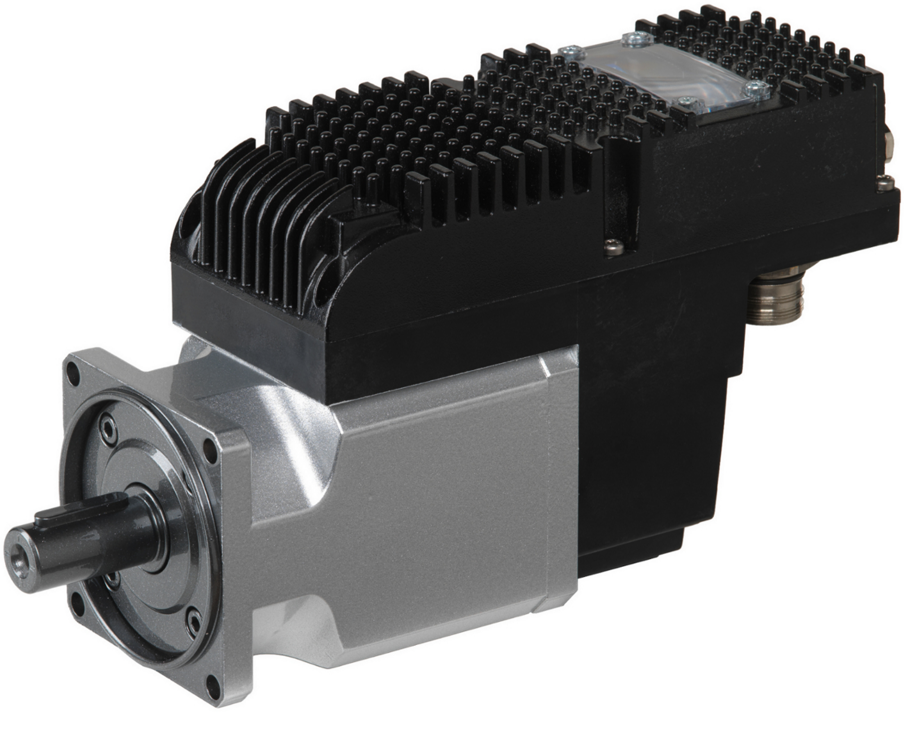
Bonfiglioli
Forever Forward

# iBMD

**Servomotor with integrated drive**



# iBMD

*Ed. 2.1 - English - 20/07/2016*

*(Translated from the original instructions)*

**IMPORTANT**

Bonfiglioli Mechatronic Research Spa reserves the right to make changes to the products described in this document at any time without notice.

This document has been prepared by Bonfiglioli Mechatronic Research Spa solely for use by its customers, guaranteeing that at the date of issue it is the most up-to-date document on the products.

Users use the document under their own responsibility and certain functions described in this document should be used with due caution to avoid danger for personnel and damage to the machines.

No other guarantee is therefore provided by Bonfiglioli Mechatronic Research Spa, in particular for any imperfections, incompleteness or operating difficulties.

# Table of Contents

# Chapter 1
# General informations on this manual

CANopen is a registered trade mark of CAN in Automation e. V.

EtherCAT is a registered trade mark and a patented technology, the licence granted by Beckhoff Automation GmbH.

Modbus is a registered trade mark of Schneider Automation Inc.

WINDOWS is a registered trade mark of Microsoft Corporation.

## 1.1. Aim

This manual is a complete guide to the installation, commissioning, functioning and use of the drives of the series *i*BMD. There are general purpose informations about the functionalities and about the drive structure, notices related to the safety for people and for the product; furthermore, for the technicians in-charge to installation, all the data and specifications to be observed for the wiring and the installation are described.

IMPORTANT

**What is written in this manual refers to the versions of firmware 31 et seq., except from any other different instructions.**

**Previous firmware versions could not implement all the functionalities described in this manual.**

The description of the different functionalities always refers to the whole series of the drives *i*BMD unless it is specified in which versions you can find the functionalities mentioned in the description.

## 1.2. Recipients

Only specialized staff can modify the drives of the series *i*BMD and use them, who previously read the manual and all the documents related to the product. Specialized staff must have been adequately trained about safety in order to prevent any possible risks. The technical training, foreground and experience of the specialized staff must help them preventing from any possible risk occurring during the product use, from the settings modification to the functioning of the mechanical, electrical and electronic equipment of the device. The specialized staff must know all the current regulations and safe working practices in case of any intervention on the product.

This manual must be read by the following staff members:

- Transport: only for personnel expert in handling sensitive parts of electrostatic charges.
- Unpacking: only for qualified electricians.
- Installation: only for qualified electricians.
- Use: only for qualified staff expert in electro-technology and activation technology.

The qualified staff must know and follow these rules:

- EN ISO 12100, EN 60364 ed EN 60664;
- national safe working practices.

This manual is addressed to all users of the drives *i*BMD with or without master controller.

⚠ WARNING

**The nonobservance of the precautions included in this document may cause risk of death, serious injuries or material damage. For a safe functioning, follow all the safety instructions in this manual. The security officer must check that the staff working with the drives read and understood this manual before using them.**

## 1.3. Responsibilities

⚠ ATTENTION

**Bonfiglioli Mechatronic Research S.p.a can modify the described products in this manual in any time and without any notice.**

**This manual was written by Bonfiglioli Mechatronic Research S.p.a only for their customers' use providing the most updated version of the products.**

**The responsibility to use this manual belongs to every user and the use of some functions must be under strict care to avoid any danger for the staff and the equipment.**

**No other warranty is provided by Bonfiglioli Mechatronic Research S.p.a, in particular for possible imperfections, incompleteness, and/or any other difficulties.**

# 1.4. Abbreviations

| Abbreviation | Meaning |
|---|---|
| 0x | Number in hexadecimal notation |
| +HV | DC bus voltage, supply voltage of the power section |
| ABS_E | Absolute encoder |
| A | Motor phase U |
| AI | Analog input |
| ac | Alternating current |
| B | Motor phase V |
| BDM | Base drive module |
| C | Motor phase W |
| CAN | Controller Area Network, filed bus |
| CDM | Complete drive module |
| CE | Communité Européenne |
| CH A | Channel A of a torque of quadrature signals |
| CH B | Channel B of a torque of quadrature signals |
| COB-ID | Communication object identifier |
| CoE | CANopen over EtherCAT |
| COM | Serial communication interface for personal computer |
| CRC | Cyclic redundancy control |
| D | Phase signal with the magnetic field of the motor rotor |
| D. Fan | Digital output starting the drive external cooling fan |
| dc | Direct current |
| Dir | Direction |
| Drv OK | Digital output with drive function OK |
| EEPROM | Electrically erasable programmable memory (permanent memory) |
| EMC | Electromagnetic compatibility |
| EMCY | Emergency: protocol object CANopen and EtherCAT to notify any errors |
| ESC | EtherCAT Slave Controller (ET1100 component) |
| ETC | EtherCAT, Ethernet for Control Automation Technology, field bus |
| FA | Phase A of a torque of quadrature signals |
| FB | Phase B of a torque of quadrature signals |
| FC + / FC POS | Positive limit switch or positive limit of hardware position |
| FC - / FC NEG | Negative limit switch or negative hardware position |
| FC + sw | Positive limit position programmable via software |
| FC - sw | Negative limit position programmable via software |
| FoE | File access over EtherCAT |
| FW | Firmware |
| GND | Ground |
| HOME | Digital input of Home |
| HW | Hardware |
| I | Input, generally digital |
| I2C | Inter-Integrated Circuit, two-wire serial communication system used among integrated circuits |

| Abbreviation | Meaning |
|---|---|
| I2T | Passing specific over energy |
| *i*BMD | Integrated brushless drive |
| ID | Identifier |
| LED | Light-emitting diode |
| LSB | Byte (or bit) less important |
| MB | Mega Byte |
| M. Fan | Digital output starting the external fan to cool down the motor |
| MDB | Modbus, serial communication protocol Modbus on RS232 |
| MSB | Byte (or bit) less important |
| neg | Negative |
| NC | Not connected |
| NMT | Network management: protocol object CANopen to manage the network |
| O | Output, generally digital |
| OSC | Integrated oscilloscope |
| OSI | Open system interconnection |
| PC | Personal computer |
| PDO | Process data object: object of the CANopen and EtherCAT protocols to read and write the cyclic data (mappable parameters on the PDOs) |
| PDS | Power Drive System |
| PE | Protection Earth, protection conductor |
| PLC | Programmable logic control, drive integrated programmability |
| PLL | Phase locked loop |
| pos | Positive |
| Q | Quadrature signal with the magnetic field of the motor rotor |
| PWM | Pulse-width modulation |
| Pwm O | Output PWM |
| RAM | Random access memory (non permanent memory) |
| RES | Resolver |
| RMS | Root Mean Square, effective value |
| RTR | Remote transmission request |
| RX | Reception |
| SDO | Service data object: object of the CANopen andEtherCAT protocol for the reading and writing of the drive parameters |
| S1 | Continuous service, functioning condition used to reach the thermal value |
| a.s.l. | Above sea level |
| SYNC | Synchronization: protocol object CANopen to synchronize the network nodes |
| SM | Sync manager (SM) |
| /STO | Safe Torque Off (this is an active-low logic signal) |
| SW | Software |
| TBD | To be defined |
| Temp | Temperature |
| $T_{SYNC}$ | Period in which the process data (PDO) are exchanged |
| TX | Transmission |
| U | Motor phase U |

| Abbreviation | Meaning |
|---|---|
| V | Motor phase V |
| W | Motor phase W |

# 1.5. Symbols

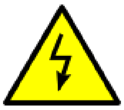
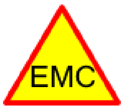
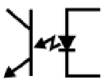
| Symbol | Description |
|---|---|
| ⚠ DANGER | It shows a surely dangerous situation, in case of failure to comply with safety rules it can lead to a serious or fatal accident |
| ⚠ WARNING | It shows a probably dangerous situation, in case of failure to comply with safety rules it can lead to a serious or fatal accident or damage to the equipment. |
| ⚠ ATTENTION | It shows a probably dangerous situation, in case of failure to comply with safety rules it can lead to a serious accident or damage to the equipment. |
| ⚠ NOTICE | It shows a potentially dangerous situation, in case of failure to comply with safety rules it can lead to a serious accident or damage to the equipment. |
|  | It shows the presence of dangerous voltages that can cause electrical shocks. |
|  | It shows the problems related to electromagnetic compatibility. |
|  | It shows the presence of surfaces and/or heat sources that can cause burns. |

***Table 1.1. Danger classes***

| Symbol | Description |
|---|---|
|  | It shows some information to pay attention to. Please pay attention to what has been shown. |
| IMPORTANT | It shows some important information on the mentioned topic. |
| NOTE | It shows some important information on the text about the mentioned topic. |
| ADVICE | It shows some useful information on the mentioned topic. |

***Table 1.2. Information classification***

| Symbol | Description |
|---|---|
|  | Optical coupler |

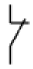
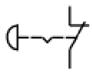
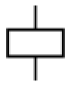
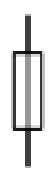
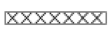
| Symbol | Description |
|---|---|
|  | Make contact |
|  | Break contact |
|  | Break command with emergency shutdown, started by a "mushroom-headed" button and at a certain position |
|  | Control coil |
|  | Lamp |
|  | Fuse |
|  | Ground protection |
|  | Screen |

***Table 1.3. Electrical symbols.***

## 1.6. Definitions

**BDM Base drive module**
Activation module made up by a conversion section and a section used to check the speed, the torque, the current or the voltage, etc. In this manual the BDW is called **power section**.

**CAN 2.0**
Standard that describes the *data link layer* and the *physical layer* (ISO/OSI Reference Model) of the CAN bus.

**CANopen over EtherCAT**
Protocol on EtherCAT bus that allows the access to the drive parameters through SDO CANopen.

**CDM Complete drive module**
Activation without motor and sensors mechanically coupled to the drive shaft, made up by a BDM, but not limited to it, and other devices, such as the charge section and the auxiliaries.

**CiA-301**
Detailed note on communication protocols and objects to manage the network CANopen (*Communication Profile* DS301).

**CiA-402**
Detailed note to define the rules for a standardized behaviour of the drives connected to a fieldbus. The drive management according to this specification is described in Section 8.5, *CiA402 state machine*. In case of a CANopen network, the specification refers to the *Device profile for drives and motion control* DSP402 v.3.0, the profile is constructed basing on the CiA-301. In case of an EtherCAT network, the specificateion refers to the *Implementation Directive for CiA-402Drive profile* ETG.6010.

**Distributed clocks**
Mechanism used to synchronize the masters and the slaves in the network EtherCAT (functionality implemented in the chip ET1100).

**Drive**
See PDS Power drive system definition

**Drive disable**
States of the CiA402 State Machine (see Section 8.5, *CiA402 state machine*) in which the torque is not applied to the motor and the motor control loops aren't active.

**Drive enable**
States of the CiA402 State Machine (see Section 8.5, *CiA402 state machine*) in which the torque is applied to the motor and the motor control loops are active.

**File access over EtherCAT**
Protocol on bus EtherCAT used to update the drive directly from the files.

**Index**
Encoder zero mark.

**Master**
Node taking control of the communication bus and starting first to interact with the other connected nodes.

**Node**
Hardware device (drive, sensors, actuators) connected to the communication bus which can communicate with the other devices.

**Operation disable**
States of the CiA402 State Machine (see Section 8.5, *CiA402 state machine*) where it is not possible to command the motor motion. The drive can be enabled or disabled.

**Operation enable**
State of the CiA402 State Machine (see Section 8.5, *CiA402 state machine*) where it is possible to command the motor motion.

**Integrated PDS**
PDS where the motor and the CDM/BDM are mechanically integrated in a single unit.

**PDS Power drive system**
System used to control the speed of an electric motor, including the CDM and the motor, but not the already started equipment. In this manual the PDS is simply called **drive**.

**Network protocol**
All rules, mechanisms and formalities that two or more electronic devices connected one another must respect to start a communication.

**Real-time**
Drive command mode used to control the motor motion continuously and in pre-arranged time.

**Modbus register**
Memory area of 1 Word = 16 bit = 2 byte that contains a numeric value, accessible both in reading and in writing. It's identified by a number that represents its memory position and it's used to exchange data in the Modbus protocol.

**Functional safety**
Part of the safety of the machine and the machine control system which depends on the correct functioning of the Safety system, other technology safety-related systems and external risk reduction facilities.

**Safety system**
(Even named SRECS) Machine electrical control system whose failure can result in an immediate increase of the risk(s).

**Sync manager (SM)**
Functionality of the chip ET1100 (contained in the drives version ETC) used to exchange data in a safe and sound way between the master and the drives EtherCAT. For any further information please look EtherCAT Technology Group (ETG).

**Sync Signal**
Hardware signal generated by ET1100 and managing the Distributed clocks.

**Discrepancy time (maximum)**
(Maximum) time interval during which a difference between the signal logic level is allowed.

**Transition**
Intermediate phase that allows the transition from a state to another one of the CiA402 State Machine (see Section 8.5, *CiA402 state machine*).

## 1.7. Reviews

| Revision History | | |
|---|---|---|
| Revision 2.1 | 20/07/2016 | Author: Bonfiglioli Mechatronic Research S.p.a |
| • First manual revision. | | |

# Chapter 2
# Information on the *i*BMD drives

The drives *i*BMD are digital drives for three-phase sinusoidal brushless motors with permanent magnets. In particular, the *i*BMD drive is composed by a brushless motor, a feedback position sensor, static brake (optional), interface to the field buses, power section and control section. All versions of this drive type have digital I/O, analog input, leds and dip switches. There is also a permanent memory and an auxiliary serial port in which the protocol Modbus has been implemented.

The software on the drives of the series *i*BMD is divided into two types:

- *Boot firmware*: it boots the drive by enabling some basic services and, after an initial phase of identification and diagnostics of the system, it runs the firmware
- *Firmware*: it manages all the drive operating functions

The firmware provides some different working operating modes that can be divided into three classes:

- *Position modes*: the drive receives a position reference and follows the motion in order to minimize the error between the reference value and the current position.
- *Speed modes*: the drive receives a speed reference and runs the motion in order to minimize the error between the reference value and the current speed.
- *Torque modes* : the drive receives a torque reference and runs the motion in order to minimize the error between the torque reference and the current found in the phases.

NOTE

**The drive mission time is 20 years.**

NOTE

**For further details on the features and options of the available versions, see Chapter 5, *Technical features* and Chapter 29, *Order codes*.**

## 2.1. Laws and standards

The products of the *i*BMD series are in accordance with the following specifications:

- 2004/108/CE relating to electromagnetic compatibility;
- 2006/95/CE relating to electrical equipment designed for use within certain voltage limits;
- 2006/42/CE relating on machinery;

in conditions in which the installation may be considered as typical (then the instructions in the user manual are respected and there are not particular work environment or installation needs).
Bonfiglioli Mechatronic Research S.p.a guarantees the conformity of the drive to the harmonized standards:

| | |
|---|---|
| EN 61800-5-1:2007 | Adjustable speed electrical power drive systems<br>Part 5-1: Safety requirements - Electrical, thermal and energy |
| EN 61800-3:2004<br>EN 61800-3/A1:2012 | Adjustable speed electrical power drive systems<br>Part 3: EMC requirements and specific test methods |
| EN 61000-6-2:2005/<br>AC:2005 | Electromagnetic Compatibility (EMC)<br>Part 6-2: Generic Standards - Immunity for industrial environments |
| EN 55011:2009<br>EN 55011/A1:2010 | Limits and methods of measurement of radio disturbance characteristics of industrial, scientific and medical (ISM) radio frequency equipment |
| EN 60204-1:2006<br>EN 60204-1/A1:2009 | Safety of machinery - Electrical equipment of machines<br>Part 1: General requirements |
| EN 61800-5-2:2007 | Adjustable speed electrical power drive systems<br>Part 5-2: Safety requirements - Functional safety |
| EN ISO 13849-1:2008 | Safety of machinery - Safety-related parts of control systems<br>Parte 1: General principles for design |

The products of the *i*BMD series are commercialized as components of a Power Drive System, belong to the restricted distribution category and are intended to the installation in industrial environment. If used in domestic environment it may be necessary to take further emission measures and appropriate precautions.

The installation of these devices is intended to specialized personnel that has an in-depth knowledge about the safety requirements and the electromagnetic compatibility (EMC).

The planner has the responsibility to guarantee that the product or the final system comply to the pertinent regulations that are in force in the country in which the product (or the entire system) is used.

If the entire system is connected to a low voltage distribution public network it will be necessary to pay attention to the network harmonic and flicker inclusion effects to guarantee the overall certification.

## 2.2. General features of the *i*BMD drives

| Features | *i*BMD |
|---|---|
| Motor size | See the table at the beginning of the Chapter 5, *Technical features* |
| Range of supply of the power section | See the table at the beginning of the Chapter 5, *Technical features* |
| Range of supply of the control section | See the table at the beginning of the Chapter 5, *Technical features* |
| Feedback sensor | Hyperface absolute Encoder |
| Main communication port (field bus) | CANopen or EtherCAT with CiA-402 device profile |
| Auxiliary communication port | Modbus on RS232 |
| Multifunction port | N. 3 differential bidirectional lines (RS485 compatible) for presettable default functions (master encoder input, pulse-dir#, others) |
| Rotary dip switches | Setting the node number and/or communication speed of the main bus |
| Leds | Information and local diagnostics through transparent window |
| Number of digital bidirectional I/O | 4 |
| Number of digital inputs | 6 |
| Number of digital outputs | 3 |
| Number of analog inputs | 1 |
| Functional safety | See Chapter 6, *STO safety function: Safe Torque Off* |
| Electrical insulation | Adequate distances of electric insulation both in the surface and in the air are guaranteed according to the EN61800-5-1 regulation, between the input voltage of the DC bus/motor connections and the signal and communication electronics of the control section (refer to Chapter 5, *Technical features*). |
| Protections | • DC bus overvoltage<br>• short circuit and/or overcurrent on the motor phases<br>• power section overtemperature<br>• control section circuits overtemperature<br>• Motor winding overtemperature<br>• motor energy overload, through $I^2T$<br><br>(please refer to Chapter 24, *Fault and Warning* and to Section 14.6, *I2T*). |
| Permanent memory | *YES* |
| Integrated PLC | *YES* |

***Table 2.1. Features of the i*BMD *drives***

### 2.2.1. Equipment of *i*BMD series drives

The drive package includes:

- the *i*BMD drive (in the CAN or ETC version)
- plastic cap for the M8 connector
- “dust cover” plastic cap for the M23 I/O connector
- package leaflet

NOTE

**No flying connector or cable is included in the standard equipment.**

Before to begin to work with the drive, verify that there are not visible damages. Be sure that the *i*BMD drive you have taken from the package is the correct model for your application, that it corresponds to what you have ordered and that you can provide a voltage supply as prescribed for the system.

### 2.2.2. Block diagram of *i*BMD drives

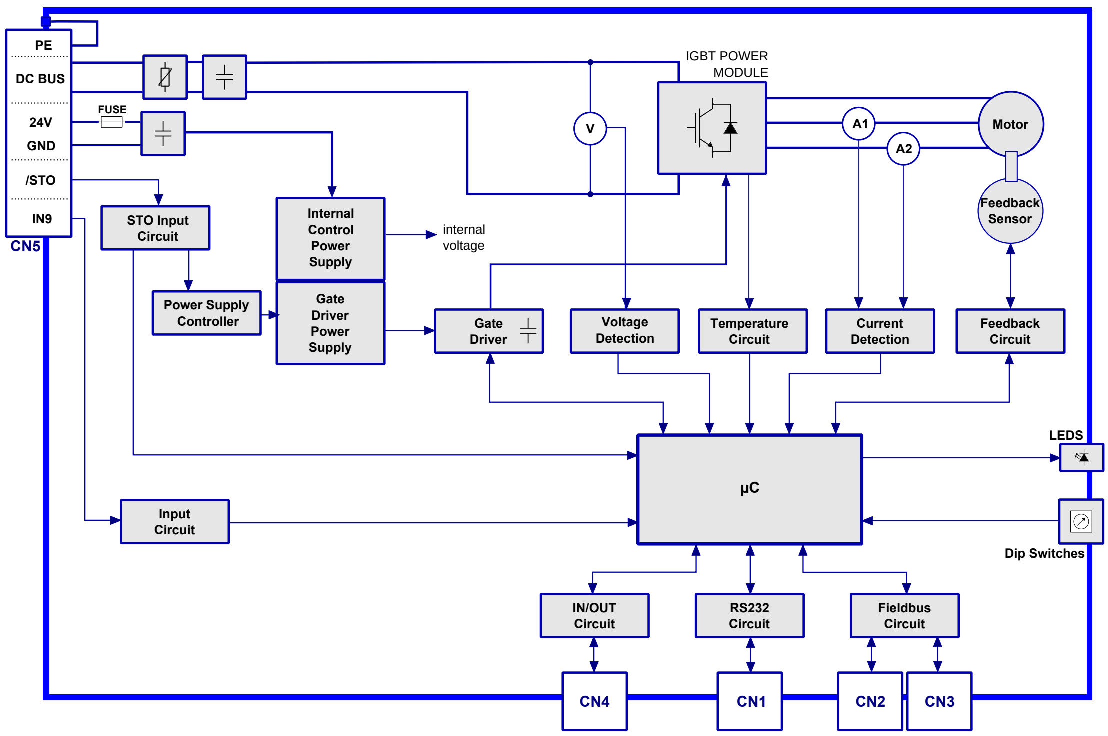



*Figure 2.1. Block diagram of i*BMD *drives*

### 2.2.3. Identificative plate

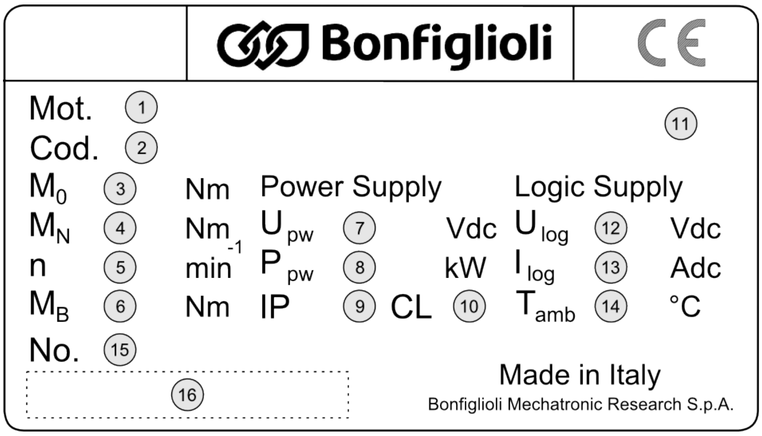



*Figure 2.2. Product plate example.*

| Reference | Meaning |
|---|---|
| 1 | Product designation |
| 2 | Product code |
| 3 | Stall torque |
| 4 | Nominal torque |
| 5 | Rated speed |
| 6 | Brake nominal torque |
| 7 | Range of input voltage of the power section |
| 8 | Nominal power[a] of the power section input |
| 9 | Protection degree |
| 10 | Motor class of insulation |
| 11 | Bus communication protocol mark |
| 12 | Control section voltage range |
| 13 | Maximum current of the control section |
| 14 | Working ambient temperature for a functioning compliant to the technical data |
| 15 | Serial number |
| 16 | Serial number in bar code format |

[a]referring to the torque at the rated speed

*Table 2.2. Plate fields*

## 2.3. Safety precautions and limits

⚠ WARNING

**The precautions described in this paragraph are perfect to avoid any dangerous situation by suggesting the right use of the product. Only qualified staff who read and understood all the documents on this product can use it. The specialized staff must follow a safety training in order to know which individual protective equipments to be taken and to avoid any risks related to the product use (included any changes in the parameters) and to find a possible solution.**

⚠ DANGER

**The drive must not be used in an explosive or corrosive environment, in the presence of inflammables, water or fuels. There can be risk of fire, electrical shock or injuries.**

**In case of failures because of accidental circumstances or wiring errors the power section can even cause electric arcs. The drive must be installed in an environment without any inflammables. It is particularly forbidden to use it in the presence of inflammable gases or vapours.**

⚠ NOTICE

**The drives can be used/installed outdoor, but can't be directly exposed to the sunlight (UV rays).**

⚠ DANGER

**Do not transport, install or make any connections or inspections when the drive is charged. In such cases switch the power off, wait for some seconds until the voltage is lower than 50 volts, otherwise there can be risks of electrical shock or damaging the drive.**

**The connector used to connect the power supply may have a high voltage. Do not touch these devices when the drive is supplied, even if it is disabled.**

⚠ DANGER

**The usage of this product implies the presence of a voltage greater than 50V, therefore there is life-threatening and a risk of electric shock and serious injury. Follow the general and safety regulations when you are working on the power related installations.**

**Do not connect the power connector (power supply and/or motor) when the drive is powered. There can be electrical arcs that can damage the connector and the drive and cause a fire.**

⚠ WARNING

**The protection from surge must be delegated to a device that's external from the *i*BMD drives, after an accurate risk analysis made by the integrator of the machine.**

⚠ WARNING

**Keep the drive power supply within the specified ranges in order to avoid any risks of fire, electrical shocks and damaging the drive. In the same way connect the cables in a safe way by respecting the connections.**

⚠ DANGER

**Do not touch the connection devices when the drive is in voltage. In case of maintenance be aware that the voltage in the power connectors is lower enough to not cause an electrical shock.**

⚠ DANGER

**Do not touch the drive or the motor during functioning or immediately after its disabling: the surface temperature can be higher than 80°C.**

**To prevent any risks of damaging the drive do not obstruct or limit its ventilation.**

⚠ WARNING

**Do not open and do not modify the system: for any internal checks please contact Bonfiglioli Mechatronic Research S.p.a. In case of forcing the system the warranty expires.**

⚠ ATTENTION

**Please do not short-circuit any signals from the power connector with the drive case or logic signals (for example coming from the connectors for the field bus).**

**It's recommended a control section supply wiring separated by the power supply one, in order to avoid malfunctioning and to limit the control logic signals noises.**

**The cables section for the power stage must be adequate to the drive power.**

⚠ WARNING

**The section of the power supply conductors must be adequate to the drive power and not lower than 1.5 mm$^2$.**

**Always connect the protection ground and the functional mass with two separate cables.**

⚠ DANGER

**In case of simultaneous failure of two power semiconductors (IGBT) of the power section, it is possible that the motor turns by 180°/number of pole pairs.**

⚠ DANGER

**Don't apply an excessive force on the motor shaft, in order to avoid the damage of the bearings.**

**If the key is installed on the motor shaft, be sure that it is fixed to the machinery so that the key cannot slip off from its seat.**

⚠ NOTICE

**Insofar as their purpose allows, accessible parts of the drive have no sharp edges and no rough surfaces likely to cause injury. In case the key is removed, the sharp edges of the key seat are accessible to the user, unless the shaft plastic cover is installed: beware the injury danger in case of removing the plastic cover or the shaft key.**

⚠ NOTICE

**The drive has dip switches to set the node number and/or communication speed of the main bus. All this settings must be made when the drive is switched off. To prevent damages to the drive it's recommended to pay particular attention when working on this settings because in**

**the drive there are some components that are sensitive to the electrostatic discharge. It's in particular advisable to preventively discharge the static electricity, to place the drive on a conductive support and to avoid contact with highly insulating materials. BEFORE TO POWER THE SYSTEM, REMEMBER TO FASTEN THE TRANSPARENT COVER (if it has been removed).**

⚠ NOTICE

**When some Fault is found, the drive automatically disables and a led signal shows the possible cause: the motor is no longer in torque and it can move to another position and may damage the devices and/or the surrounding people. It must be made a evaluation of the risk about the particular machine in which the product is used. In consequence the user must take appropriate measures to avoid risks to the safety of the person.**

⚠ NOTICE

**When there is a Fault, the drive is disabled; before enabling it again by rebooting the system or by some correct commands through the field bus, remove the cause generating the Fault.**

⚠ WARNING

**The magnetic and electromagnetic fields, that are generated by the conductors in which the current flows or by permanent magnets inside the electric motors, represent a serious danger for the people with the pacemaker, metallic prostheses and hearing aids. Be sure that these people have no access to the areas in which these systems are presents (both during functioning and in storage). Eventually, if these persons have to enter in the described areas, consult a doctor.**

⚠ WARNING

**The device builder using the drives *i*BMD must analyse the risk for the device and implement the necessary measures to safeguard the device itself and the surrounding people from any unforeseen motions.**

⚠ NOTICE

**The drive has been designed and constructed so that risks resulting from the emission of airborne noise are reduced to the lowest level. The airborne noise emission and the related risks for the user are in any case depending on the application and must be analyzed by the machine designer.**

⚠ ATTENTION

**The drive has been designed and constructed to limit the build-up of potentially dangerous electrostatic charges and is provided of a discharging system: the chassis is made of metal and polimeric material. The metallic part is protected by the grounding system. The non-metallic materials may build-up electrostatic charges in case of contact with other insulating materials.**

⚠ DANGER

**This product is intended to be exclusively used in machines and systems in industrial environment, respecting the described application, environmental and functioning conditions.**

**Follow the safety regulations and the ordinances of the country in which the product (or the relative control and command system) is used.**

**It is recommended not to use the product for any further purpose than those specified in this manual.**

# Chapter 3
# Main features of MotionDrive

MotionDrive is a programme for *personal computer* used to control, configure and programme in a simple, quick and perceptive way the drives of the *i*BMD series.

From the tab Main of MotionDrive it is possible to know the whole drive status. For example: the detailed description of the found errors, the status of the outputs and the digital and analog inputs, position and speed of the shaft, drive operative status, connection status, etc... From MotionDrive it is possible to export the drive parameters in a text file to clone more drives in the same mode by exporting the parameters from one drive to another.

MotionDrive offers a window for programming, diagnostics and control of the internal PLC.

IMPORTANT

**What is written in this manual refers to the MotionDrive 3.0.1.223 versions et seq. Previous versions of MotionDrive could not implement all the functionalities described here.**

## 3.1. Requirements and compatibility

Minimum PC requirements:

- System with compatible processor Pentium 133 MHz or higher.
- Sufficient memory for the operating system, minimum 128 MB, recommended 512.
- Hard disk with minimum available space to install the programme, at least 35MB.

- Display adapter and monitor Super VGA, minimum resolution 800 x 600 px, better 1024x768 px or higher.

Compatibility with the following operating systems:

- Microsoft Windows XP
- Microsoft Windows Vista, 7, 8 or latest versions, 32bit and 64bit.

## 3.2. Installation

Check if all the system prerequisites are respected (Section 3.1, *Requirements and compatibility*).

**Installation from the Bonfiglioli website**

- Connect to the *http://www.bonfiglioli.com* website and enter the *Products & Solutions* through the *i*BMD drop-down menu.
- Enter in the pages dedicated to the *Business Unit Industrial* and, from here, enter the section that's dedicated to the products and download the MotionDrive.msi file.
- Run the *MotionDrive.msi* downloaded file by following the proposed installation procedure.
- Every MotionDrive version is released with the most updated firmware, in relation to the release date.

**Installation from file**

- If the *MotionDrive.msi* file is already on the PC, run the file and follow the proposed procedure, otherwise contact Bonfiglioli Mechatronic Research S.p.a in order to request the file.
- Every MotionDrive version is released with the most updated firmwares and motors database, in relation to the release date.

ADVICE

**Please accept all the configurations proposed during the installation.**

## 3.3. MotionDrive overview

All the functionalities of MotionDrive can be accessible through the three tabs (Bus, Main), the menus or the toolbars.

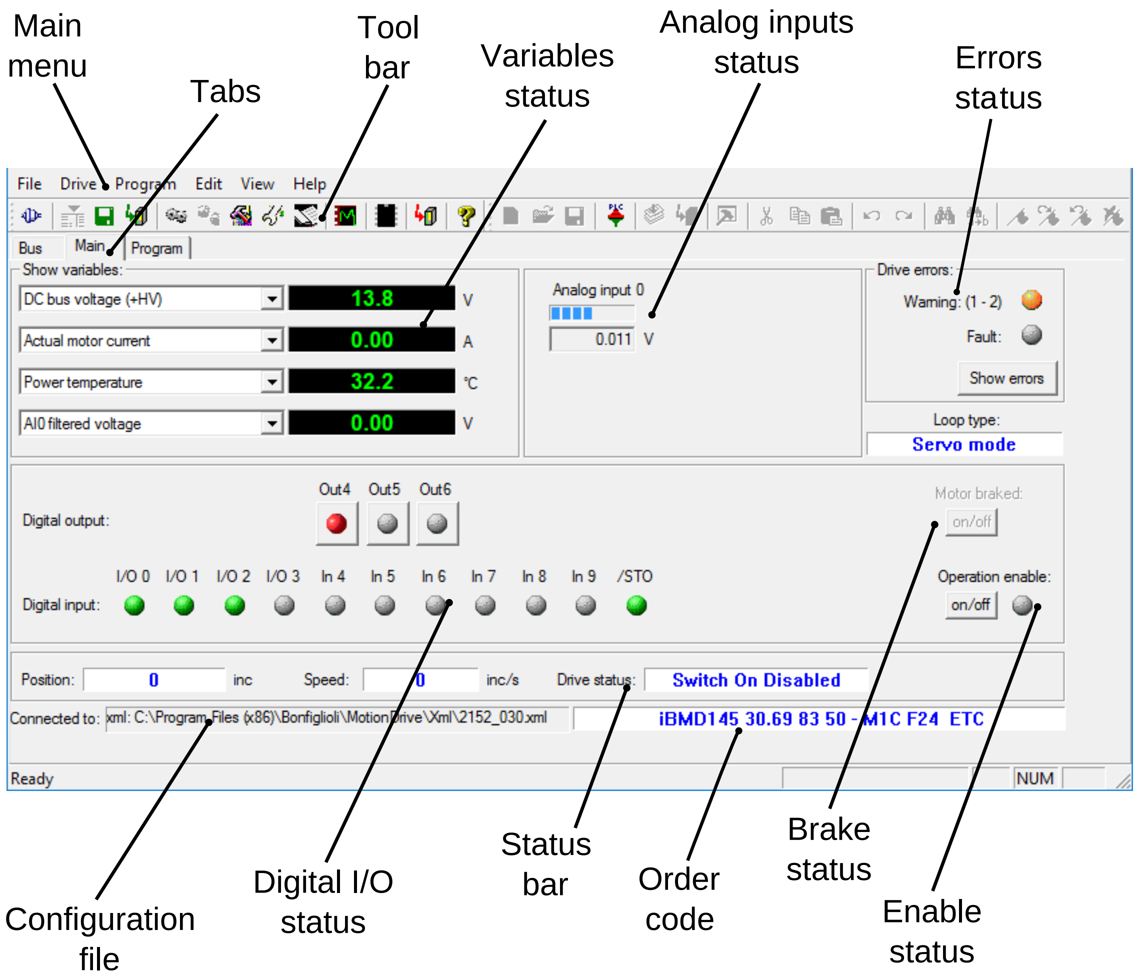



*Figure 3.1. Tab Main of MotionDrive*

| Tab | Functionalities | Link |
|---|---|---|
| Main | Variables status | Section 23.1, *Parameters monitoring* |
| | Analog inputs status | Chapter 17, *Analog input* |
| | Errors status | Chapter 24, *Fault and Warning* |
| | Configuration File | Section 26.5, *Updating the Configuration File* |
| | Digital I/O status | PhysicalOutputs and DigitalInputs |
| | ManufacturerDeviceName | Section 29.1, *OrderCode* |
| | Enable status | Operation enable |
| Bus | Configure the Main bus | Chapter 8, *Communicating with the drive* |
| Program | Programme the internal PLC | Chapter under construction. |

***Table 3.1. Functionalities of the two main tabs***

# Chapter 4
# Quick start for drives *i*BMD

For a quick test installation of the drives *i***BMD**, follow what is reported in this chapter.

## 4.1. Before starting

### Safety precautions

⚠ WARNING

**Before installing the drive, read the paragraph on safety Section 2.3, *Safety precautions and limits*. If you do not follow the safety instructions you may damage the equipment or be hurt.**

### Requested instruments, materials and equipment

- Supply system to supply the control and power section;
- Supplying cable to connect to the connector CN5;
- Serial cable to connect to the connector CN1;
- Screwdrivers to tighten the supply conductor according to the suitable wiring;
- PC with serial port RS232 and with the requisites in Section 3.1, *Requirements and compatibility*.

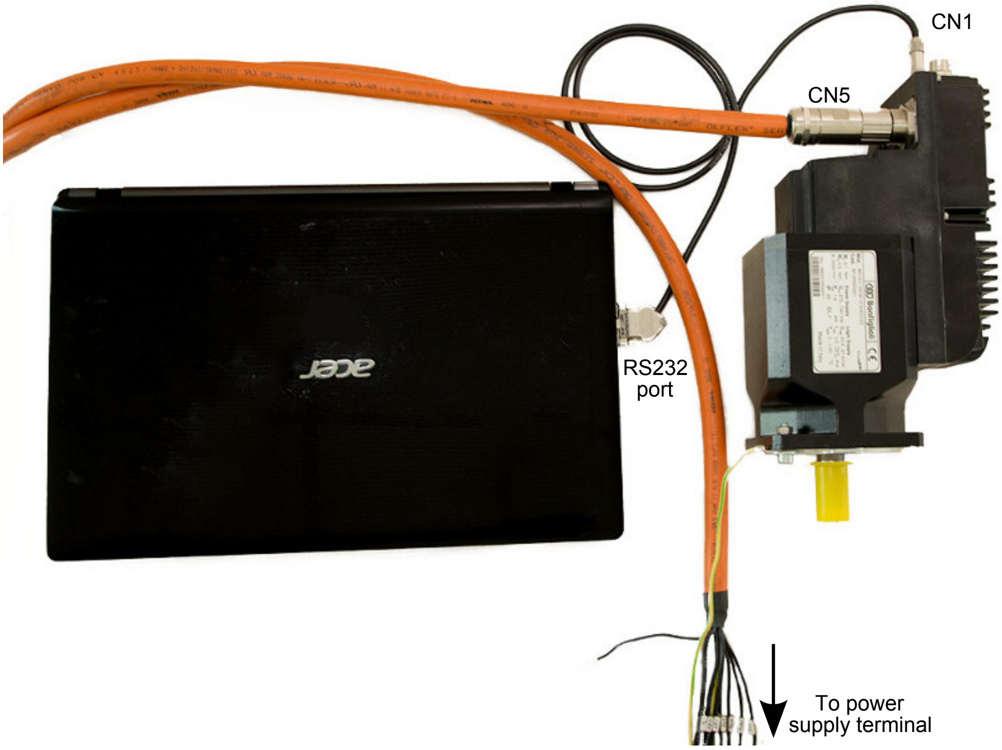



***Figure 4.1. Minimum setup for the quick start.***

## 4.2. Hardware installation

**1. Mechanical installation**
For the system installation use the 4 holes on the motor anterior flange. The dimensions are reported on the Section 5.2, *iBMD dimensions and sizes*. Be sure that the drive and the motor ventilation is free, respecting however the maximum admitted environment temperature (see Chapter 5, *Technical features*).

**2. Connection of the protection conductors**
Connect the PE protection conductor to the motor flange as shown in the Figure 4.2. For detailed informations about the connection to the protection conductors see Section 7.2.1, *System Supply*.

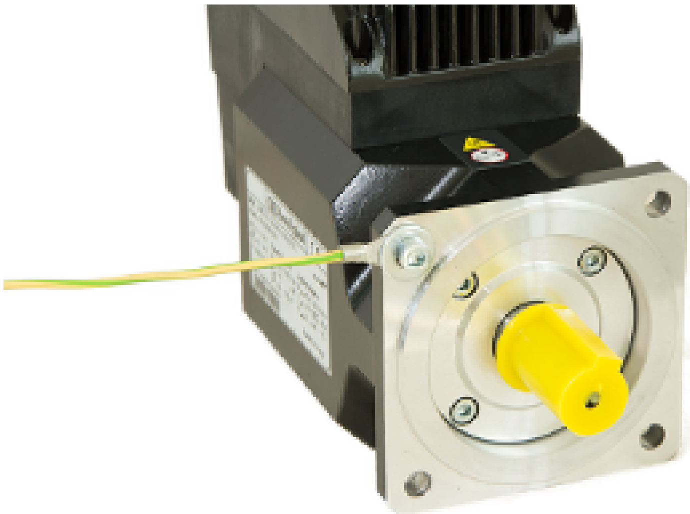


*Figure 4.2. Connection of the PE protection conductor to the flange*

### 3. Connection of supplies and /STO

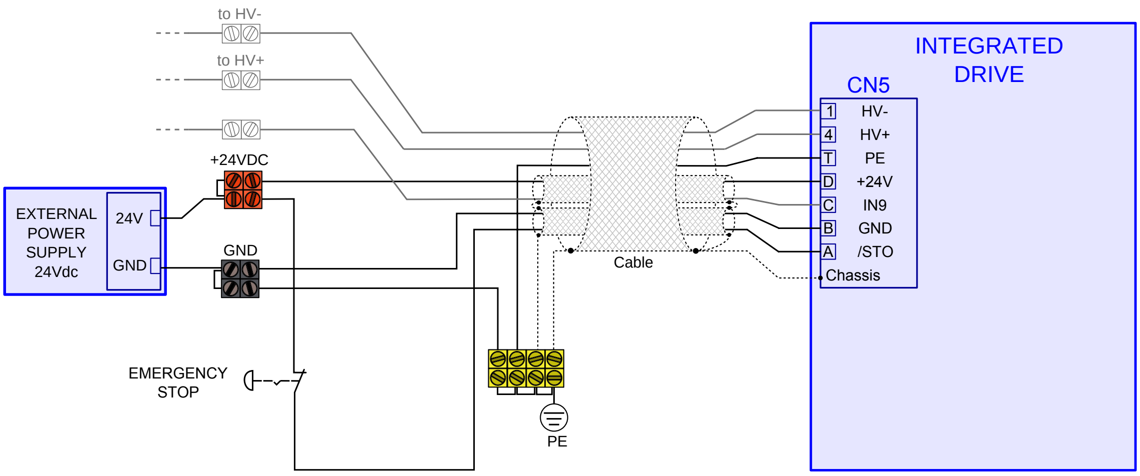
ADVICE

**For further details please refer to Chapter 6, *STO safety function: Safe Torque Off*.**

Connect the supplies and the /STO input according to what's reported in the following scheme.

DANGER

**Before proceeding, check if the power supplies are switched off and there is no more voltage in the connection terminal boards.**




To connect the pins of CN5, please pay attention to what is shown in the following table:

| PIN | Signal | Description |
|---|---|---|
| 1 | HV - | DC Power supply (negative pole) |
| 3 | - | Not connected |
| 4 | HV + | DC Power supply (positive pole) |
| T | PE | Protection Earth |
| A | /STO | Safe Torque Off Input (this is an active-low logic signal) |
| B | GND | Ground Control supply |
| C | IN9 | Digital Input 9 |
| D | +24 V | +24Vdc Control supply |
| Chassis | | |

The STO integrated function is implemented in the drives according to the EN 61800-5-2 norms and is used to execute a stop of 0 cathegory according to the EN 60204-1 norms.

⚠ WARNING

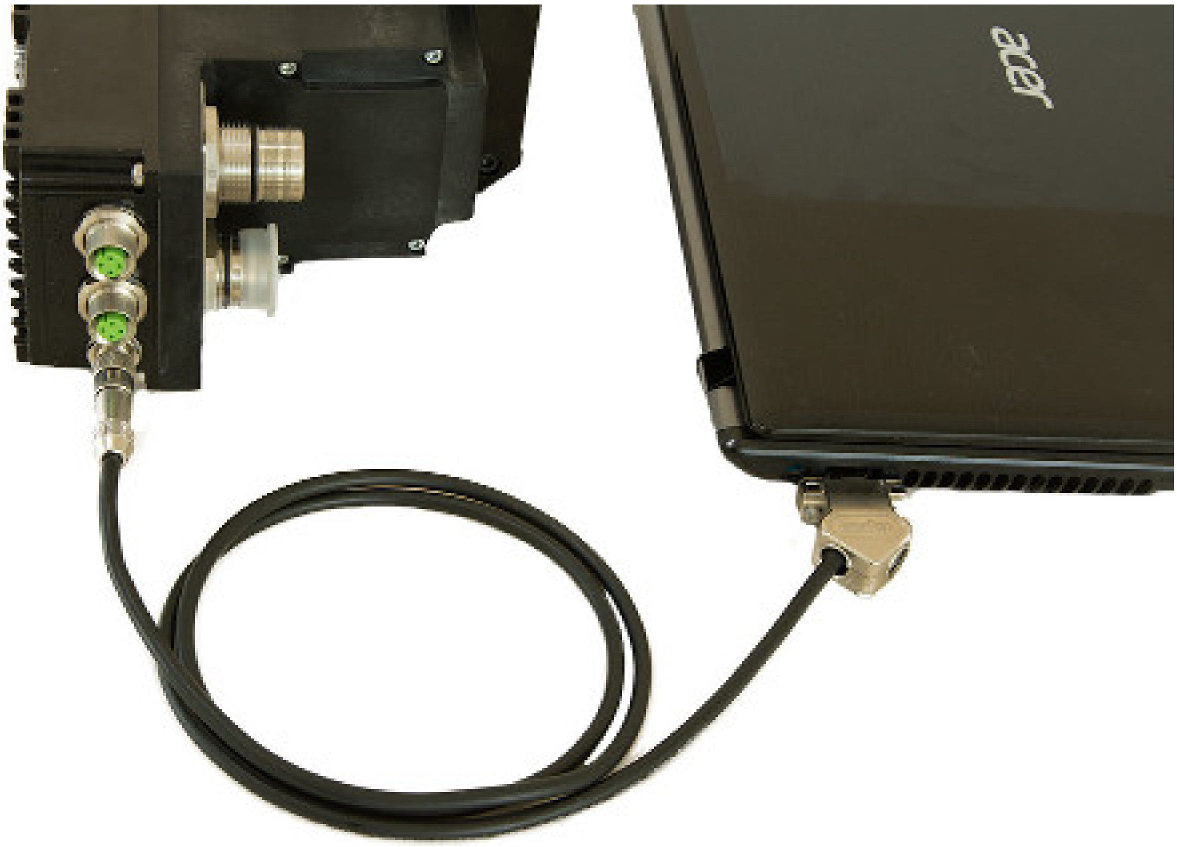
**When the voltage of the digital input associated to the STO function is cut off, the motor torque is disabled in a safe way, the drive power section is disabled without cutting the voltage to the DC bus and it is not possible to control the motor motion. It is advisable to always stop the motor before disabling the input /STO.**

## 4. Connection of the serial port

Connect the serial port RS232 to the connector CN1 of the *i*BMD drive.

⚠ WARNING

**Connect and disconnect the communication connectors only when the drive is switched off. Check if the Ground Control Supply pin of CN5, the drive and the PC are correctly connected to the protection conductor.**




To connect the pins of CN1, please pay attention to what is shown in the following table:

| PIN | Signal | Description |
|---|---|---|
| 1 | TX232 | Transmit Data RS232 |
| 2 | RX232 | Receive Data RS232 |
| 3 | NC | Reserved, not connected |
| 4 | GND_COM | Ground RS232 |
| Chassis | PE | Protection Earth |

**5. Confirmation of the connections**

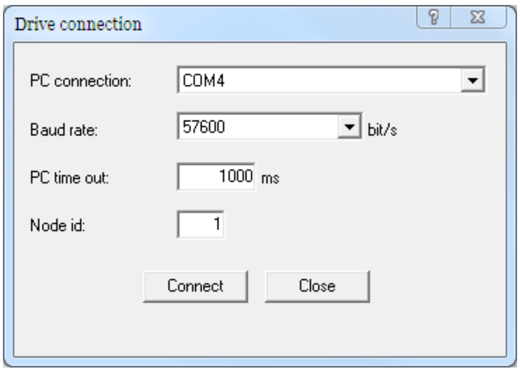
After having completed the connections, check if they are correctly connected and switch on the power supply of the control section (24Vdc). The leds of the transparent window should have the following configuration. If it is not so, see Table 7.6.

- L1 RED ON; L2 RED 2 FLASH; drive in Fault for DC bus under voltage;
- L4 GREEN, 1 FLASH, ActualMotorCurrent at 0;
- L3 and L5 OFF;
- L6 OFF, no voltage on the /STO input.


## 4.3. Setup software

**1. MotionDrive installation**
Install the last available version of MotionDrive that you can download from the http://www.bonfiglioli.com website or that is provided by Bonfiglioli . Accept the configurations proposed by the installation procedure. For further details see Section 3.2, *Installation*.

**2. Starting MotionDrive**
Start MotionDrive from: **Start menu** > **Programs** > **Bonfiglioli** > **MotionDrive** and set the Connection parameters in the proposed window *Drive connection* .

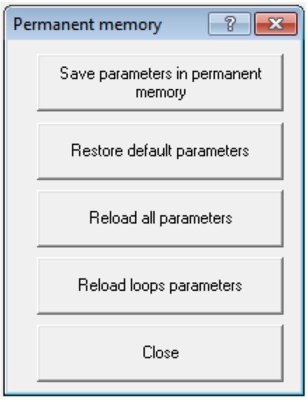



In case of problems see Section 12.3, *Communication errors with MotionDrive* or Section 25.5, *Communication problems*.

### 3. Configuration restoring

Check if you are using the default configuration, push the button 

in the window *Permanent memory* (**Main menu** > **Drive** > **Permanent memory...**)



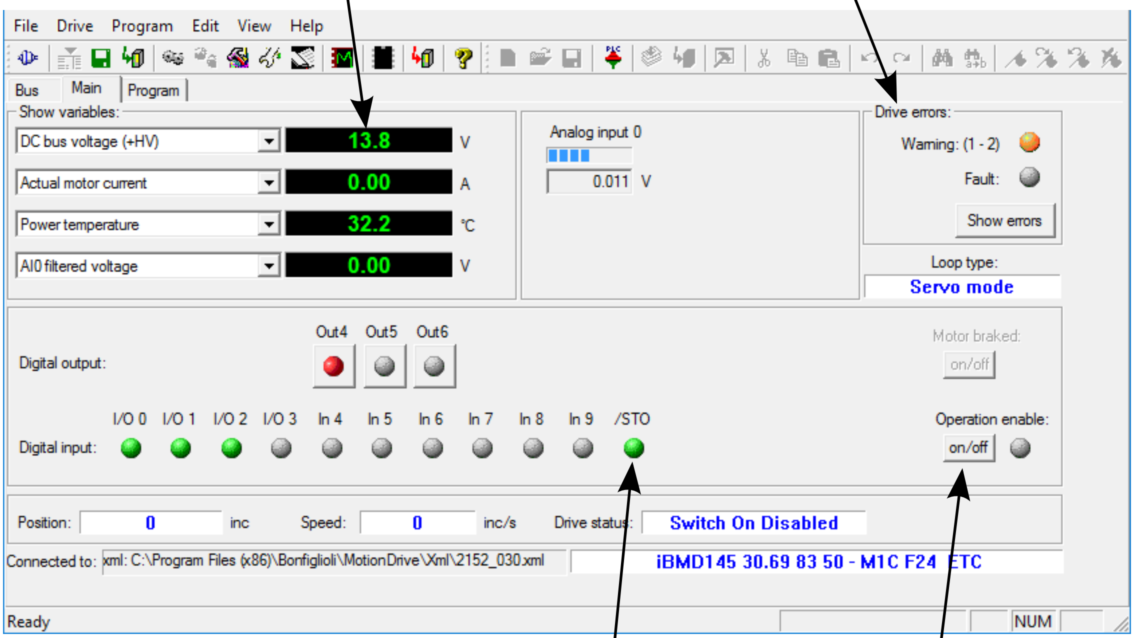

## 4.4. Starting the motion

IMPORTANT

**The quick start guide is designed to perform the movement command only with the MotionDrive tool, without the interference of the fieldbus (CANopen / EtherCAT) master. The MotionDrive tool is not designed to command the *i*BMD drives in one of the real time modes.**

### 1. Enabling the power

After having checked if the connections are correct and safe, switch on the power supply of the power section. Check if the voltage applied is included in the right ranges and reset any errors.




## 2. Enabling/disabling the drive and the motor motion.

- Turn on the voltage to the /STO input and check if the led L6 is switched on (GREEN);
- enable the drive by pushing the On/Off button shown in the previous picture; the drive enters the status Operation enable by giving voltage to the motor phases;
- write a rotation speed:
  - Open the *Object dictionary* window; **Main menu** > **Drive** > **Object dictionary...**
  - write 4700 in the field *Address Modbus* (parameter TargetVelocity);
  - write 8000 in the field *Value*;
  - by pushing the *Write* button, the motor starts moving at 8000inc/s. To modify the speed, modify the value of the parameter TargetVelocity.

IMPORTANT

**To stop the motion, write 0 in the parameter TargetVelocity and only then disable the drive.**

# Chapter 5
# Technical features

## 5.1. *i*BMD technical features

| Power section | | |
|---|---|---|
| Supply voltage | Vdc | Nominal 560; Minimum 275; Maximum 730[a] |
| Internal fuse | - | - |
| Capacity on DC bus | µF | 2.2 for motors with 82/102 mm flange<br>5.45 for motors with greater flange |

[a]The DC bus must be obtained from a TT or TN network; system voltage (nominal voltage between phase and earth): MAX 300V [overvoltage category III]

| Control section | | |
|---|---|---|
| Supply voltage without brake | Vdc | 24 (-15% / +15%) |
| Supply voltage with brake | Vdc | 24 (-10% / + 6%) |
| Threshold drive disabling | Vdc | 18.3 |
| Error threshold for the brake | Vdc | 20.9 |
| Absorbed current @ 24Vdc<br>(control section only) | mA | Nominal 250; Max 500; |
| ADDITIONAL absorbed current @ 24Vdc<br>(if brake is present) | mA | 500 with 4.5Nm brake; 750 with 9Nm brake;<br>1000 with 18Nm brake; 1100 with 36Nm brake. |

| Control section | | |
|---|---|---|
| ADDITIONAL absorbed current @ 24Vdc (with outputs ON) | mA | See Table 16.5 |
| Internal fuse | - | 4A-T not replaceable |

| *i*BMD features (flange of 82mm, 102mm) | | | | | |
|---|---|---|---|---|---|
| | | *i*BMD 82 2.7 | *i*BMD 82 3.8 | *i*BMD 102 5.1 | *i*BMD 102 6.2 |
| Flange dimension | mm | 82 | 82 | 102 | 102 |
| MotorPoles | - | 8 | | | |
| MotorRatedSpeed @ 560Vdc | rpm | 3000 | | | |
| Stall torque (continuous supply with flanged motor on metallic heat sink 280Ø x 20mm) | Nm | 2.7 | 3.8 | 5.1 | 6.2 |
| Torque peak | Nm | 8.4 | 11.0 | 16 | 23 |
| Nominal torque @ 3000rpm e 560Vdc | Nm | 2.4 | 3.3 | 4.5 | 5.5 |
| Motor nominal power @ 3000rpm e 560Vdc | kW | 0.75 | 1.04 | 1.41 | 1.73 |
| Power consumption in continuous functioning [a] | kW | 0.91 | 1.20 | 1.60 | 1.92 |
| Motor inertia moment | kg $m^2 * 10^4$ | 1.40 | 1.70 | 3.70 | 4.70 |
| Maximum radial load @ 3000rpm (applied on the shaft centreline) | N | 470 | 500 | 610 | 650 |
| Maximum axial load (applied on the shaft centreline) | N | 94 | 100 | 120 | 130 |
| Bearings duration | h | 20000 | | | |
| Mechanical Shock according to the IEC 60068-2-27 standard 3 shocks per direction, on 3 axes. Pulse duration of 11ms. | g | 14 | | | |
| Sinusoidal vibration according to the IEC 60068-2-6 standard from 5 to 500 Hz, on 3 axes. | g | 2 | | | |
| Class of insulation | - | F | | | |
| Weight without brake | kg | 4.0 | 5.1 | 6.3 | 7.9 |
| ADDITIONAL weight in version with brake | kg | 0.7 | 0.6 | 1.2 | 1.0 |
| ADDITIONAL weight in versions with fly-wheel | kg | 0.7 | 0.7 | 1.3 | 1.3 |
| Brake static torque at 20°/100° | Nm | 4.5/4 | 4.5/4 | 9/8 | 9/8 |
| ADDITIONAL moment of inertia in version with brake | kg $m^2 * 10^4$ | 0.18 | 0.18 | 0.54 | 0.54 |
| ADDITIONAL moment of inertia in version with fly-wheel | kg $m^2 * 10^4$ | 3 | 3 | 7.5 | 7.5 |
| Working ambient temperature | °C | 0 ÷ 40 | | | |
| Storage ambient temperature | °C | -20 ÷ 70 | | | |
| Humidity related to storage and working (without condensation) | % | 5 ÷ 95 | | | |
| Maximum installation altitude (without adding devices that can limit the overvoltage[b] | m | 2000 m.s.l. | | | |

| *i*BMD features (flange of 82mm, 102mm) | | | | | |
|---|---|---|---|---|---|
| | | *i*BMD 82 2.7 | *i*BMD 82 3.8 | *i*BMD 102 5.1 | *i*BMD 102 6.2 |
| Ventilation | - | Natural | | | |
| Pollution degree | - | 3[c] | | | |
| Protection degree | - | IP65[d] if the connectors are inserted | | | |

[a]at 3000rpm, at the overtemperature limits and with environment temperature of 40°C
[b]for an installation altitude between 2000m and 4000m m.s.l., it's necessary to install an overvoltage limiter device on the machine to oppose the transient overvoltages, so that the power circuit overvoltages are limited to the overvoltage category II. It can be achieved with a galvanic insulation transformer.
[c]Conductive pollution or dry non conductive pollution that can became conductive in case of condensation
[d]anterior flange excluded.

| *i*BMD features (flange of 145mm, 170mm) | | | | | |
|---|---|---|---|---|---|
| | | *i*BMD 145 14.5 | *i*BMD 145 18.5 | *i*BMD 170 29 | *i*BMD 170 36 |
| Flange dimension | mm | 145 | 145 | 170 | 170 |
| MotorPoles | - | 8 | | | |
| MotorRatedSpeed @ 560Vdc | rpm | 3000 | | | |
| Stall torque (continuous supply with flanged motor on metallic heat sink 390Ø x 20mm) | Nm | 14.5 | 18.5 | 29 | 36 |
| Torque peak | Nm | 39 | 45 | 62 | 70 |
| Nominal torque @ 3000rpm e 560Vdc | Nm | 11 | 12.5 | 14.7 | 21 |
| Motor nominal power @ 3000rpm e 560Vdc | kW | 3.45 | 3.93 | 4.62 | 6.60 |
| Power consumption in continuous functioning [a] | kW | 3.85 | 4.33 | 5.05 | 7.16 |
| Motor inertia moment | $kg\ m^2 * 10^4$ | 12.8 | 17.6 | 28.2 | 47.5 |
| Maximum radial load @ 3000rpm (applied on the shaft centreline) | N | 1150 | 1200 | 1400 | 1500 |
| Maximum axial load (applied on the shaft centreline) | N | 229 | 240 | 285 | 305 |
| Bearings duration | h | 20000 | | | |
| Mechanical Shock according to the IEC 60068-2-27 standard 3 shocks per direction, on 3 axes. Pulse duration of 11ms. | g | 14 | | | |
| Sinusoidal vibration according to the IEC 60068-2-6 standard from 5 to 500 Hz, on 3 axes. | g | 2 | | | |
| Class of insulation | - | F | | | |
| Weight without brake | kg | 17.6 | 20.6 | 27.4 | 32.4 |
| Additional weight in version with brake | kg | 2.6 | 2.6 | 4.5 | 4.5 |
| ADDITIONAL weight in versions with fly-wheel | kg | 3.6 | 3.6 | 5.5 | 5.5 |
| Brake static torque at 20°/100° | Nm | 18/15 | 18/15 | 36/32 | 36/32 |
| Additional moment of inertia in version with brake | $kg\ m^2 * 10^4$ | 1.66 | 1.66 | 5.56 | 5.56 |
| ADDITIONAL moment of inertia in version with fly-wheel | $kg\ m^2 * 10^4$ | 36 | 36 | 70 | 70 |

| *i*BMD features (flange of 145mm, 170mm) | | | | | |
|---|---|---|---|---|---|
| | | *i*BMD 145 14.5 | *i*BMD 145 18.5 | *i*BMD 170 29 | *i*BMD 170 36 |
| Working ambient temperature | °C | 0 ÷ 40 | | | |
| Storage ambient temperature | °C | -20 ÷ 70 | | | |
| Humidity related to storage and working (without condensation) | % | 5 ÷ 95 | | | |
| Maximum installation altitude (without adding devices that can limit the overvoltage[b] | m | 2000 m.s.l. | | | |
| Ventilation | - | Forced with integrated fans | | | |
| Pollution degree | - | 3[c] | | | |
| Protection degree | - | IP65[d] if the connectors are inserted | | | |

[a]at 3000rpm, at the overtemperature limits and with environment temperature of 40°C

[b]for an installation altitude between 2000m and 4000m m.s.l., it's necessary to install an overvoltage limiter device on the machine to oppose the transient overvoltages, so that the power circuit overvoltages are limited to the overvoltage category II. It can be achieved with a galvanic insulation transformer.

[c]Conductive pollution or dry non conductive pollution that can became conductive in case of condensation

[d]anterior flange excluded.

NOTE

**The torque values and the related power values are referred to the maximum allowed functioning environment temperature, that is 40°C.**

## 5.2. *i*BMD dimensions and sizes

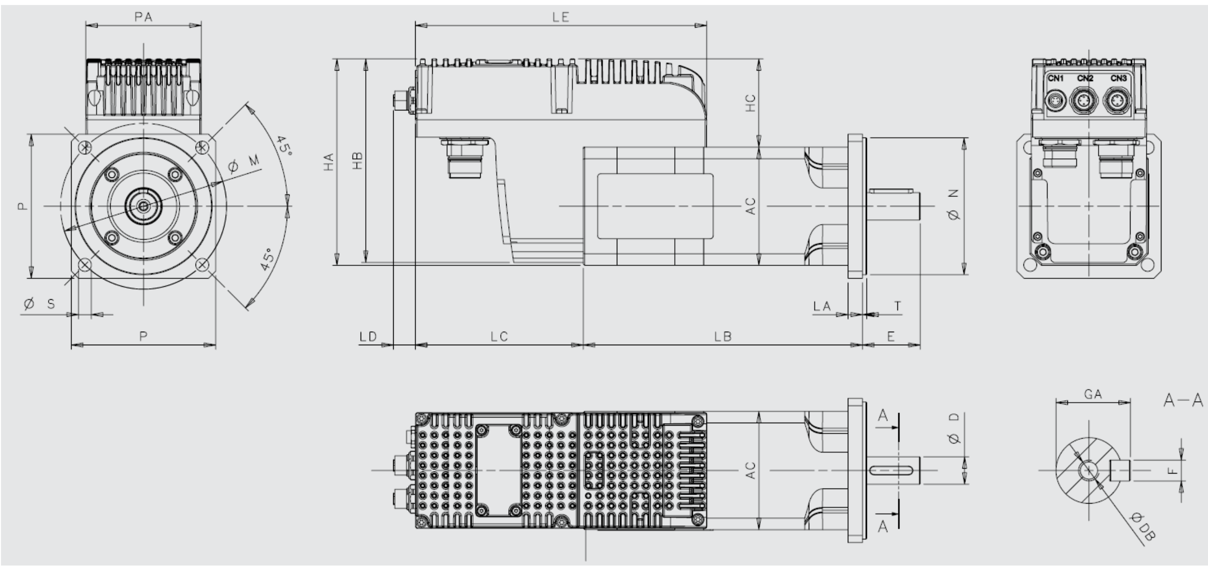



*Figure 5.1. i*BMD *dimensions*

| Size | Shaft | | | | |
|---|---|---|---|---|---|
| | D | E | DB | GA[a] | F[a] |
| 82 | 11 | 23 | M4 | 12.5 | 4 |
| | 14 | 30 | M5 | 16 | 5 |

| Size | Shaft | | | | |
|---|---|---|---|---|---|
| | D | E | DB | GA[a] | F[a] |
| | 19 | 40 | M6 | 21.5 | 6 |
| 102 | 19 | 40 | M6 | 21.5 | 6 |
| | 24 | 50 | M8 | 27 | 8 |
| 145 | 19 | 40 | M6 | 21.5 | 6 |
| | 24 | 50 | M8 | 27 | 8 |
| | 28 | 60 | M10 | 31 | 8 |
| 170 | 24 | 50 | M8 | 27 | 8 |
| | 28 | 60 | M10 | 31 | 8 |
| | 32 | 60 | M12 | 35 | 10 |

[a]Available in no-keyed shaft versions

| Size | Flange | | | | | |
|---|---|---|---|---|---|---|
| | M | N | P | S | T | LA |
| 82 | 100 | 80 | 82 | 6.5 | 3 | 10 |
| | 115 | 95 | 100 | 9 | 3 | 10 |
| 102 | 100 | 80 | 102 | 7 | 3 | 10 |
| | 115 | 95 | 102 | 9 | 3 | 10 |
| 145 | 165 | 130 | 145 | 11.5 | 3.5 | 12 |
| 170 | 165 | 130 | 170 | 11.5 | 3.5 | 12 |

| Size | Motor with integrated drive | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | T0 | AC | PA | LB2[a] | LB3[b] | LC | LD | LE | HA | HB | HC |
| 82 | 2.7 | 82 | 80 | 121 | 174 | 117 | 16 | 202 | 144 | 142 | 62 |
| | 3.8 | | 80 | 141 | 194 | | | | | | |
| 102 | 5.1 | 102 | 80 | 141 | 191 | 117 | 16 | 202 | 164 | 142 | 62 |
| | 6.2 | | 80 | 168 | 218 | | | | | | |
| 145 | 14.5 | 145 | 142 | 228 | 275 | 120 | - | 300 | 225 | 222 | 80 |
| | 18.5 | | 142 | 228 | 310 | | | | | | |
| 170 | 29 | 170 | 142 | 233 | 305 | 120 | - | 300 | 225 | 222 | 80 |
| | 36 | | 142 | 286 | 357 | | | | | | |

[a]Standard motor length
[b]Standard motor length with brake or fly-wheel

# 5.3. Downgrading with altitude

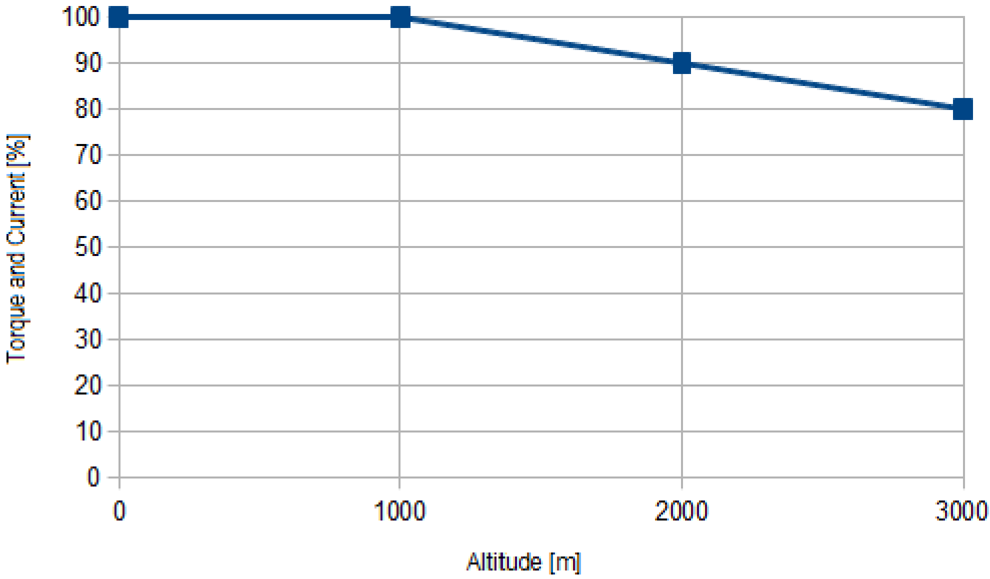



*Figure 5.2. Torque and current downgrading in relation to the altitude.*

# Chapter 6
# STO safety function: Safe Torque Off

IMPORTANT

**By “STO” is meant the safety function, while in order to refer to the physical input and to the external signal it is used the “/STO”. In this last definition the bar "/" represents the "NOT" logical function, to indicate that the safety function removes the motor torque if the signal voltage is at low logic level.**

## 6.1. General informations

The STO integrated function is implemented in the drive according to the EN 61800-5-2:2007, EN ISO 13849-1:2008/AC:2009 standards. When, the digital input the function STO is linked to is switched to the low logic level, the motor torque is disabled according to a stop of category 0, as defined in the EN 60204-1:2006/A1:2009 standards.

⚠ WARNING

**If the digital input with /STO function is disabled, the drive power section is disabled without cutting the DC bus voltage and it is not possible to control the motor motion anymore. Always stop the motor before to switch to the low logic level the /STO input. In case of suspended loads, some other measures in order to reduce the risk of load falling must be considered, for example installing a dynamic brake.**

⚠ WARNING

**It must be made an evaluation of the risk about the particular machine in which the product is used. In consequence the user must take appropriate measures to avoid risks to the safety of the person. In order to guarantee the safe removal of the motor torque, it is recommended to use the "Safe Torque Off" function, by using only the provided input named /STO and the related instructions included in this guide.**

The examples and the procedures described in this manual are based on the reaching of the de-energized state of the drive as safe state (e.g. in case of emergency).

In case there are external forces on the load (e.g. in case of vertical loads), some additional actions must be considered in order to prevent danger risks (e.g. by using a mechanical brake, eventually commanded by a safe output).

⚠ DANGER

**The Safe Torque Off function DOESN'T cut off the voltage nor in the drive power and logic circuits neither in the motor, therefore it CANNOT be considered as an insulation system of the drive from supply sources (DCbus). In order to execute the maintenance service on the drive electrical components or on the motor, it is necessary to insulate the supply system first.**

⚠ DANGER

**The /STO input is not protected against overcurrent: the user, if he deems it appropriate, can provide external protections.**

The STO function can be used to prevent an unintentional start of the motor: the STO function use is possible in case some quick operations have to be executed (e.g. the machine cleaning) and/or for maintenance services on NOT electrical parts of the machine, without cutting off the drive supply voltage.

⚠ WARNING

**It is recommended to not stop the drive and/or the motor by using the Safe Torque Off function. If the motor is stopped through the STO function, the drive cuts off the power to the motor and it stops due to the inertia. Furthermore, in the motors that are provided of the internal holding brake, the motor brake can be damaged. In order to avoid these dangerous/damaging situations, stop the drive and the motor with the provided modes before to use the STO function.**

It has been provided a diagnostics system of the STO system status, that allows the anomalies report to the user.

NOTE

**The Safety system has been completely hardware realized: the STO functionality is independent from the software configuration and version of the drive. The software manages only the anomalies warning signal, but doesn't prevent the system safety setting.**

**Environment conditions and EMI**
The environment and operating conditions in which the Safety system functioning is provided and guaranteed, are the same ones of the entire system (see Chapter 5, *Technical features*).

**Stop categories (IEC EN 60204-1)**
**Uncontrolled stop:** the intervention of the STO function disables the drive and cuts off the torque from the motor, therefore the axis is free to move according to the uncontrolled stop of cat.0 (EN 60204-1).
**Controlled stop:** if the application requires a controlled stop according with the stop of cat.1 (EN 60204-1), the following actions must be executed respecting the listed order:

1. deceleration of the motor through the braking ramp, executed by the drive (see Section 22.3, *Carrying out a stop by using the master*);
2. drive disable (this operation must be executed when the motor is in standstill)[1] (see Section 22.2, *Disabling by using the master*);
3. motor torque safe removal through the /STO input switched to the low logic level;

In the following picture a graphic representation of the above described phases is reported.

[1]The voltage must be cut off after a delay, that has to be programmed on the safety relay, that has to be sufficient to include the stop of the motor; otherwise the final part of the movement will be uncontrolled.

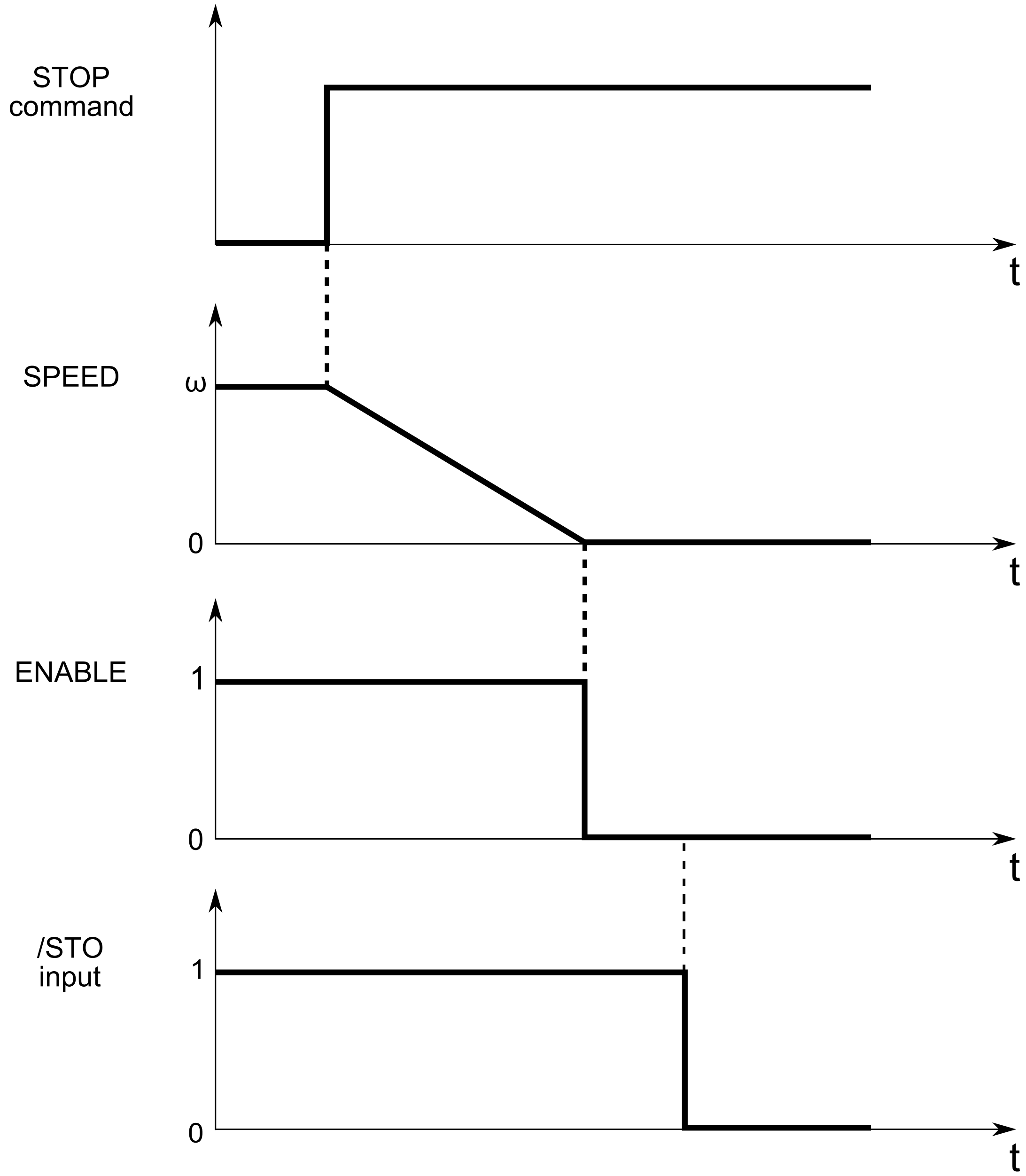



*Figure 6.1. Steps of a controlled stop*

NOTE

**The digital inputs to which is possible to associate the Enable functionality are not part of the Safety system system and, consequently, the motor deceleration is not a safety function.**

**Residual risk**
In case of fault for short circuit on one or more IGBT power semiconductors, despite the safe removal of the motor torque, there is the residual risk that the drive produces on the motor shaft a maximum rotation of 360° / (2p), where 2p is the motor poles number.

## 6.2. *i*BMD functional specifications

### Safety system

In the picture below the bloks that are part of the Safety system with STO function are included in the yellow/black dotted outline, marked with the «SAFETY SYSTEM» writing. The references in the picture, represented by the yellow circled numbers, are related to the following parts of the system:

1. /STO input on CN5 connector;
2. STO input circuit;
3. IC controller for the management of the converter that supplies the gate drivers;
4. Converter for the gate drivers voltage supply;
5. Gate driver for the IGBTs.

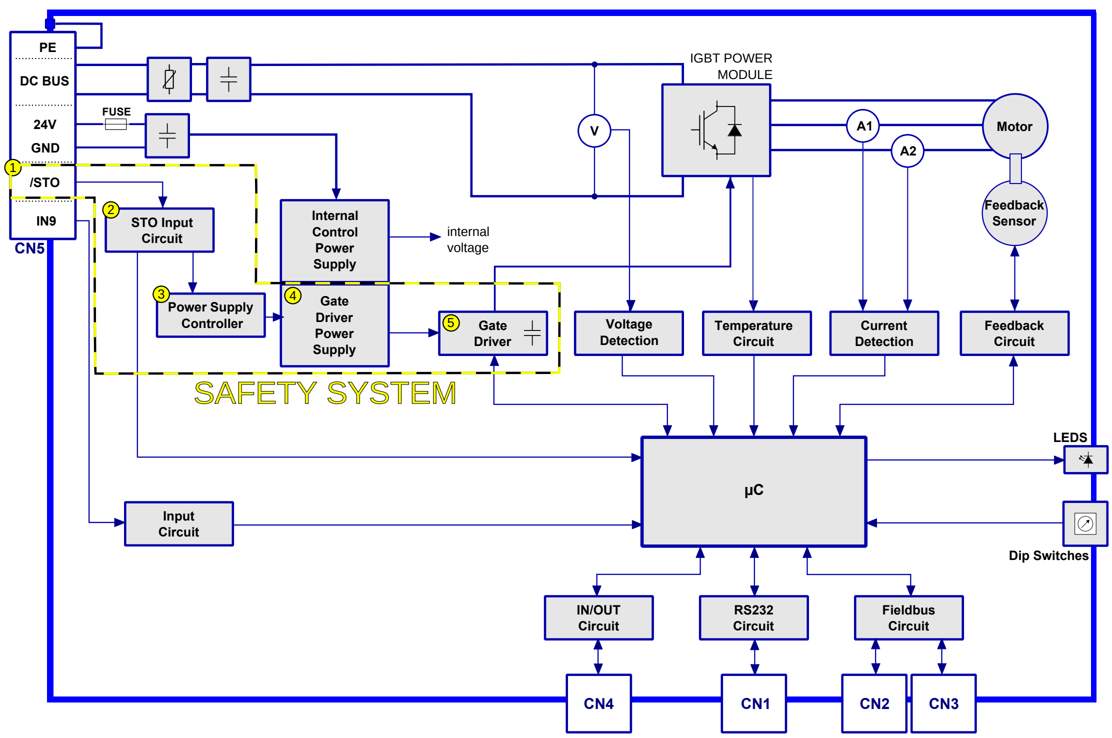



*Figure 6.2. System block diagram*

### Mode of operation

The STO Safety system allows the deactivation of the control voltage of the power semiconductors (IGBT) of the drive output stage through the input /STO signal, avoiding the generation of enough voltage to provide power to the motor.

The states of the Safety system are the following:

- in case the voltage that's applied to the /STO input is at high logic level: the STO Safety system allows the drive enabling and so the torque may be present on the motor **(potentially not safe status)**;

- in case the voltage that's applied to the /STO input is at low logic level: the STO Safety system cuts off the voltage on the motor phases, so that there is no torque and the automatic start-up is disabled **(safe status)**;

After the disabling, the drive can be enabled after these operations have been executed:

- restore of the high logic level on the /STO external input;
- deleting of the alarms via software;
- sending of the command to switch the drive in the Drive enable state.

The safety function is independent of the status of the drive: it is always active and continuously executed. In fact there are no configurations able to temporary disable the safety function.

### /STO input electric features

| /STO INPUT | |
|---|---|
| Input type | PNP |
| Input current (typical) with Vin = 24Vdc | 10.5 mA |
| Input voltage (low or high logic level) | |
| Nominal | +24Vdc |
| for low signals | -30V ÷ +5Vdc |
| for high signals | +20V ÷ +30Vdc |

ATTENTION

**The /STO input is compatible with the auto-diagnostic digital outputs of a command device, in which the test pulse has a maximum duration of 1ms. This implies that the motor torque is not removed if the /STO input receives a low logic level pulse with a duration lower than 1ms.**

### Led 6 diagnostics

The STO system logic state (for the meaning see Mode of operation) is monitored by a microcontroller and signaled through the software (bit 24 di LogicalDigitalInputStatus) and the Led 6, as reported in the following table:

| STO logic status | Led 6 |
|---|---|
| Voltage presence on the input and Fault absence | ON |
| Voltage absence on the input and/or Fault presence | OFF |

IMPORTANT

**In the boot and firmware start-up phases of the *i*BMD drives, the above description of the led is no more valid. Please refer to Chapter 26, *Software updating***

### Continuous /STO input validation

The /STO input validation function continuously monitors the voltage level applied to the input: if this level keeps a value between the reference thresholds (+20V for the high level and +5V for the low one, see /STO input electric features) for more than 500ms, the error is reported through the MainError (see the bit 14 in the Table 24.1).

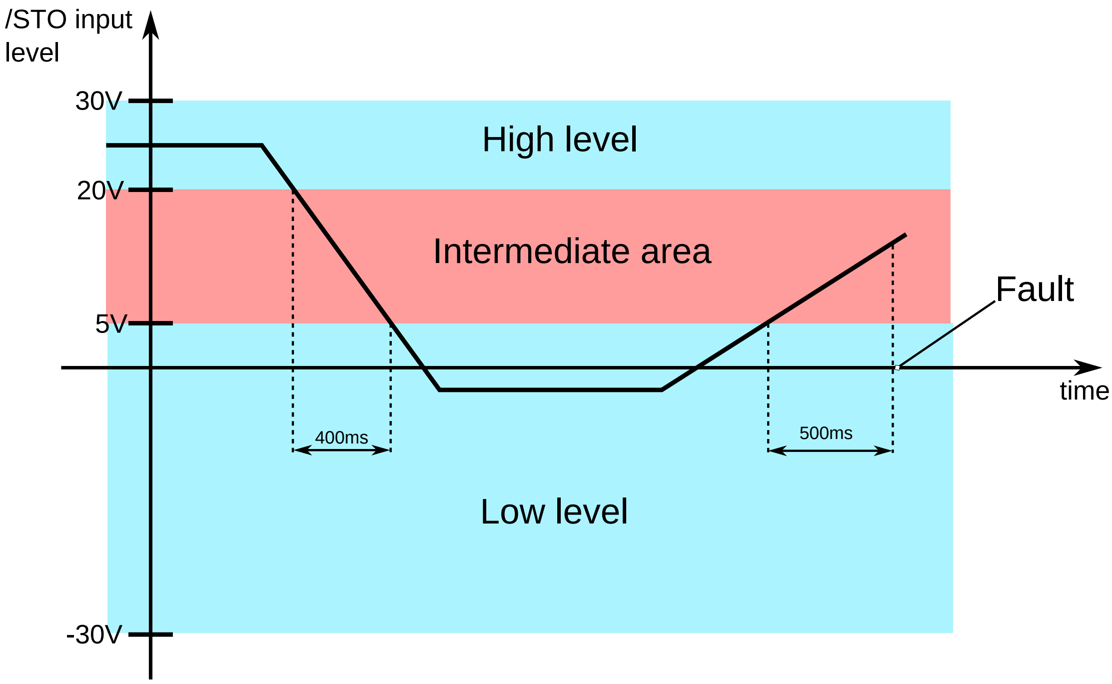



***Figure 6.3. Voltage levels of the /STO input***

**Environment conditions and EMI**

The environment and operating conditions in which the Safety system functioning is provided and guaranteed, are the same ones of the entire system (see Chapter 5, *Technical features*).

**Safety related data**

The STO function is completely implemented via hardware as safety function with single channel.

It is furthermore provided a STO system status monitoring circuit that detects the presence of failures on the Safety system.

| Functional integrity level (EN 61800-5-2) | Performance level (ISO EN 13849-1) |
|---|---|
| SIL3 | PLd |

According to the EN 61800-5-2 standards it is guaranteed a probability of dangerous failure PFHd = $6.27*10^{-8}$ with safe failure fraction SFF = 99%, obtaining a Functional safety integrity level equal to SIL3.

According to the ISO EN 13849-1 standard, the STO Safety system is characterized by an architecture that complies to the Cat. 2 with DC=90% reaching a performance level equal to PLd. The applied failure exclusions correspond to the ones that are reported in the Attached D of the EN 61800-5-2 standard.

### Response time

> ⚠ ATTENTION
>
> **The STO function has a maximum response time of 120ms. This time has been measured by cutting the voltage applied to the /STO input (the maximum allowed, 30V), with negligible external capacity.**

The response time corresponds to the time interval between the instant of the /STO input state transition from high to low logic level and the instant in which the voltage that's present on the IGBT supply circuit is reset to zero, that coincides with the system safety state.

### STO system diagnostics

The Safety system includes a diagnostics, that works via software, that detects the hardware faults of the STO circuit. For a description of the errors that are detected by the STO system, please refer to /STO Management Error.

### Risk mitigation

The STO function has a single channel architecture. It implies that if in the /STO external signal connection happens a single fault that provides to the input enough energy to keep it at the high logic level (e.g. a /STO signal short circuit with a voltage of 24V), and this fault is not detected, it may occur a STO function interruption: in order to reduce the risk due to any eventual failure, the external connection of the /STO signal must be protected from faults that accidentally may provide it a voltage that puts the system in a not safe condition. It can be obtained in several modes:

a. by completely isolating the /STO signal from the output of the device that commands the *i*BMD input; or
b. by shielding the /STO signal connection cable and connecting the shield to GND (/STO signal command voltage potential reference); or
c. by monitoring the /STO signal status from the control device (e.g. by using a control device with an output, which can generate a test pulse, connected to the /STO) that allows an independent disabling of the drive.

### /STO input connection examples

The connection examples in /STO input connection examples report the general scheme for the /STO input wiring and don't have to be intended as detailed and complete pictures.
The integration of the STO function in the machine must be the result of the complete risks analysis of the machine made by the machine designer.

### Test and periodic check of the STO function

The machine final assembly manager must do the test and the periodic check of the safety functions, verifying the correct functioning.
The periodic test/check procedure must be done with a frequency that respects the following situations:

- at the first safety function start-up;
- after every modification related to the safety function (wiring, settings, ecc.);
- after every system maintenance intervention;
- at least every 24 months (2 years), starting from the first put in service of the system and, in any case, after the machine has not been used for an equal or greater time period;

The test/periodic check of the safety functions must be executed only by expert and authorized personnel, that have the necessary competence related to the use of the functions and knows the risks in case the safety functions don't work as they should. The periodic test/check must be documented and undersigned by authorized personnel and the reports of the result of the test, eventual faults alarms and problems resolutions must be archived.

**Test/periodic check procedure**

- check that the STO circuit wirings related to the drive and the control circuit are correctly executed as reported in /STO input connection examples;
- check that the shield of the /STO input cable is electrically connected to the GND in correspondence of the signal source and the drive connector (mandatory test in case the mode “b” has been adopted, in reference to the Risk mitigation paragraph);
- with /STO input at high logic level and the drive in the Drive enable status, check that the torque is present on the motor and that there are no safety alarms;
- disable the drive and, successively, cut off the voltage from the /STO input (through the control device/s suitable for this function). Check that the “/STO = 0V with drive enabled error” or other safety alarms are not presents (see /STO Management Error) and that the motor is free to move (torque absence);
- when the drive is disabled and the /STO input is at low logic level, enable again the drive. Check that the alarm report related to the enable attempt with /STO input at low logic level ("/STO = 0V with drive enabled error”) is present and that the motor is still free to move (torque absence - safe status);
- in these conditions switch the /STO input to the high logic level and check that the motor is still free to move (torque absence - safe status); furthermore, check that the fault written above is present;
- reset the alarm and, with the /STO input set to the high logic level, enable the drive. In these conditions the motor must be enabled (on torque) and neither the "/STO = 0V with drive enabled error” alarm nor other safety alarms must be present.

⚠ ATTENTION

**In case some inconsistencies are detected, in relation to the overwritten expected results, the Safety system cannot be considered intact and the drive mustn't be used. Contact Bonfiglioli Mechatronic Research S.p.a in order to obtain an adequate support and for the record of the fault event: in addition, it will be necessary to proceed with the repairing operation by the authorized personnel. This procedure is necessary to guarantee the safe use of the drive.**

# Chapter 7
# Electrical connections, leds and dip switches

## 7.1. Installation notes

⚠ ATTENTION

**The *i*BMD systems must be installed by specialized personnel olny that must have an in-depth knowledge about the safety requirements and the electromagnetic compatibility (EMC). The planner has the responsibility to guarantee that the product or the final system comply to the pertinent regulations that are in force in the country in which the product (or the entire system) is used.**

IMPORTANT

**The producer must analyze the risks and apply the correct measures to avoid damages to people or things that may be caused by unexpected movements (due for example to a drive or its command system anomaly).**

⚠ NOTICE

**The *i*BMD system must be installed in an environment that guarantees the conditions that this manual prescribes (see Section 2.3, *Safety precautions and limits*), in particular it must be protected from excessive humidity and/or condensation. Furthermore it must be respected the maximum environment temperature (see Chapter 5, *Technical features*), considering that the heat that's produced by the system must be adequately dissipated in order to not exceed the maxi-**

**mum working temperature. To ensure the maximum reliability of the system and of the related installation, the regular controls for the maintenance of the overwritten conditions must be done.**

IMPORTANT

**Before to make any intervention (as for example the transparent cover removal for the settings of the communication bus) always disconnect the voltage supply through an approved isolation device and wait at least 1 minute to be sure that the residual voltages will revert to the security levels. Please consider also that the permanent magnets motors generate electric power if they are rotated, even when the system supply is disconnected. Therefore pay attention if the load connected to the motor may rotate it when the drive is not powered.**

⚠ NOTICE

**The removing of the transparent window to set the dip switches exposes the electronics to the external environment, causing the risk of an involuntary infiltration of foreign bodies that may cause damages. Limit the window opening to the strictly necessary time to set the dip switches.**

## 7.2. Electrical connections

The section about the electrical connection includes both the connectors pins and the characteristics and the description of the different parts which the system is made of; in particular the supply section, with the related limits, and the interface section (communication bus, digital inputs and outputs, analog input, debug serial port).

⚠ NOTICE

**A correct cable, ground and shield wiring is essential for the drive safety and correct functioning. It's better if the cables are not interrupted; if it is not possible, be sure that the interruptions are reduced to the shortest possible length. It's recommended to always wire the cables without voltage presence.**

### 7.2.1. System Supply

For the system supply a voltage for the control section and another one, separated from the first one, for the power section are necessary. Both these voltages must be of DC type (direct voltage) The connector for the voltage supply is CN5.

There are no restrictions about the supply sequence: it can be provided the control voltage supply first and then the power one, or vice versa. But without the control voltage the system doesn't turn on, therefore in this situation the leds don't light and it's not possible any communication (even if the power voltage is present). In the technical data table in the Chapter 5, *Technical features* there are the limits of the control and power sections voltage.

**Connection notes**

To connect the voltage supply use a shielded cable with an adequate section. The cable shield must be connected to the ground on the power pack side.

The grounding of the drive is made through the grounding wire of CN5 that must be connected to the equipotential collector of the machine. In order to the safety, to a well functioning of the drive and to a better behaviour against the noises, it's necessary to make the connection of the metallic structure, where the motor flange is fixed, to the ground equipotential network (through a low impedance conductor with a not lower than 4 $mm^2$ section).

**ATTENTION**

**NEVER apply neither an AC type voltage (alternating voltage), nor a DC type voltage (direct voltage) out of the described limits range or with a reverse polarity than the one described in the manual: this may cause the damage of the power and/or control sections of the drive, and imply electrical arcs or fire risk.**

**ATTENTION**

**The drive is provided of a control in case of overvoltage or undervoltage, so that the drive is disabled if there are some supply problem, but this doesn't exclude to maintain the voltage between the limits, in particular in case of overvoltage. In fact, no “dump” circuit on the supply voltage is present.**

**ATTENTION**

**The unit is NOT protected against the +HV supply polarity reverse: pay attention during the connector wiring.**

**NOTICE**

**The control section voltage supply must be guaranteed "on the system connector level". Be sure that this range is respected in particular if a long cable is used (eventually compensate the voltage drop in the cable by giving a higher voltage upstream).**

**ADVICE**

**Refer to Chapter 14, *Power configuration*.**

## 7.2.2. Fuses

**Control section**

The drive is provided, internally to the control section, of a non replaceable fuse (SMT type): DO NOT SUBSTITUTE FOR ANY REASON THIS FUSE. In fact the fuse breaking probably implies a damage of the electronics: in this case please contact Bonfiglioli Mechatronic Research S.p.a

**Power section**

On the power section there are no fuses.

## 7.2.3. Field bus (CAN)

The CAN version drives are provided with a CANopen serial communication port for the connection with a network master with DS301 standard protocol and a maximum communication speed of 1MBaud. To facilitate the installation the dip switches for the baud rate and node ID selection and for the termination resistance insertion are presents on the drive (see Section 7.4, *Dip switches*).

On the system are also provided two connection connectors CAN-IN and CAN-OUT (CN2 e CN3) to which is possible to connect the CAN input cable from the previous node and the output cable to the next one.

In the next table there are the main characteristics of the serial connection. About the general characteristics of the serial connection, of the topology, of the maximum connectable nodes number, of the baud rate/length relationship, and of the transmissive device specifications, it's necessary to refer to the specific manual "CANopen Net peripherals”.

| CAN SERIAL CONNECTION CHARACTERISTICS | |
|---|---|
| Protocol | CAN (ISO-11898 Ver. 2.0 Part B) |
| Admitted Baud Rate | 20, 50, 100, 125, 250, 500, 1000 Kbaud |
| Galvanic isolation | YES |
| Termination resistance | Can be inserted through the DP4 dip switch (see Table 8.2) |
| Communication protocol | CANopen DS301 |

**Transmission medium: CAN cable**

The transmission medium that has to be used for the physical connection must be a *shielded cable with 2 twisted pairs*. In particular you have to use *a pair for the CAN_H and CAN_L signals and the other one for the CAN_GND*. It's necessary to use only one type of cable on a single network.

In the Table 7.2 and Table 7.3. the characteristics for the realization of a cable for a physical CANopen network are described.

## 7.2.4. *i*BMD drive connectors

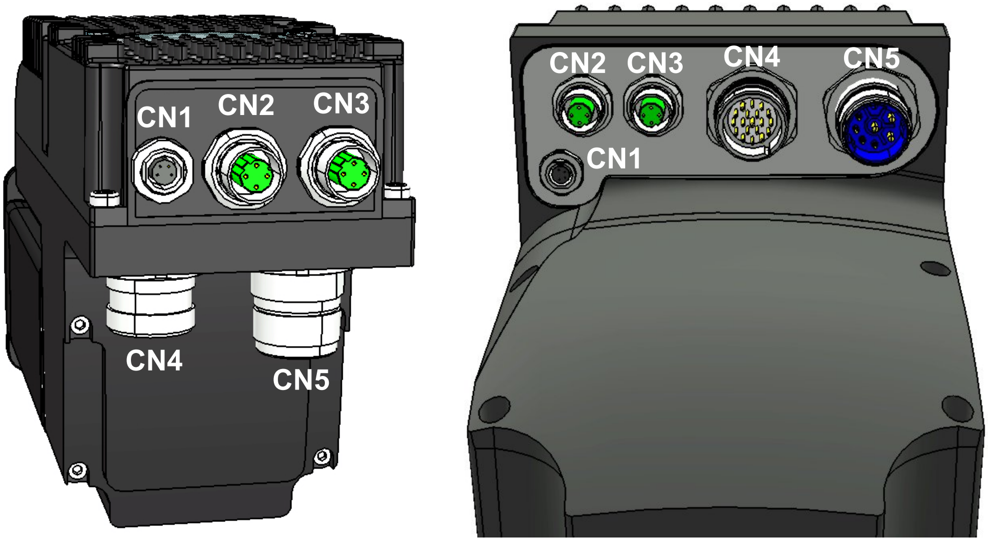
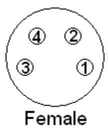



*Figure 7.1. Connectors arrangement: flange rear view 82-102mm (left) and flange lower view 145-170mm (right).*

**CN1 Auxiliary bus (RS232 Serial port)**
Connector for the auxiliary bus with Modbus protocol on RS232, M8 female, 4 poles (this serial port is insulated).

IMPORTANT

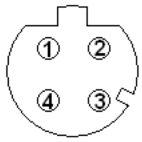
**Make the serial cable connection when the drive is not powered. The cable shield must be connected to the ground both on the host (PC) side and on the drive side, by using the M8 connector chassis. If between the drive PE earth potential and the connected master system potential (e.g. a PC) there is a non zero difference, it's necessary to make equipotential the two references. When it is not possible, connect the serial cable shield on one side only.**




| PIN | Signal | Description |
|---|---|---|
| 1 | TX232 | Transmit Data RS232 |
| 2 | RX232 | Receive Data RS232 |
| 3 | NC | Reserved, not connected |
| 4 | GND_COM | Ground RS232 |
| Chassis | PE | Protection Earth |

### CN2 e CN3 Main bus (ETC)

Connectors for the main bus with EtherCAT protocol, M12 female, 4 poles, D-code, output and input respectively of CN2 e CN3.



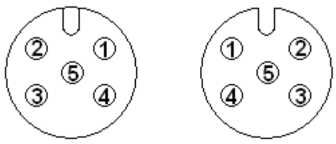

| PIN | Signal | Description |
|---|---|---|
| 1 | TX Data+ | Transmit Data + |
| 2 | RX Data+ | Receive Data + |
| 3 | TX Data- | Transmit Data - |
| 4 | RX Data- | Receive Data - |
| Chassis | PE | Protection Earth |

### CN2 e CN3 Main bus (CAN)

Connectors for the main bus with protocol CANopen, M12 male and female, 5 poles, A-code.

IMPORTANT

**The cable shield must be continuous on the whole line and must be connected to the earth through the network master, while on the *i*BMD peripherals the shield must be connected on the pin 1 of the M12 connectors.**



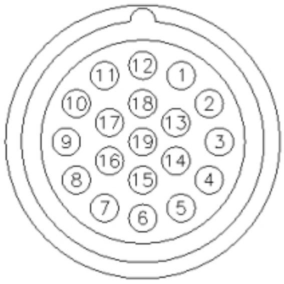

| PIN | Signal | Description |
|---|---|---|
| 1 | SHIELD | Shield |
| 2 | NC | Reserved, not connected |
| 3 | GND_CAN | Ground CAN |
| 4 | CAN-H | CAN High |
| 5 | CAN-L | CAN Low |
| Chassis | PE | Protection Earth |

### CN4 Input/Output

Connector for the digital and analog inputs and outputs, M23 male, 19 poles (16+3), Hummel.

NOTE

**The PNP digital inputs (24V) have the common ground internally connected to the system on the GND signal, that is the 24V supply ground present on CN5-pin B. For this reason it's sufficient to connect on the inputs a signal which level is referred to this ground.**



Male, Hummel

| PIN | Signal | Description |
|---|---|---|
| 1 | IN/OUT1 - | Differential digital Input/Output 1 (-) |
| 2 | IN/OUT2 - | Differential digital Input/Output 2 (-) |
| 3 | AN_IN - | Analog Input (-) |
| 4 | AN_IN + | Analog Input (+) |
| 5 | IN/OUT2 + | Differential digital Input/Output 2 (+) |
| 6 | GND_5V | Ground of +5V |
| 7 | +5V | +5V Supply (max 150mA) for master encoder |
| 8 | IN8 | Digital Input 8 |
| 9 | OUT5 | Output 5 |
| 10 | IN/OUT3 | Digital Input/Output 3 |
| 11 | IN7 | Digital Input 7 |
| 12 | IN/OUT0 - | Differential digital Input/Output 0 (-) |
| 13 | IN/OUT0 + | Differential digital Input/Output 0 (+) |
| 14 | IN/OUT1 + | Differential digital Input/Output 1 (+) |
| 15 | IN4 | Digital Input 4 |
| 16 | OUT4 | Output 4 |
| 17 | OUT6 | Output 6 |
| 18 | IN6 | Digital Input 6 |
| 19 | IN5 | Digital Input 5 (the function Simulated GND is available) |
| Chassis | | |

IMPORTANT

**In/Out0, In/Out1, In/Out2 are differential inputs they DON'T have to be connected to 24V signals. It's recommended to respect the maximum differential voltage and to report this voltage to the GND_5V ground [pin 6 of CN4].**

IMPORTANT

**Absolutely avoid to place the I/O signals cable in parallel to the power cables by suitably selecting separated paths. It's recommended to use a shielded cable for the connection and to connect the shield to the metallic part of the M23 circular connector. On the controller/PLC side follow the constructor instructions about the shield connection.**

## Example of PNP 24V inputs and outputs wiring

On the *i***BMD** system PNP inputs may be connected some devices with PNP 24V output. The ground reference of these outputs must be the same on which the *i***BMD** system control supply is referred (pin B of CN5, GND signal). In fact, as can be seen on the following diagram, the inputs have a system internal common ground that's connected on the GND signal. The *i***BMD** system outputs are internally powered by the 24V with which the control section is powered (pin D of CN5). On this voltage there is a current limiter that is a protection in case of overload or short circuit on the outputs themselves. The ground of the loads that are connected to the outputs must be the same one of which the *i***BMD** system control supply is referred (pin B of CN5, GND signal).

In the following figure an outputs and inputs connection example is reported in which the 24V voltage for the inputs supply and the outputs ground reference are made through connections that are external to the *i***BMD**.

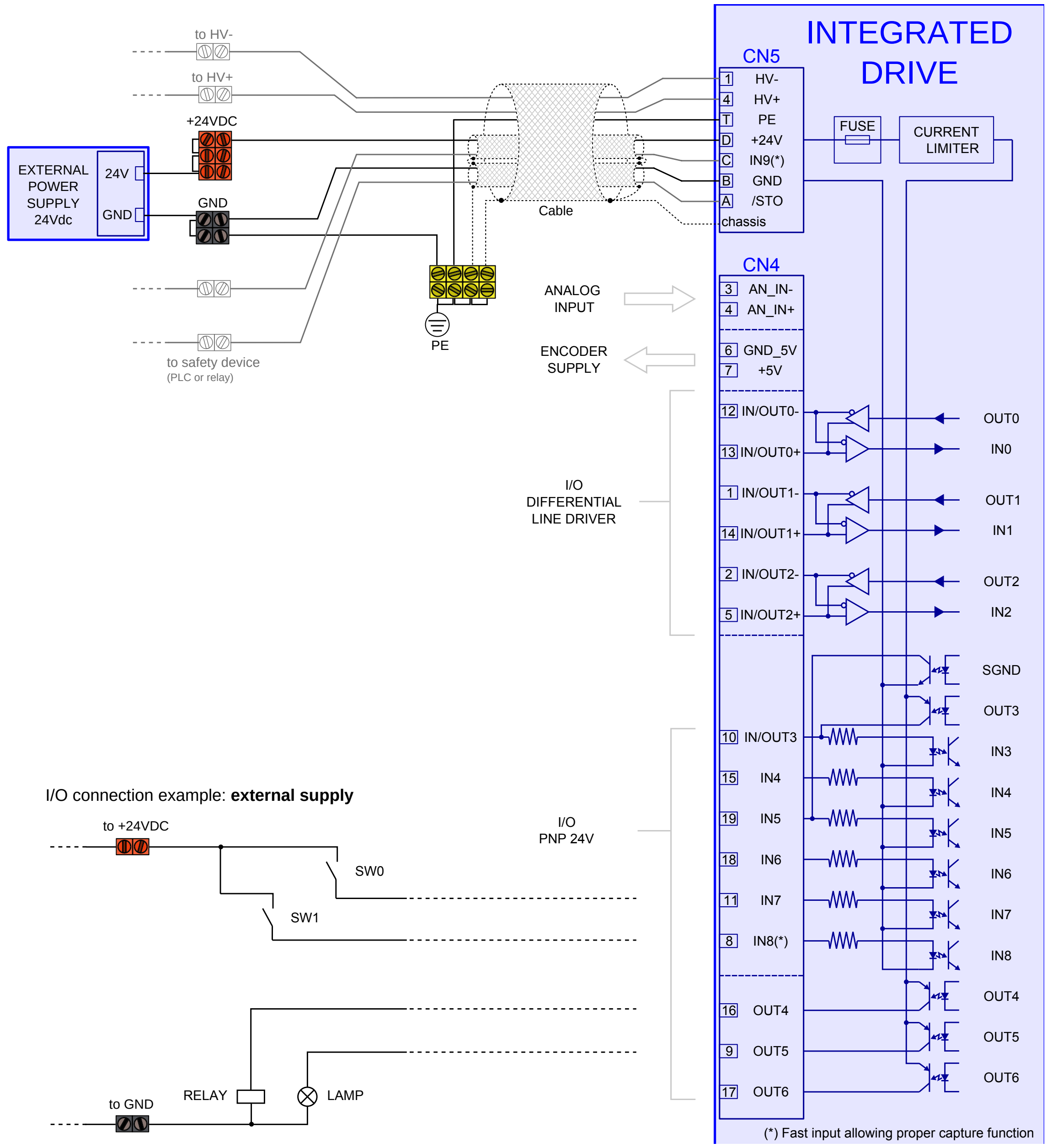



*Figure 7.2. Example of inputs and outputs wiring with external supply.*

It's possible to make these connections directly on the *i*BMD system (as reported on the Figure 7.3 example). Through the software settings it's possible to configure the IN5 input (pin 19 of CN4) as “SIMULATED GROUND”. In this case the IN5 can't be no more used as input because it is, internally to the system, connected to the GND (the same ground of pin B of CN5). This pin can be used to connect the output ground references. In the same way on one (or more) outputs it's possible to configure through software the “SIMULATED 24V” functionality. In this case the configured as described output can't be no more used as output because it is, internally to the system, connected to the 24V (the same 24V of pin D of CN5). The pin that's related to this output can be used to provide the supply 24V.

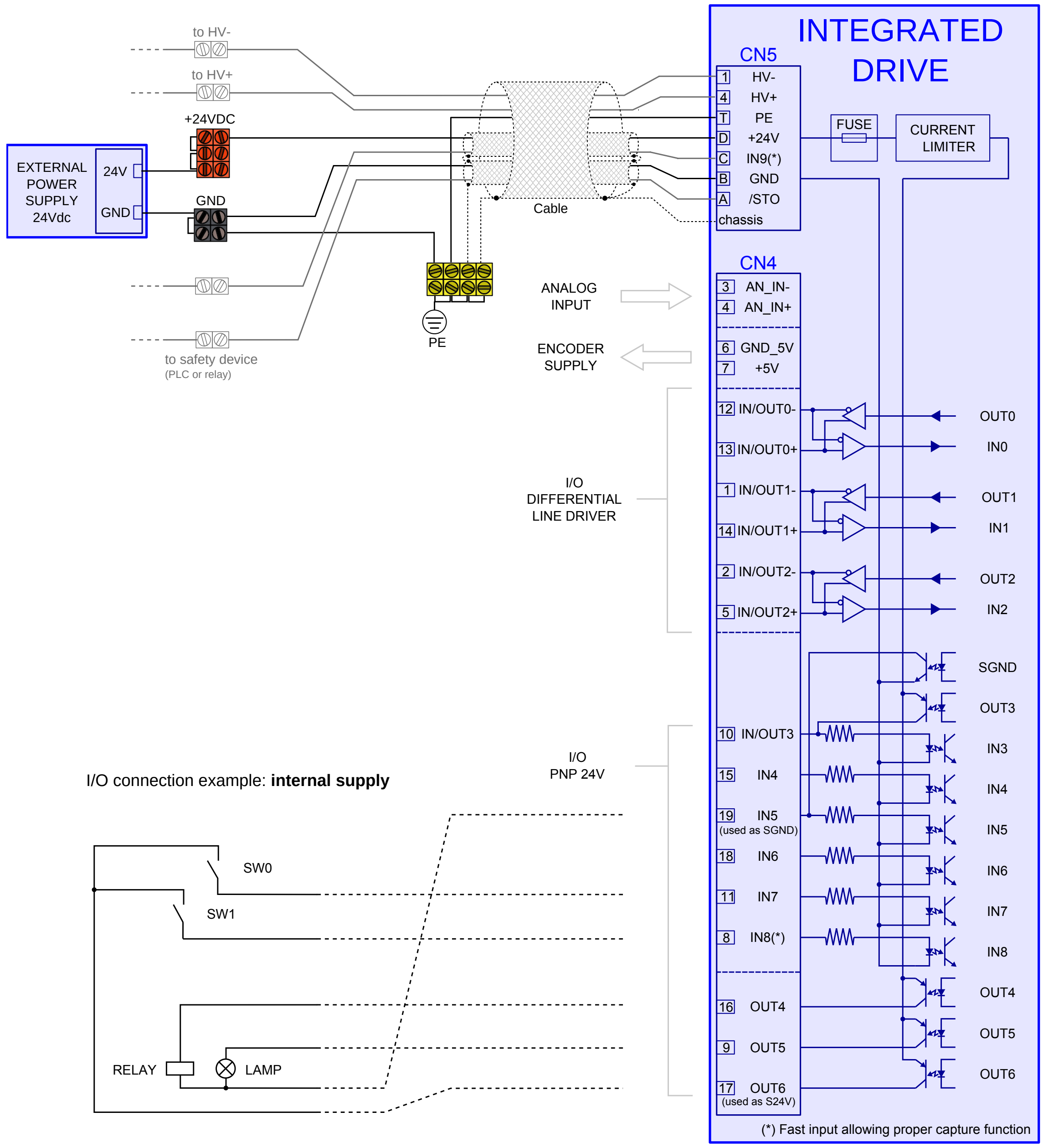



*Figure 7.3. Example of inputs and outputs wiring with internal supply.*

The current limits of the pins that have been used as SGND ("SIMULATED GROUND") and S24V (“SIMULATED 24V”) are reported on the Table 16.5. On the S24V configured pins a protection for the overcurrent or short circuit is present. The pin 19 of CN4 (IN5) configured as SGND is not protected from the overcurrent. Is therefore recommended to respect the maximum declared current absorption. If a greater absorption is needed it' is necessary to connect the ground externally from the *i***BMD**, as showed in Figure 7.2.

**Generic differential IN/OUT**

The IN/OUT0, IN/OUT1, IN/OUT2 inputs (line-driver differentials), can be even used as normal digital inputs (generic input functionality) The voltage levels are not 24V as for the PNP inputs, but they are referred to the line-drive specific levels that are described in the Table 16.3.

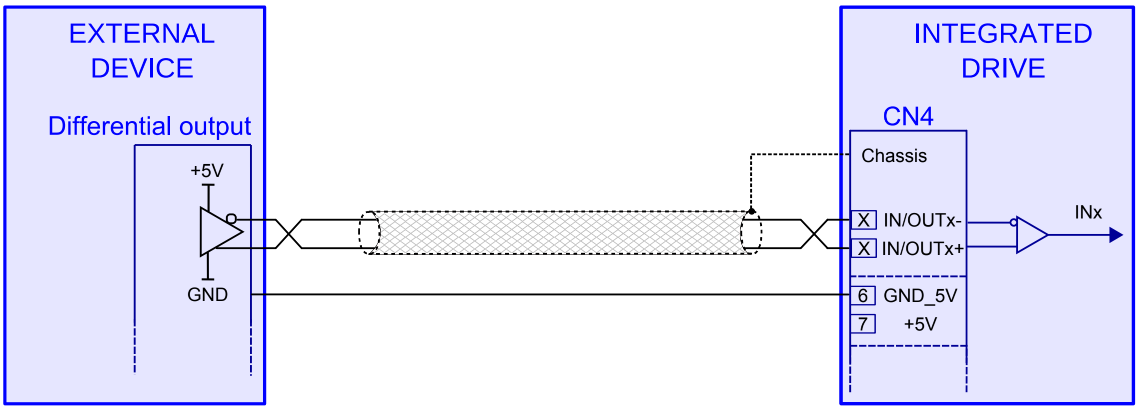



*Figure 7.4. Example of IN/OUT wiring with generic input functionality.*

⚠ NOTICE

**The differential IN/OUT, even if used with generic input functionality, are differential line-drive type. Do not connect signals with 24V levels! Please refer to the electrical features described in the Table 16.3.**

**Master Encoder Input (differential IN/OUT)**

The IN/OUT0, IN/OUT1, IN/OUT2 inputs (line-driver differentials), as well as with generic input functionality, can be used as incremental encoder inputs (phase A and phase B): to select the functionality please refer to Chapter 16, *Digital inputs and outputs*. If used as encoder inputs, IN/OUT0 and IN/OUT1 must be respectively connected to the encoder phase A and phase B and IN/OUT2 can be eventually used for the Index connection. They can be used incremental encoders whose supply can be provided externally or directly by the drive. For this purpose on the CN4 connector of the *i***BMD** is available a 5V voltage (max 150mA). In case the encoder is externally powered, or a simulated encoder is used, in addition to the differential signals (phase A, phase B and eventually the Index) the encoder ground must be connected to the GND_5V signal of the *i***BMD** (pin 6 of CN4).

For the external encoder connection on CN4 please refer to the Figure 7.5 in which are showed both the wirings when the supply is provided by the *i***BMD** and when the supply is external.

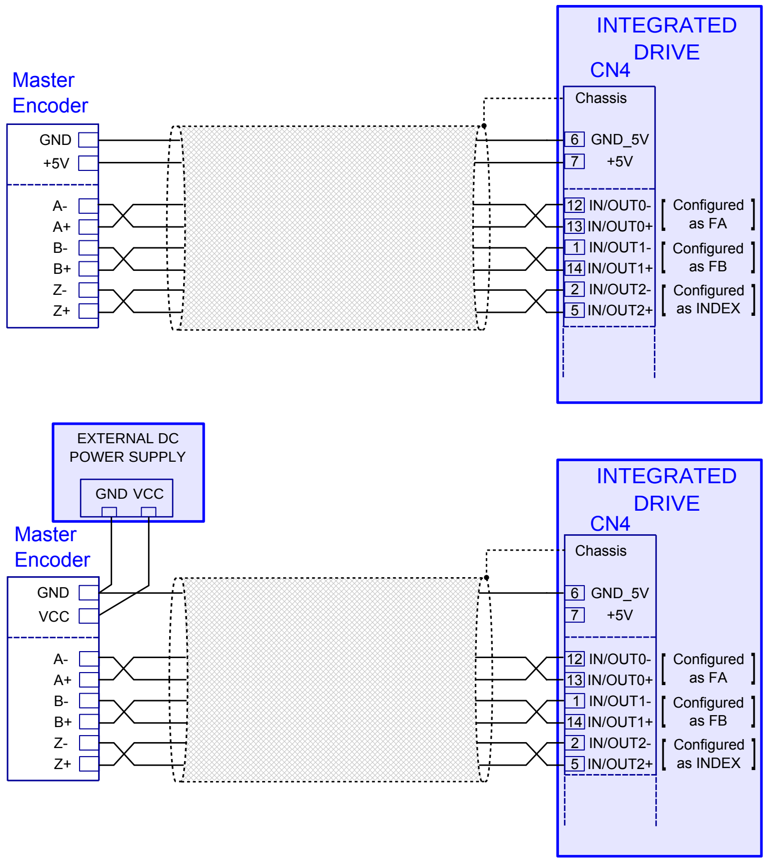



*Figure 7.5. Master encoder to differential IN/OUT wiring example.*

NOTE

**In this figure only one pair of twisted wires is showed. For informations about the cable see Table 7.4.**

⚠ NOTICE

**The encoder outputs must be differential line-driver, compatibles with the Table 16.3described characteristics. Do not connect signals with 24V levels!**

ADVICE

**It is suggested the usage of a shielded cable with twisted pairs to make the connection. To ensure the maximum noise immunity it can be used a double shielded cable (shield on each single twisted pair plus whole cable shield). It's suggested to connect the shield to the ground (connector chassis) only on the *i*BMD side. If possible the cable must not be interrupted. If the interruptions cannot be avoided, ensure that the shield is continuous and that the not shielded part has the minimum possible length.**

## Analog input

The drive has a differential analog input (CN4 connector: pin 3 and 4) to which different functionalities can be associated (see Section 17.3, *Conversion* and Section 22.16, *Profile Velocity AI Mode*).

ADVICE

**For the analog input configuration please see the Chapter 17, *Analog input*.**
**For the analog input electric characteristics see the Table 17.2.**

ATTENTION

**The maximum common-mode voltage of the differential analog input must not exceed the value that is reported in Table 17.2. For that reason it's recommended to refer the analog device supply ground to the GND_5V signal [pin 6 of CN4], as reported in the Figure 7.6 connection diagram.**

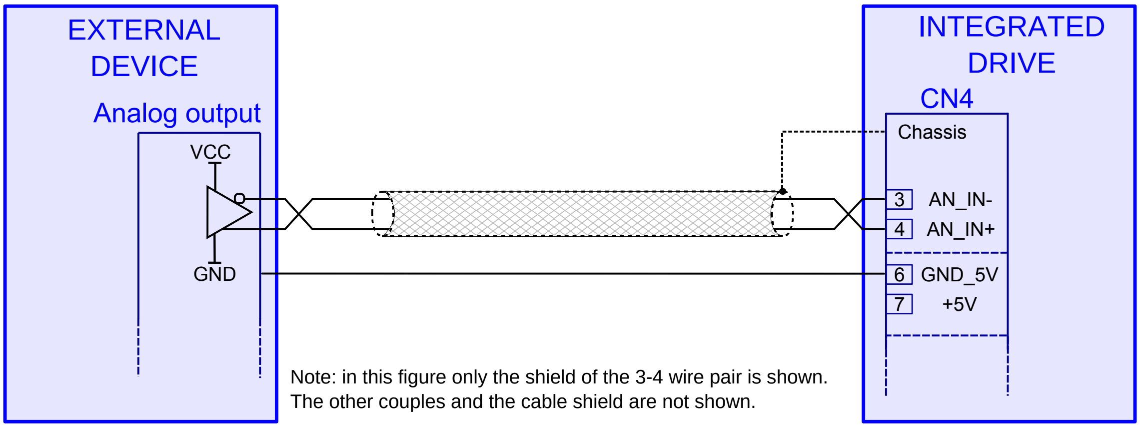



*Figure 7.6. Analog input wiring example (on CN4).*

NOTE

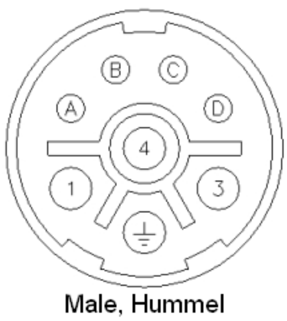
**In this figure only one pair of twisted wires is showed, relative to the pins 3 and 4. For informations about the cable see Table 7.4.**

ADVICE

**It is suggested the usage of a shielded cable with twisted pairs to make the connection. To ensure the maximum noise immunity it can be used a double shielded cable (shield on each single twisted pair plus whole cable shield). It's suggested to connect the shield to the ground (connector chassis) only on the *i*BMD side. If possible the cable must not be interrupted. If the interruptions cannot be avoided, ensure that the shield is continuous and that the not shielded part has the minimum possible length.**

## CN5 DC Power and Control supply, /STO, IN9

Connector for the supply of the power section and of the control section, plus two digital inputs /STO and IN9, M23 male, 8 poles (4 + 3 + PE), Hummel.



| PIN | Signal | Description |
|---|---|---|
| 1 | HV - | DC Power supply (negative pole) |
| 3 | - | Not connected |
| 4 | HV + | DC Power supply (positive pole) |
| T | PE | Protection Earth |
| A | /STO | Safe Torque Off Input (this is an active-low logic signal) |
| B | GND | Ground Control supply |
| C | IN9 | Digital Input 9 |
| D | +24 V | +24Vdc Control supply |
| Chassis | | |

### /STO input connection examples

The external connection of the /STO input must be protected and this can be obtained by isolating the connection or by using more simply a shielded connection with the shield connected to PE. Alternatively the /STO input can be driven by an output of a safe PLC with a test pulse duration of 1 ms.

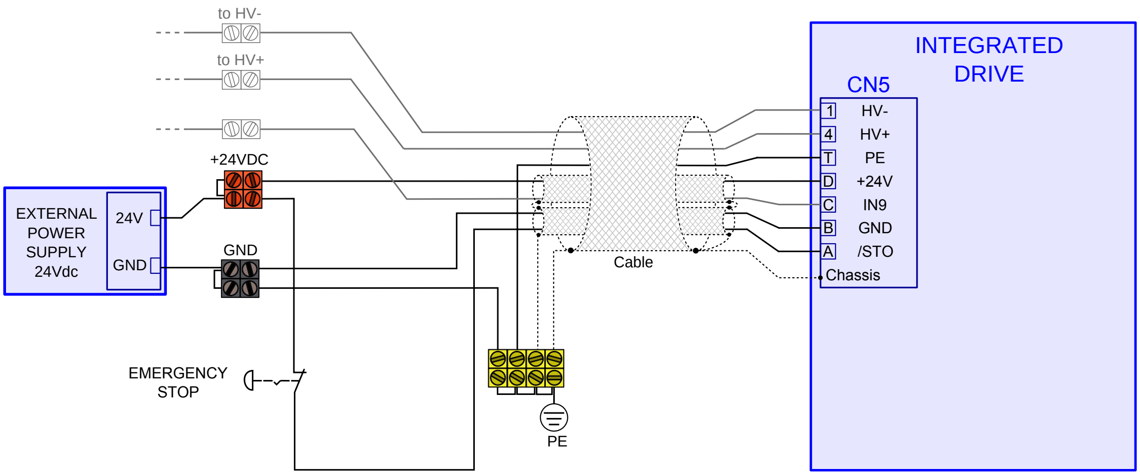



*Figure 7.7. Example of connection of the /STO input for stop of category 0 without module fuse relay: stop not controlled.*

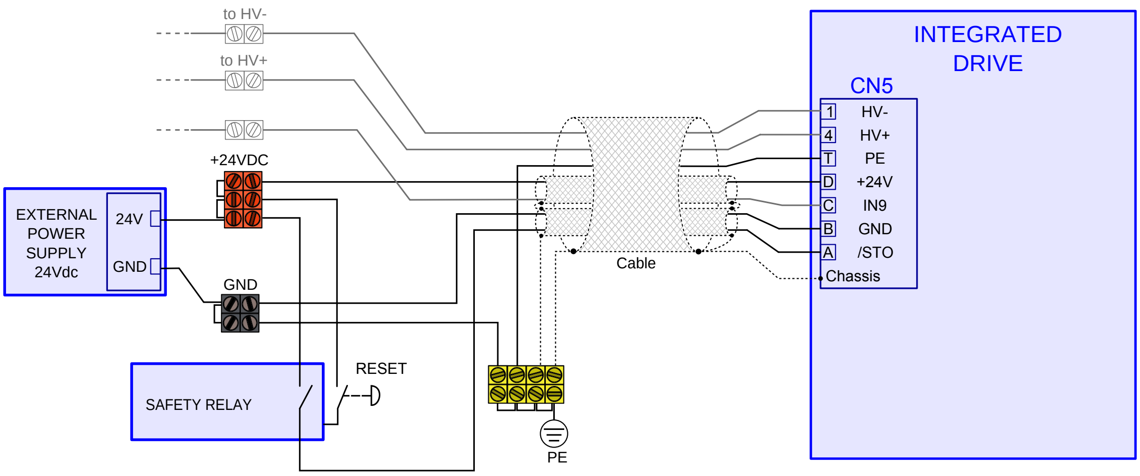



*Figure 7.8. Example of connection of the /STO input for stop of category 0 with fuse relay and interlock at the reboot: stop not controlled.*

To obtain a stop of category 1 connect the /STO input as shown in the following picture. The request for emergency stop initially causes the disabling of the digital input of Enable. This causes a motor stop according to the settings in the parameter DisableOption. After having programmed a delay on the fuse relay, the voltage on the input /STO is cut off and the power section is deactivated. The programmed delay must be enough to stop the motor, otherwise the final part of the motion becomes uncontrolled.

> IMPORTANT
>
> **The digital inputs to which it is possible to associate the functionality of Enable, are not safe inputs and consequently the deceleration is not safe.**

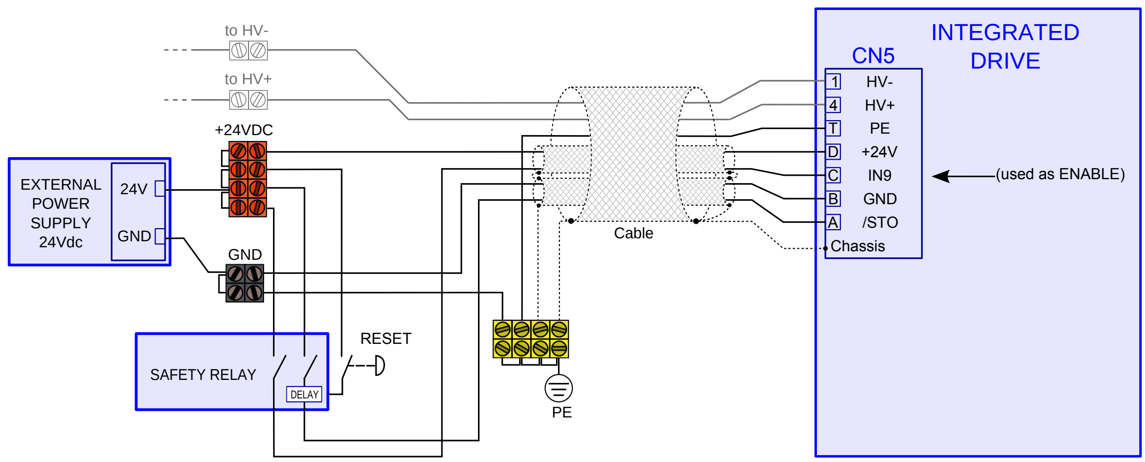



*Figure 7.9. Example of connection of the /STO input for stop of category 1 with fuse relay and interlock at the reboot. The functionality of Enable is associated to the input N9.*

## 7.2.5. Cables for *i*BMD

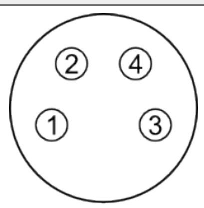
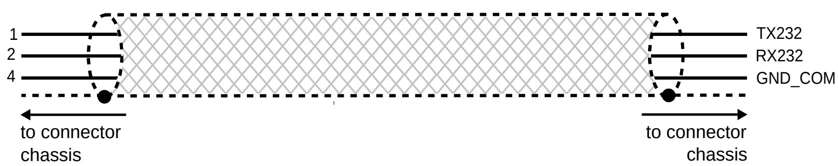
| Connector | Cable |
|---|---|
| Male<br>Flying connector on *i*BMD side, connection view | *i*BMD side host side<br> |
| **Example**:<br>*Productor*: Binder<br>*Connector:* 9933630004 | • Total shielding<br>• 3 or 4 wires<br>• maximum total length 5m |

*Table 7.1. Specifications for the RS232 serial cable (CN1).*

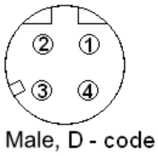
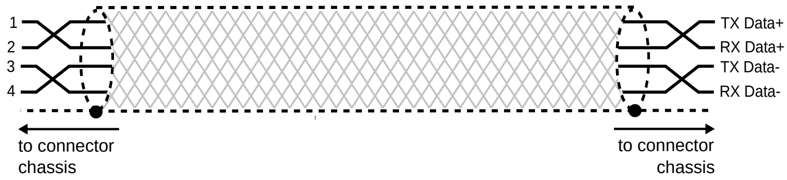
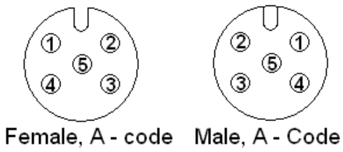
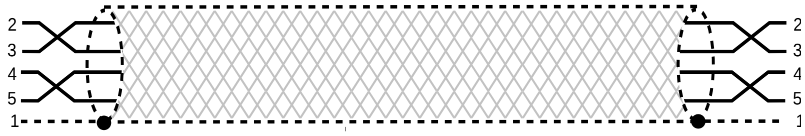
| Connector | Cable | |
|---|---|---|
|   Flying connector on *i*BMD side, connection view | *i*BMD side master side   | ETC |
|   Flying connector on *i*BMD side, connection view |   | CAN |
| **Example**:<br>*Connector:* M12, D-code (ETC) / M12, A-code (CAN) | • Total shielding (mixed: film and sock)<br>• 2 twisted pairs<br>• maximum total length: 30m (CAN) / 100m (ETC)<br>• Conductive material: tinned copper<br>Added requirements for the CAN cable<br>• Characteristic impedance $Z_0$: 120 Ω (100 Ω ≤ $Z_0$ ≤ 150 Ω)<br>• Net termination: Resistor 120 Ω, 1/4 W (equal to $Z_0$) | |

***Table 7.2. Specifications for the Main bus cable for the iBMD - master connection (CN2 e CN3 and CN2 e CN3)***

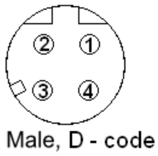
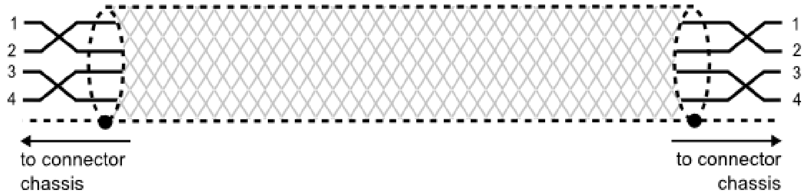
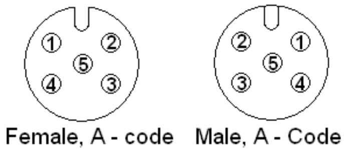
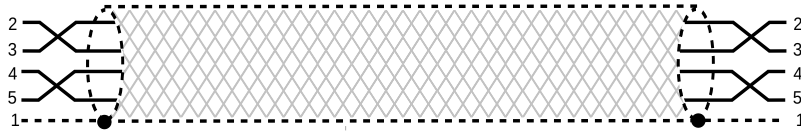
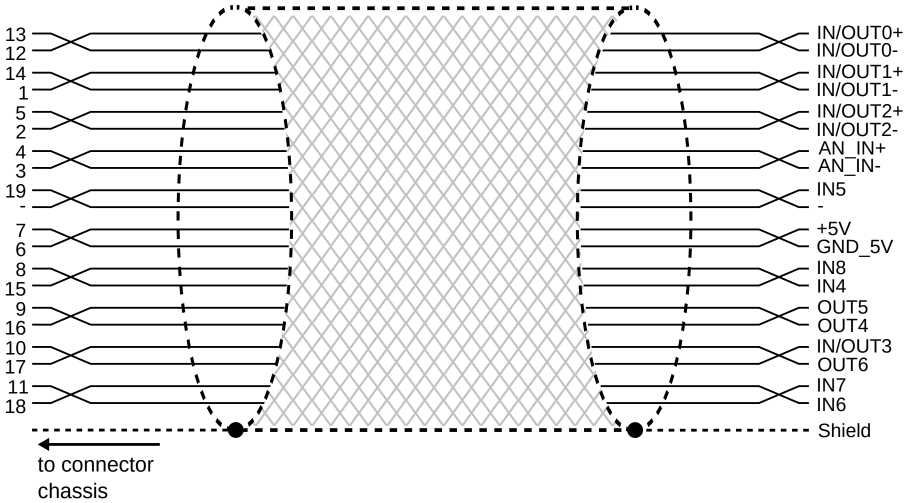
<table>
<tr><th>Connector</th><th colspan="2">Cable</th></tr>
<tr><td><br>Flying connector on both sides, connection view</td><td>iBMD side     iBMD side<br></td><td>ETC</td></tr>
<tr><td><br>Flying connector on both sides, connection view</td><td></td><td>CAN</td></tr>
<tr><td>Example:<br>Connector: M12, D-code (ETC) / M12, A-code (CAN)</td><td colspan="2"><ul><li>Total shielding (mixed: film and sock)</li><li>2 twisted pairs</li><li>maximum total length: 30m (CAN) / 100m (ETC)</li><li>Conductive material: tinned copper</li></ul>Added requirements for the CAN cable<ul><li>Characteristic impedance $Z_0$: 120 Ω (100 Ω ≤ $Z_0$ ≤ 150 Ω)</li><li>Net termination: Resistor 120 Ω, 1/4 W (equal to $Z_0$)</li></ul></td></tr>
</table>

***Table 7.3. Specifications for the Main bus cable for the*** *i*BMD - *i*BMD ***(for CAN version CN2 e CN3 and for ETC version CN2 e CN3)***

| Connector | Cable |
|---|---|
|  <br>Flying connector on *i*BMD side, connection view | *i*BMD side Command devices or actuators side<br>  |
| **Example**:<br>*Productor*: Hummel<br>*Connector:* 7.106.500.000<br>*Insert*: 7.003.919.102<br>*Contacts*: (3x, pin 12, 19, 6) 7.010.901.512; (16x) 7.010.901.012 | • Total shielding<br>• 10 twisted pairs (as an alternative 11 fili + 4 twisted pairs[a])<br>• maximum total length 30m |

[a]The wires pairs that must be twisted are the ones that correspond to the differential IN/OUT and to the analog input.

***Table 7.4. Specifications for the Input / Output cable (CN4).***

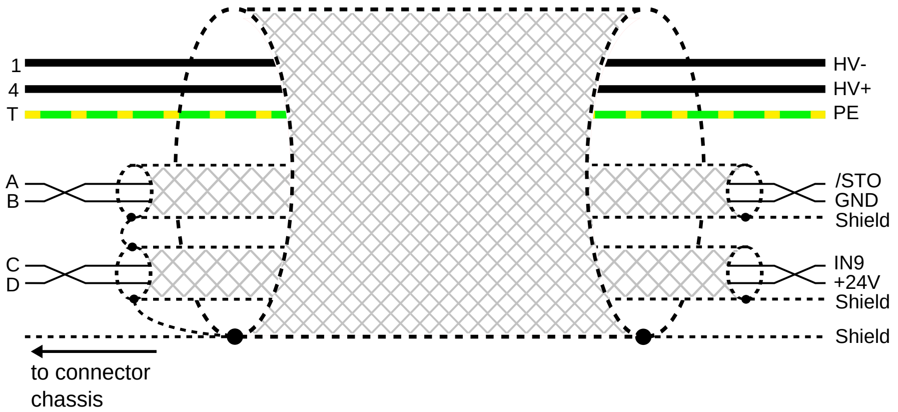
| Connector | Cable |
|---|---|
|  <br>Flying connector on *i*BMD side, connection view | *i*BMD side supplier side<br>  |
| **Example**:<br>*Productor*: Hummel<br>*Connector:* 7.106.500.000<br>*Insert*: 7.084.943.102<br>*Contacts*: (3x, pin 1, 4, T) 7.010.942.002, (4x) 7.010.941.002 | • Total shielding<br>• 2 wires + PE + 2 shielded and twisted pairs<br>• low capacity (example: UNIKA UNIDRALL 5005 cod. 3J049)<br>• maximum total length 30m |

***Table 7.5. Specifications for the DC bus, control, STO and IN9 cable (CN5)***

## 7.3. Leds

The leds can have the following statuses:

- *OFF*: led switched off;
- *ON*: fixed led switched on;
- *BLK* (blinking): led 200 ms on, 200 ms off;
- *1 FL* (1 flash): led 200 ms on, 1 s off;
- *2 FL* (2 flash): led 200 ms on, 200 ms off, 200 ms on, 1 s off;
- *3 FL* (3 flash): led 200 ms on, 200 ms off, 200 ms on, 200 ms off, 200 ms on, 1 s off;
- *FLK* (flicker): led 50 ms on, 50 ms off.

The notifications meaning, shown through the leds, can be found in the link in the following table:

IMPORTANT

**In the boot and firmware start-up phases of the description of the six leds L1-L6 is not the one indicated below. Please refer to Chapter 26, *Software updating***

| Leds | Description | Link |
|---|---|---|
| L1, L2 | Drive status (Fault, Warning, enabling) | Table 8.10 |
| L3, L5 | Reserved (led off) | - |
| L4 | Limitation status I2T | Table 14.2 |
| L6 | STO logic status | *i*BMD: Led 6 diagnostics |
| LA | CANopen error led (ERR) | Table 8.3 |
| LB | CANopen run led (RUN) | Table 8.4 |
| L/A 0 | Status of the Physical link/activity of the EtherCAT port on the CN3 connector (for *i*BMD). | Table 8.6 |
| L/A 1 | Status of the Physical link/activity of the EtherCAT port on the CN2 connector (for *i*BMD). | |
| ERR | EtherCAT error led (ERR) | Table 8.7 |
| RUN | EtherCAT run led (RUN) | Table 8.8 |

***Table 7.6. Leds description.***

### 7.3.1. Leds arrangement in the *i*BMD

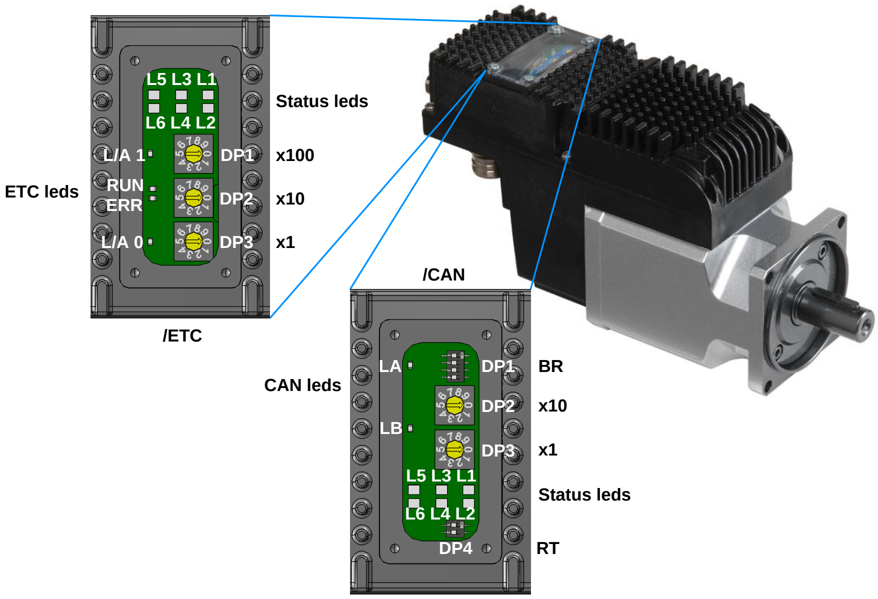



*Figure 7.10. Leds and dipswitches arrangement in the transparent window of the i*BMD*.*

## 7.4. Dip switches

### 7.4.1. Arrangement and characteristics of the dipswitches of the *i*BMD

The dip switches arrangement for the *i*BMD drives is shown in Figure 7.10. The parameters defined by the dip switches can be found in the following charts.

| Dip sw | Description | Link |
|---|---|---|
| DP1 | EtherCAT user address (station alias) x100 | Configured station alias |
| DP2 | EtherCAT user address (station alias) x10 | |
| DP3 | EtherCAT user address (station alias) x1 | |

***Table 7.7. Description of the dip switches for i*BMD/ETC drives.***

| Dip sw | Description | Link |
|---|---|---|
| DP1 | Baud rate CANopen | Baud rate through dip switch |
| DP2 | Node identification x10 | Node number through dip switch |
| DP3 | Node identification x1 | |
| DP4 | Termination of the CANopen network | Table 8.2 |

***Table 7.8. Description of the dip switches for i*BMD/CAN drives.***

In the *i*BMD/CAN system the node number setting (Node Identification) is possible through DP3 (marked with “x1” and used to set the units) and DP2 (marked with “x10” and used for the tens). The factory default has all the switches on the 0 position. Through the dip switches it's possible to set a valid node number from 1 to 99 (in decimal format).

The communication speed (baud rate) in the CANopen network of the *i*BMD system can be set through DP1. The maximum settable baud rate is, in general, function of the network length and of the peripherals number that are connected on the network. It is furthermore possible to configure this velocity via software (for further informations see Access with MotionDrive from **Tab Bus**.).

To insert the termination resistance in the *i*BMD/CAN drives use the DP4.

NOTE

**Remember that the termination resistance has to be inserted on both the extremities of the CAN network and also that the value of the resistance, if measured, is about 60 Ω when the network is offline.**

# Chapter 8
# Communicating with the drive

WARNING

**Connect and disconnect the communication connectors only when the drive is switched off. Check if the pin Ground Control supply of CN5, the drive, the master, the PC and all devices are correctly connected to the protective conductor.**

## 8.1. Communicate with master CANopen

NOTE

**The information in this paragraph are valid only for the drives version CAN. The details on the protocol implementation are described in Chapter 9, *Communication port CANopen*.**

Connect the cables of the network CANopen a CN2 e CN3. For further information see Section 7.2, *Electrical connections*.

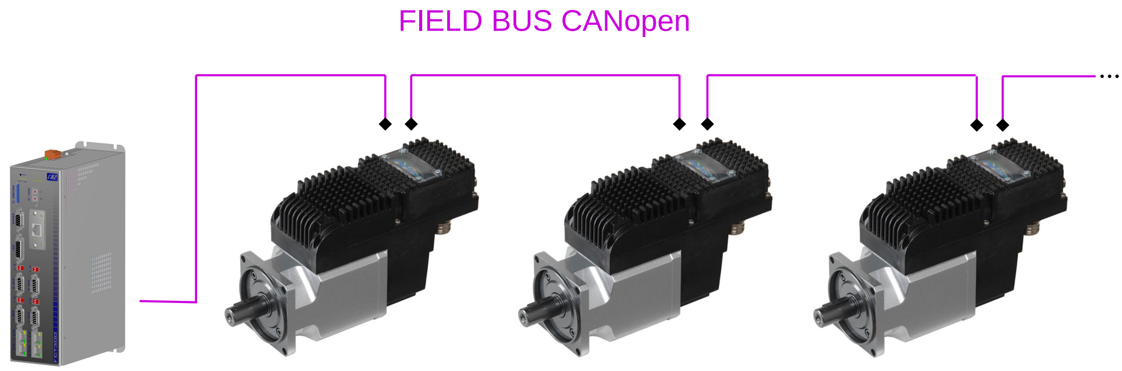



*Figure 8.1. Communication scheme with CANopen Master.*

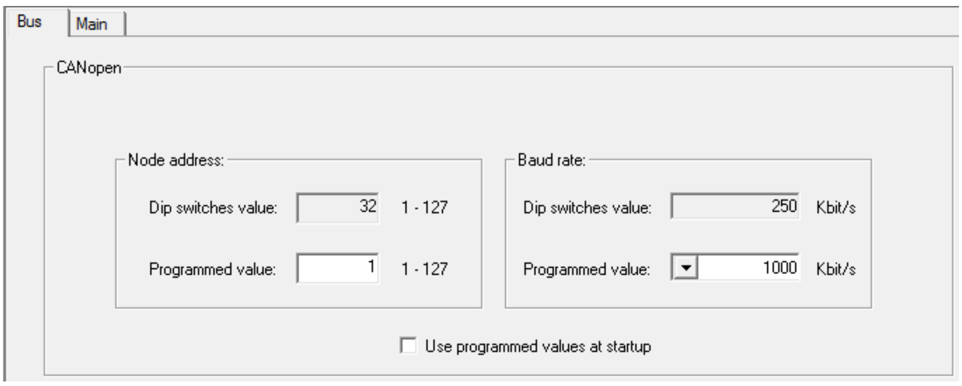
Configure the communication port CANopen by setting the node number and the baud rate (communication speed). The node number and the baud rate are set by dip switch (Figure 7.10) or by writing the parameters CANopenPortSetup: the choice of the configuration to use is made by using the CANopenPortSwitchSetup parameter.

NOTE

**Any modifications to the configuration of the CANopen port (dip switch, CANopenPortSwitchSetup and CANopenPortSetup) are applied only after a drive reset (see Enabled parameters after reset).**

**Access with MotionDrive from Tab Bus.**

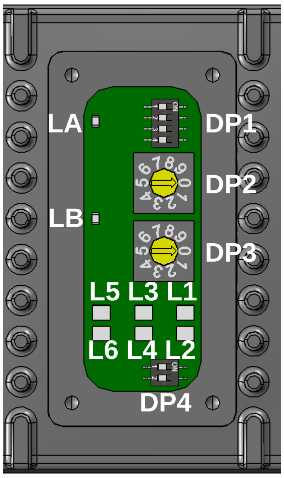
The node number and the Baud Rate of the drive can be assigned via software through the *Programmed value* fields as an alternative to the configuration through physical dip switches. To enable the configuration through *Programmed value* fields you have to check the *Use programmed value at start-up* box. In this way the dip switches will be ignored.




NOTE

**Therefore the node number setting has to be done via dip switches or via software. If the node number that is set with the dip-switches is 0, and the box "Use programmed value at startup" is not checked, it will be used the last saved configuration in the permanent memory, related to the CANopenPortSetup parameters.**

### Configuration from dip switch




With the dip switches installed in the drive it is possible to configure:

- the node number, through DP2 and DP3, according to the formula:
  Node number = DP3 + DP2 x 10;
- The baud rate, through DP1, as shown in the following table:

| Switch 4 | Switch 3 | Switch 2 | Switch 1 | Baud rate [bit/s] |
|---|---|---|---|---|
| OFF | OFF | OFF | OFF | 1000000 |
| OFF | OFF | OFF | ON | 500000 |
| OFF | OFF | ON | OFF | 250000 |
| OFF | OFF | ON | ON | 125000 |
| OFF | ON | OFF | OFF | 100000 |
| OFF | ON | OFF | ON | 50000 |
| OFF | ON | ON | OFF | 20000 |
| OFF | ON | ON | ON | Reserved |
| ON | X | X | X | |

***Table 8.1. Selection of the baud rate of the port CANopen through DP1.***

### Termination resistance

If the drive is the last Node of the network CANopen it is necessary to insert the termination resistance through the dip switch DP4.

| Switch 2 | Switch 1 | Termination resistance |
|---|---|---|
| OFF | OFF | Not inserted |
| OFF | ON | Configuration not allowed |
| ON | OFF | Configuration not allowed |
| ON | ON | Inserted |

***Table 8.2. Setting the termination resistance of the port CANopen through dip switch DP4.***

NOTE

**Remember that the termination resistance has to be inserted on both the extremities of the CAN network and also that the value of the resistance, if measured, is about 60 Ω when the network is offline.**

To configure and map the PDOs, see Section 9.5, *Process data object (PDO)*.

To monitor any errors at frame level and any errors generated by the services, read the group of parameters CANopenCounters (types of detected errors and error frame counters) or enter in the **Tab Bus** of MotionDrive.


To check the emergences sent, see the parameters CANopenEmcyRegister.

To check the status of the NMT state machine, enable the Error control services.

**Status of the LA and LB leds**

Interpretation of the leds status:

LA: status of the errors detected by the CANopen port; the coding is reported in Table 8.3;
LB: status of the NMT state machine; the encoding is taken to Table 8.4.

| Errors of the port CANopen | Description | LA |
|---|---|---|
| No error | The port CANopen is working correctly. | OFF |
| Warning limit reached | At least one of the error counters (TEC or REC, see CANopenStatusRegister) reached or overcame the warning level at 96. | 1 FL |
| Error control event | The communication port has a Life Guard error. | 2 FL |
| Sync error | Error in the SYNC controller. | 3 FL |
| Bus-off | The communication port is in the Bus-off status. | ON |

***Table 8.3. Encoding of the status of the led LA (ERROR).***

| Status of the NMT state machine | LB |
|---|---|
| PRE-OPERATIONAL | BLK |
| STOPPED | 1 FL |
| OPERATIONAL | ON |

***Table 8.4. Encoding of the status of the led LB (RUN).***

# 8.2. Communicate with master EtherCAT

NOTE

**The information in this paragraph are valid only for the drive version ETC. The details on the protocol implementation are described in Chapter 10, *Communication port EtherCAT*.**

Connect the cables of the EtherCAT network to CN2 e CN3: connect the output cable on the first connector and the input cable on the second one. For further information see Section 7.2, *Electrical connections*.

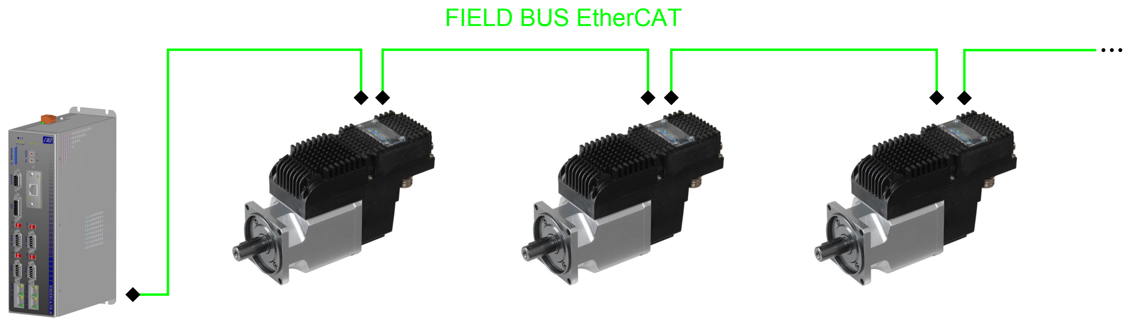



*Figure 8.2. Communication scheme with EtherCAT Master.*

Configure the communication port EtherCAT, by defining the node number. The baud rate (communication speed), according to its technical feature, is set to 100Mbit/s. To define the node number, the master can choose one of the following modes:

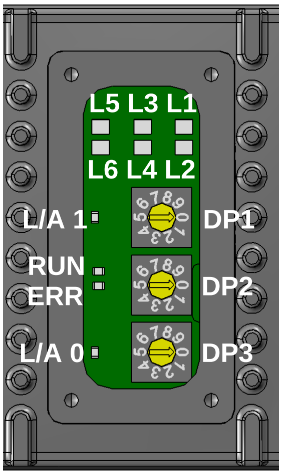
- *Positional (Position Address)*: this method is generally used by the masters for the automatic detection of the drives in the EtherCAT network; the master gives to every node an address which is coherent with the physical position owned by the drive in the network;

- *Fixed (Node Address)*: For the fixed addressing it's necessary that the master writes in the "Configured Station Address" register (address 0x0010-0x0011 of the ET1100 memory) the address with which it wants to identify, in univocal way, the drive. This address can be set before (according to an own algorithm in the master) or can be read from the "Configured Station Alias" register (address 0x0012-0x0013 of the ET1100 memory).
  The "Configured Station Alias" register has the following behaviour:
  - it takes the value of the rotative dip-switches (DP1, DP2 e DP3) if their total value is different from 0
  - it takes the value that's contained in the word address 0x0004 of the ESI eeprom if the rotative dip-switches total value is 0.

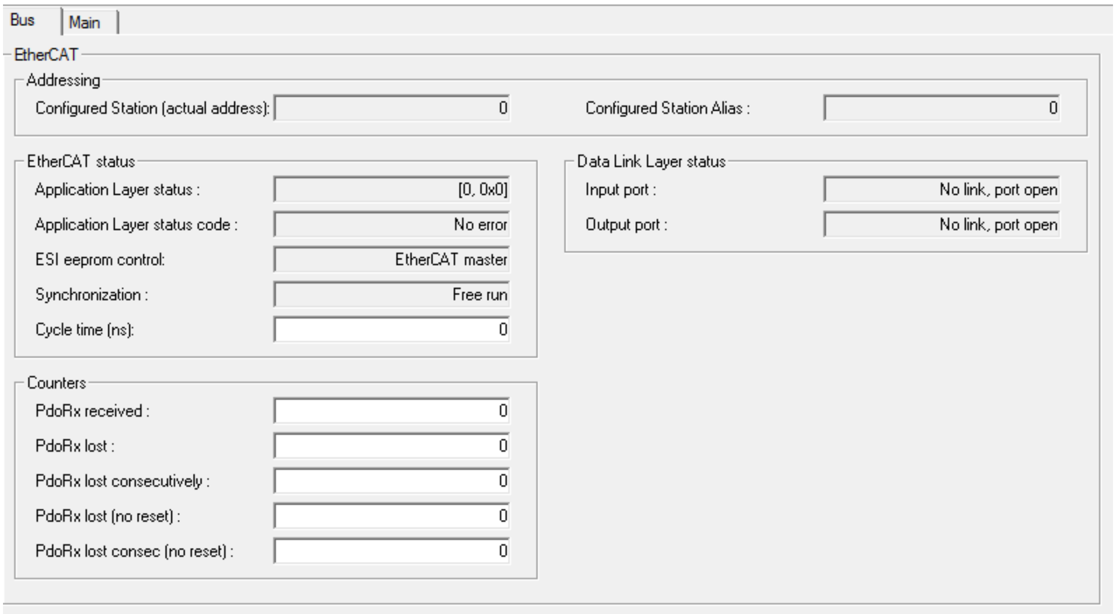
  *node number* = DP3 + DP2 x 10 + DP1 x 100 (refer to Figure 7.10 and to Section 7.4, *Dip switches*).

NOTE

**Every time the dip-switches and/or the "Configured Station Alias" value is changed, it's necessary to reset the drive (or to execute a turn-off/turn on sequence, see Enabled parameters after reset).**




**Access with MotionDrive from Tab Bus.**
In the *Bus address* field you can find the node number given to the drive while in the *Dip switches value* field you can find the node number calculated on the basis of the value of the dip switches.




| **Text in the page** | **(EtherCAT_Diagnostics parameter)** |
|---|---|
| Configured Station (actual address) | EtcConfiguredStation |
| Configured Station alias | EtcConfiguredStationAlias |
| Application Layer status | EtcRegAlStatus |
| Application Layer status code | EtcRegAlStatusCode |
| ESI eeprom control | EtcRegEEpromConfiguration |
| Synchronization | SM2_SynchronizationType |
| Cycle time (ns) | SM2_CycleTime |
| Pdo Rx received | EtcPdoRxTotal |
| Pdo Rx lost | EtcPdoRxLostTotalReset |

| Text in the page | (EtherCAT_Diagnostics parameter) |
|---|---|
| Pdo Rx lost consecutively | EtcResetPdoRxLostMaxConsecReset |
| Pdo Rx lost (no reset) | EtcPdoRxLostTotal |
| Pdo Rx lost consecutively (no reset) | EtcPdoRxLostConsecutive |
| Input port | EtcRegDllStatus, port A |
| Output port | EtcRegDllStatus, port B |

*Table 8.5. Correspondence to the EtherCAT_Diagnostics parameters of the "Bus" page.*

**Status of the L/A 0, L/A 1, ERR and RUN leds**
Interpretation of the leds status:

L/A 0 and L/A 1: link status and possible ongoing activity in the CN2 e CN3 physical ports; L/A 0 takes the status of the accessible input port from the connector CN3, L/A 1 takes the status of the accessible output port from the connector CN2; the encoding of the two leds is taken to Table 8.6;

ERR: error status found by the port EtherCAT; the encoding is taken to Table 8.7;

RUN: status of the EtherCAT state machine; the encoding is taken to Table 8.8.

| Link of the physical port | Activity of the physical port | L/A 0 and L/A 1 |
|---|---|---|
| No connection | - | OFF |
| Connected | No message | ON |
| Connected | Communication enabled | FLK |

*Table 8.6. Encoding of the leds status L/A 0 and L/A 1.*

| Errors of the port EtherCAT | Description | ERR |
|---|---|---|
| No error | The port EtherCAT is working correctly. | OFF |
| Configuration not valid | Wrong settings of the communication port EtherCAT: the change of the state of the EtherCAT state machine requested by the master is not possible. | BLK |
| Change of status not requested | The drive has automatically changed the state of the EtherCAT state machine without any command by the master. This solution is generally chosen when there is an error in the synchronization. | 1 FL |
| Sync Manager watchdog expired | The watchdog of the Sync manager (SM) of the PDO RX has expired. | 2 FL |
| Hardware failure | Serious error in the ET1100; please contact Bonfiglioli Mechatronic Research S.p.a | ON |

*Table 8.7. Encoding of the leds status ERR.*

| Status of the EtherCAT state machine | RUN |
|---|---|
| INIT | OFF |
| PRE-OPERATIONAL | BLK |
| SAFE-OPERATIONAL | 1 FL |
| OPERATIONAL | ON |
| BOOTSTRAP | FLK |

*Table 8.8. Encoding of the leds status RUN.*

### 8.2.1. File access over EtherCAT (FoE) protocol

The FoE protocol is implemented in the drive, but its use is limited only to update the firmware.

## 8.3. Communicate with master Modbus RS232 (auxiliary communication port)

Connect the serial cable RS232 of the master to CN1. For further information see Section 7.2, *Electrical connections*.

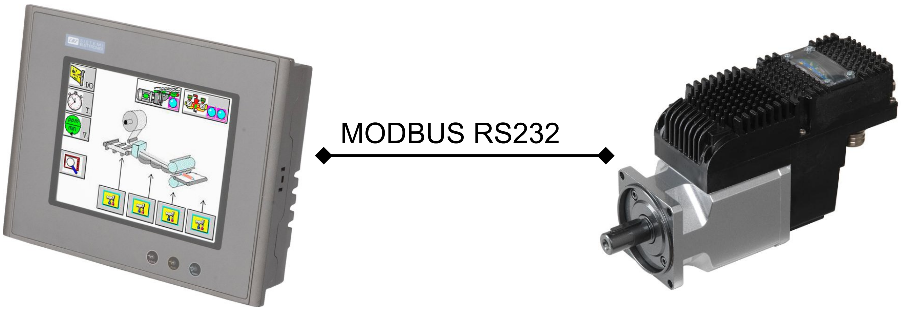



***Figure 8.3. Point-to-point communication scheme with master.***

Set the features of the serial as follows:

- Character length: 8 bits
- Type of parity: even
- Number of stop bits: 1 bit.
- Default baud rate: 57600bit/s.

The drive answers on the auxiliary communication port with node-ID equal to 1. The other features of the port are configured with the parameters of the group AuxiliaryPortSetup. The details on the protocol implementation are described in Chapter 11, *Auxiliary communication port Modbus*.

NOTE

**To analyse the latest communication error of the protocol Modbus, read the parameters of the group AuxiliaryPortError.**

## 8.4. Errors in reading / writing of the parameters

When there is an error in the reading or writing of the parameter, in order to understand which problem occurred it is necessary to get the error code:

- auxiliary communication port: you can find the error code of the last failed access in AuxiliaryPortErrorCode;
- main communication port CANopen and EtherCAT: the error code is contained in the frame **SDO abort**.

| SDO abort code | AuxiliaryPortErrorCode | Error | Description |
|---|---|---|---|
| 0x0 | 0x00 | No error | No error. |
| - | 0x01 | Modbus protocol error: illegal function | Code function Modbus not supported. In Table 11.1 you can see the accepted codes. |
| - | 0x02 | Modbus protocol error: address not existent | Address not existing: the combination of the Modbus address and the data to write/read is not valid; the addresses included in the requested range must be contained in the vocabulary of the parameters. |
| - | 0x03 | Modbus protocol error: data dimension too large | Quantity of data not admitted: too large or equal to 0. |
| - | 0x10 | Modbus protocol error: illegal upload/download code | Upload/download code not valid. |
| - | 0x11 | Modbus protocol error: unexpected upload/download state | Upload/download status unexpected. |
| - | 0x12 | Modbus protocol error initializing upload/download | Wrong initialization of the upload/download. |
| - | 0x13 | Modbus protocol error during upload/download | Error during data upload/download. |
| - | 0x14 | Modbus protocol error closing upload/download | Error during upload/download closing. |
| - | 0x15 | Modbus protocol error: memory overflow during upload/download | Memory overflow to complete upload/download. |
| 0x05030000 | 0x16 | Unexpected toggle bit | Toggle bit not alternated during upload/download. |
| 0x05040001 | - | Client / server command specifier not valid or unknown | Command specifier of the frame SDO not valid. |
| 0x05040005 | 0x20 | Memory not available | Memory overflow to run the requested operation. |
| 0x06010000 | 0x21 | Access denied | Access denied to the parameter. |
| 0x06010001 | 0x22 | Attempt to read a write only object | Reading failed, parameter only for writing. |
| 0x06010002 | 0x23 | Attempt to write a read only object | Writing failed, parameter only for reading. |
| 0x06040043 | 0x24 | General parameter incompatibility | General incompatibility of the datum. |
| 0x06040047 | 0x25 | General internal incompatibility | General internal incompatibility of the drive. |
| 0x06060000 | 0x26 | Hardware error | Access failed because of a hardware error. |
| 0x06070010 | - | Data type does not match | Datum dimension not correct. |
| 0x06090011 | 0x27 | SubIndex not existing | CANopen or EtherCAT sub-index not existing. |
| 0x06090030 | 0x28 | Parameter out of range | Parameter value out of range accepted. |
| 0x08000000 | 0x29 | Generic error | Generic error. |
| 0x08000021 | 0x2A | Internal control refuse data | Access denied because of a local control. |
| 0x08000022 | 0x2B | Internal state refuse data | Access denied because of the drive current status. |
| 0x06020000 | 0x2C | Object does not exist | CANopen or EtherCAT index not existing. |
| 0x06040041 | 0x2D | Object not mappable on PDO | Parameter not mappable in the PDOs. |

| SDO abort code | AuxiliaryPortErrorCode | Error | Description |
|---|---|---|---|
| 0x06040042 | 0x2E | Length of object mapped exceed PDO length | The dimension of the mapped parameter in the PDOs is too large. |

*Table 8.9. Encoding the errors of reading/writing of the parameters.*

## 8.5. CiA402 state machine

The drive of the *i*BMD series, follow the CiA-402 specification. For the drive management, the CiA-402 specification needs the implementation of a state machine, which scheme can be found in the following picture. The drives of the *i*BMD series follow the CiA-402 specifications.

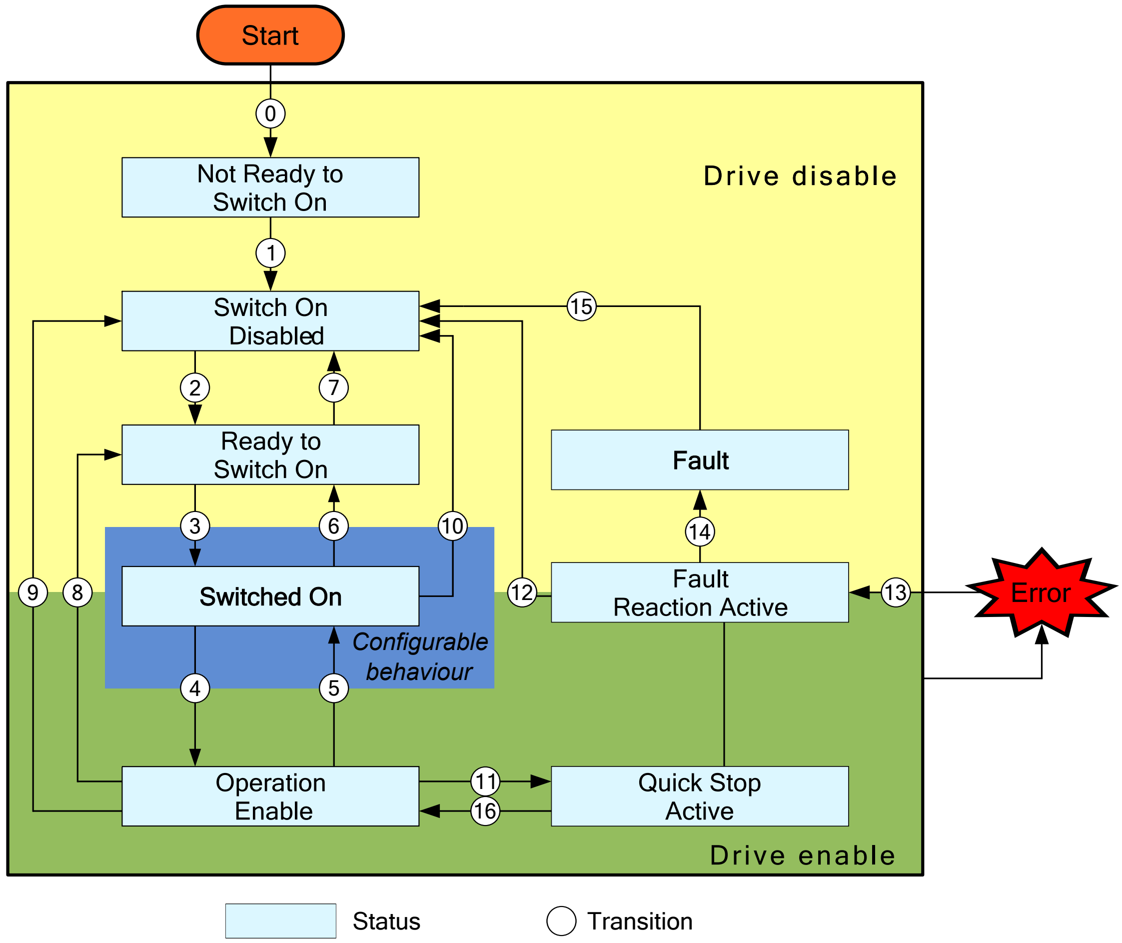
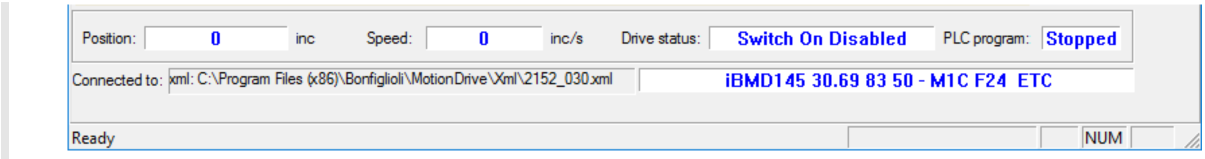



*Figure 8.4. CiA402 state machine implemented in the iBMD drives.*

To enable or disable the drive and the motor motion, to stop and reset any error, it's necessary to ask for the right transitions to the CiA402 state machine so that it can reach the desired state. The Statusword parameter reports the CiA402 state machine status.

Access with MotionDrive:

**Tab Main** > **Drive status**




The CiA402 state machine status is partly also shown with the L1 and L2 leds, according to the following chart. The encoding of the errors can be found in Section 24.2, *Monitoring the errors on the status leds*.

IMPORTANT

**In the boot and firmware start up phases of the *i*BMD the below description of the leds is no more valid. Please refer to Chapter 26, *Software updating***

| Status of the drive | Led L1 drive *i*BMD | Led L2 drive *i*BMD |
|---|---|---|
| Drive enable no error | GREEN ON | GREEN ON |
| Drive disable no error | GREEN ON | BLINKING GREEN |
| There are some errors of Warning type and not of Fault type | ORANGE VARIOUS STATUS (see Table 24.2) | |
| There are some errors of Fault type | RED VARIOUS STATUS (see Table 24.2) | |

***Table 8.10. Status of the CiA402 state machine visualized with the L1 and L2 leds.***

In the following chart you can find all the possible states and their features. The bits shown with an 'x' are not important to determine the state.

| Statusword | State | Description | Drive enable | Operation enable | PLC running | Communication active |
|---|---|---|---|---|---|---|
| xxxx xxxx x0xx 0000 | *Not Ready to Switch On* | Initializing | - | - | - | - |
| xxxx xxxx x1xx 0000 | *Switch On Disabled* | Disabled drive | - | - | *YES* | *YES* |
| xxxx xxxx x01x 0001 | *Ready to Switch On* | Preparation to enabling | - | - | *YES* | *YES* |
| xxxx xxxx x01x 0011 | *Switched On* | The drive can be enabled or disabled, depending on the SwitchedOnOptionCode parameter value. | See Description. | - | *YES* | *YES* |
| xxxx xxxx x01x 0111 | Operation enable | Drive enabled and possibility to command the motor motion | *YES* | *YES* | *YES* | *YES* |
| xxxx xxxx x00x 0111 | *Quick Stop Active* | Running a stop *Quick stop* | *YES* | - | *YES* | *YES* |
| xxxx xxxx x0xx 1111 | *Fault Reaction Active* | Reaction to a Fault situation. The drive can be enabled or not, depending on the situation before the error occurred | See Description. | - | *YES* | *YES* |

| Statusword | State | Description | Drive enable | Operation enable | PLC running | Communication active |
|---|---|---|---|---|---|---|
| xxxx xxxx x0xx 1000 | *Fault* | Fault state, finished reaction | - | - | *YES* | *YES* |

***Table 8.11. States of the CiA402 state machine.***

In the following table you can find the description of the single bits of the Statusword parameter; some bits have a different meaning depending on the value of ModesOfOperationDisplay, indicated in the Mode column.

| Bit | Mode | Name | Description |
|---|---|---|---|
| 0 | All | *Ready to switch on* | Encoding the state of the CiA402 state machine (see Table 8.11). |
| 1 | | *Switched on* | |
| 2 | | *Operation enabled* | |
| 3 | | *Fault* | Bit which is set when a retentive Fault is detected (FaultRetentive) |
| 4 | | *Voltage enabled* | Bit that indicates if the HV supply voltage is applied or not on the drive power section |
| 5 | | *Quick stop* | Encoding the state of the CiA402 state machine (see Table 8.11). |
| 6 | | *Switch on disabled* | |
| 7 | | *Warning* | Bit which is set when a dynamic Warning is detected (WarnDynamic) |
| 8 | | Reserved | |
| 9 | | *Remote* | Bit which is set when the Controlword is processable from the CiA402 state machine. If you write the parameter Controlword when this bit is equal to 0, the operation does not have any effect. |
| 10 | 8,9,10 | Reserved | |
| | Others | *Target reached* | Bit which is set when the motor reaches the set-point (see Position reached target, Speed target reached or Torque target reached). In the homing mode it is set when the procedure ends. It's always reset on the homing operative mode entrance (writing 6 on the operative mode, when the ModesOfOperationDisplay has a different value) or when a new procedure is started. |
| 11 | All | *Internal limit active* | Bit that must be set when at least one position limit is reached, speed or torque (see Chapter 19, *Motion limits*). |
| 12 | 1 | *Set-point acknowledge* | Status of capture / processing of the position set-point (see Section 22.9, *Profile Position Mode*). |
| | 3, -113 and -111 | *Speed* | Bit which is set to Stopped motor. |
| | 6 | *Homing attained* | Bit which is set when the homing procedure is regularly completed (see Section 22.19, *Homing Mode*). It continues to indicate the last executed homing status, until a new procedure is started. For the drives with an absolute encoder installed, the homing status remains stored in the drive even if it is turned off and on again. |
| | 7 | *Ip mode active* | Status of the *Interpolated Position Mode* (see Section 22.10, *Interpolated Position Mode*). |
| | 8 | Target Position ignored | Bit which is set when the TargetPosition is used (see Section 22.11, *Cyclic Synchronous Position Mode*). |
| | 9 | Target Velocity ignored | Bit which is set when the TargetVelocity is used (see Section 22.12, *Cyclic Synchronous Velocity Mode*). |
| | 10 | Target Torque ignored | Bit which is set when the TargetTorque is used (see Section 22.13, *Cyclic Synchronous Torque Mode*). |

| Bit | Mode | Name | Description |
|---|---|---|---|
| | Oth-ers | Reserved | |
| 13 | 1, 8 | *Following error* | Presence or absence of the Error of position tracking. |
| | 6 | *Homing error* | Bit which is set when an error is detected during the homing procedure (see Section 22.19, *Homing Mode*). It continues to indicate the last executed homing status, until a new procedure is started. For the drives with an absolute encoder installed, the homing status remains stored in the drive even if it is turned off and on again. |
| | Oth-ers | Reserved | |
| 14 | All | Reserved | |
| 15 | All | Reserved | |

***Table 8.12. Meaning of the bits of the Statusword.***

To run some operations with the CiA402 state machine, it's necessary to write some commands in the Controlword parameter. The bits of the parameter Controlword are divided in the following way:

- *Bit 0 - 3 and 7* to command the Transition of the CiA402 state machine.
- *Bit 8* to manage the *Halt* command.
- *Bit 4 - 6* to ask for some specific commands that change depending on the value of ModesOfOperationDisplay.
- *Bit 9 - 15*Reserved

In order to change the state of the CiA402 state machine, write in the parameter Controlword the commands in the following chart. The bits shown with a 'x' are not important to determine the command and the symbol ↑ shows a transition from 0 to 1 of the related bit.

| Command | Controlword | Transitions | Related link |
|---|---|---|---|
| Shutdown | xxxx xxxx 0xxx x110 | 2, 6, 8 | Section 22.2, *Disabling by using the master* |
| Switch On | xxxx xxxx 0xxx 0111 | configurable[a] | Section 22.1, *Enabling by using the master* |
| Switch On + Enable Operation | xxxx xxxx 0xxx 1111 | 3 + 4 | |
| Disable Voltage | xxxx xxxx 0xxx xx0x | 7, 9, 10, 12 | Section 22.2, *Disabling by using the master* |
| Disable Operation | xxxx xxxx 0xxx 0111 | 5 | |
| Enable Operation | xxxx xxxx 0xxx 1111 | 4, 16 | Section 22.1, *Enabling by using the master* |
| Quick Stop | xxxx xxxx 0xxx x01x | 7, 10 | Section 22.2, *Disabling by using the master* |
| | | 11 | Section 22.3, *Carrying out a stop by using the master* |
| Fault Reset | xxxx xxxx ↑xxx xxxx | 15 | Section 24.5, *Resetting the errors* |

[a]See SwitchedOnOptionCode.

***Table 8.13. Commands for the state transitions of the CiA402 state machine.***

NOTE

**In the command Switch On + Enable Operation, the transition 4 is automatically run after the running of the transition 3.**

In the following chart you can find the description of the single bits of the parameter Controlword; some of them have a different meaning depending on the value of ModesOfOperationDisplay: the column Mode shows the value that the parameter ModesOfOperationDisplay must have so that the bit shown has the specified meaning.

| Bit | Mode | Name | Description |
|---|---|---|---|
| 0 | All | *Switch on* | Bit used to encode the commands of the state transitions of the CiA402 state machine (see Table 8.13). |
| 1 | | *Enable voltage* | |
| 2 | | *Quick stop* | |
| 3 | | *Enable operation* | |
| 4 | 1 | *New set-point* | A rising edge of this bit enables the trajectory generator that controls the profile parameters, processes them and runs the positioning (see Section 22.9, *Profile Position Mode*). |
| | 6 | *Homing operation start* | Bit enabling the start/stop of the homing procedure (see Section 22.19, *Homing Mode*). |
| | 7 | *Enable ip mode* | Bit used for the enabling/disabling of the *Interpolated Position Mode* (see Section 22.10, *Interpolated Position Mode*). |
| | Others | Reserved | |
| 5 | 1 | *Change set immediately* | Selector of the positioning mode between Single set-point and Set of set-point, to be set with the transition of the bit *New set-point* (see Section 22.9, *Profile Position Mode*). |
| | Others | Reserved | |
| 6 | 1 | *Absolute / Relative* | Selector of the mode used to interpret the position target, to be set with the transition of the bit *New set-point* (see Section 22.9, *Profile Position Mode*). |
| | Others | Reserved | |
| 7 | All | *Fault reset* | Bit used to encode the commands of the state transitions of the CiA402 state machine (see Table 8.13). |
| 8 | | *Halt* | Bit used to run a stop of the motor (*Halt*, see Section 22.3, *Carrying out a stop by using the master*). |
| 9 - 15 | | Reserved | |

***Table 8.14. Meaning of the bits of the Controlword*.**

Please remember that a single writing of the Controlword cannot run either a transition or the start of a motion at the same time. In particular if the bits causing changes of the state (bit 0 - 3 and 7) are different from those written with the precedent access to the Controlword, the other bits (bit 4 - 6, 8 - 15) are not taken into consideration. Vice versa, if the bits causing changes of the state (bit 0 - 3 and 7) do not change, other bits are also taken into consideration, but only if the drive is in the Operation enable state.

IMPORTANT

**During the access in writing to the Controlword no bits changes are accepted during a state transition of the CiA402 state machine. This condition is reported by the Remote bit of the Statusword (see Table 8.12.**

## 8.6. System manager

To run some operations or commands different from those offered by the CiA-402 it is necessary to use the System manager. To run a command you must respect the following rules:

1. write the code of the desired command of the parameter SysMngCommand;
   - if during the writing of the parameter SysMngCommand you get the error code Attempt to write a read only object, it means that the command cannot be run since you are already running another command
   - if during the writing of the parameter SysMngCommand you get the error code Generic error, it means that the command cannot be run; the reason of this is specified by the parameter SysMngError
   - if during the writing of the parameter SysMngCommand you get the error code No error, the command is accepted and immediately run
2. wait for the end of the command, that is when the parameter SysMngCommand is equal to 0
3. check if the command was correctly run by reading the possible cause of the error in the parameter SysMngError
4. when a command is running (parameter SysMngCommand different from 0), the drive cannot be taken to the Operation enable state
5. when an axis motion command is active it's not possible to write in the ModesOf-Operation parameter, and the ModesOfOperationDisplay parameter assumes the value -127 (*Tuning Mode*).

**System manager safety conditions**

The following safety conditions are needed to run some commands:

1. disabled drive
2. setting the functionality Generic Output (I/O X - Out X) for the digital outputs and for the digital I/O (bidirectional peripheral) programmed as outputs
3. digital outputs and digital I/O (bidirectional peripheral) programmed as outputs, switched off
4. capture unit in stop

**System manager command forcing**

To ask the drive to go automatically in the System manager safety conditions, write the value 1 in the parameter SysMngEnForcing before writing the command. The safety conditions are forced *solo* only for those commands of the System manager requesting it.

⚠ WARNING

**You may force the commands only after having seen the System manager safety conditions.**

| SysMngCommand | Description |
|---|---|
| 2200 | Permanent memory: restore to default of all parameters (permanent) |
| 2201 | Reset to default of all parameters (temporary) |
| 2250 | Permanent memory: delete motor and sensor data |
| 2300 | Permanent memory: reload value of all parameters |
| 5000 | Hard firmware reset |
| 5001 | Soft firmware reset |
| 5100 | Request download firmware |
| 6000 | Downloading parameters file |

| SysMngCommand | Description |
|---|---|
| 8000 | Request download PLC program |
| 8100 | Request erase PLC program |

***Table 8.15. Commands of the System manager requesting the System manager safety conditions.***

NOTE

**The safety conditions can be set manually. In these cases it is not necessary to force the commands.**

### Reset of the Watchdog of the System manager

Some System manager commands need a cyclic writing in the ResetWatchdogTimeout parameter to inform the drive that the connection with the interlocutor that has been required by the command is still active and it's working. If the time between two writing operations is longer than 2 seconds, the current command is terminated and SysMngError assumes the value 1001. The commands which need the writing of ResetWatchdogTimeout are listed in the Table 8.16. In the ResetWatchdogTimeout parameter it has to be written the value of SysMngCommand to reset the timeout.

| SysMngCommand | Description |
|---|---|
| 1001 | Tuning: extended inertia estimator |
| 1002 | Tuning: inertia estimator |
| 1003 | Tuning: RL estimator |
| 1010 | Function Generator current D |
| 1015 | Function Generator current Q |
| 1020 | Function Generator velocity |
| 1030 | Function Generator position |
| 1040 | Function Generator profile velocity |
| 1050 | Function Generator profile position |

***Table 8.16. System manager commands that require the watchdog reset.***

# Chapter 9 Communication port CANopen

NOTE

**About the communication settings with a CANopen Master, please refer to Section 8.1, *Communicate with master CANopen*.**

NOTE

**All the information in this chapter are valid only for the drives of version CAN.**

The communication port CANopen is the interface of the main bus for the drives, version CAN. The main bus must be connected to the connectors CN2 e CN3. The implemented CANopen protocol meets the specifications of the CAN in Automation organization (CiA-301 rev. 4.2.0 and CiA-402 edit. 3.0).

**NMT state machine**

To check the messages flow of the communication port, the nodes CANopen are equipped with a state machine (*NMT state machine*).

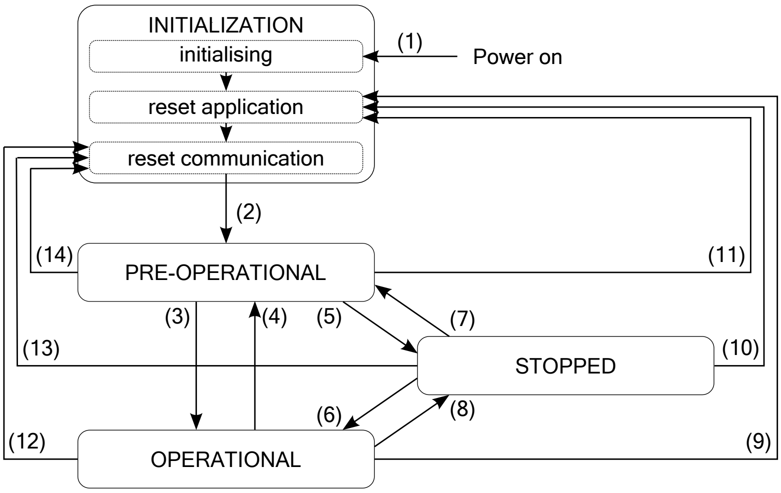



***Figure 9.1. State diagram of the NMT state machine.***

The states of the NMT state machine have the following meaning:

- INITIALIZATION: drive initialization; at the end of these stage, the drive sends the message *Boot-up* to show its presence in the network;
- PRE-OPERATIONAL: drive and PDOs configuration; all the communication services are enabled, except from the PDOs;
- OPERATIONAL: all the services CANopen are enabled;
- STOPPED: the drive communication is effectively stopped; only the NMT services are working.

After the Power on, the drive runs the operations that are scheduled in the INITIALIZATION state and later it goes to the PRE-OPERATIONAL state, ready to carry out the master's requests. The led LB shows the state of the NMT state machine, according to what is reported in ID_TAB_CANopenLedRun.

In the drives of the series *i*BMD, the communication port CANopen manages the following objects:

- Network management objects (NMT)
- Emergency object (EMCY)
- Synchronization object (SYNC)
- Service data object (SDO)
- Process data object (PDO).

| State | Value | Available services | | | | |
|---|---|---|---|---|---|---|
| | | NMT | SYNC | EMCY | SDO | PDO |
| INITIALIZATION | - | - | - | - | - | - |
| PRE-OPERATIONAL | 127 | *YES* | *YES* | *YES* | *YES* | - |
| OPERATIONAL | 5 | *YES* | *YES* | *YES* | *YES* | *YES* |
| STOPPED | 4 | *YES* | - | - | - | - |

***Table 9.1. Available services in the states of the NMT state machine.***

In order to use the different services of the port CANopen it is necessary to command and check the NMT state machine state. To edit or check the state of the NMT state machine, use the NMT objects.

NOTE

**The following paragraphs describe how the CiA-301 services are implemented in the drives *i*BMD.**

# 9.1. Network management objects (NMT)

**Node control services**
These services allow the state change of NMT state machine. The message sent out by the master contains the node number which the message is addressed to and the command that it must run (command specifier). In the following table you can find the commands and transitions allowed by the state machine (please refer to Figure 9.1):

| Command | Command specifier | Description | Transitions |
|---|---|---|---|
| Start Remote Node | 1 | Take the drive to the OPERATIONAL state | (3), (6) |
| Stop Remote Node | 2 | Take the drive to the STOPPED state | (5), (8) |
| Enter Pre-Operational | 128 | Take the drive to the PRE-OPERATIONAL state | (4), (7) |
| Reset Node | 129 | Take the drive to the *Reset Application* substate in the INITIALIZATION state | (9), (10), (11) |
| Reset Communication | 130 | Take the drive to the *Reset Communication* substate in the INITIALIZATION state | (12), (13), (14) |

***Table 9.2. Node control services commands***

**Error control services**
These services control the right working of the network and of the present nodes. The implemented protocol has two services:

1. Service node guarding event: the master sends a reading message to the drive in the NMT state machine state. This request is sent at regular intervals, with a GuardTime period. If the master does not receive any answer or the state does not coincide with the expected one, this means that there have been some problems in the drive or in the network. In this case the master is facing the error condition *Node guarding event*.
2. Service life guarding event: if this service is enabled, the drive must wait for the periodic reading of the NMT state machine state carried out by the master. The drive answers with the numerical code linked to its current state (see Table 9.1). If the drive does not receive any reading request for a longer time than the product between the GuardTime parameter and the LifeTimeFactor parameter, it enters in the condition *Life guard error* and it reports the CAN communication error error.

To enable these services, please run the following procedure:

- write the value of the period of the message sending in GuardTime;
- write the LifeTimeFactor parameter with the tolerance factor (number of periods that the drive must wait for before reporting the error);
- send periodically the request of reading the state of the NMT state machine.

## 9.2. Emergency object (EMCY)

When some errors are found in the drive, an emergency message is immediately sent. The message contains all the useful information to identify the error type and it is made up by 8 bytes divided into four fields: *Emergency Error Code* (EEC, byte 0-1), ErrorRegister (byte 2), *Manufacturer Specific Error Code* (MSEC, byte 3) and Reserved (byte 4-7, not used). In the following table you can find the values of the EEC and MESC fields of the emergencies according to the error found:

| EEC | MSEC | Description |
|---|---|---|
| 0x0000 | 0x00 | Reset error or no error. |
| 0x2250 | 0x50 | Power or motor short circuit. |
| 0x2310 | 0x51 | Power or motor over current. |
| 0x2350 | 0x52 | I2T limit reached. |
| 0x3210 | 0x42 | DC bus over voltage. |
| 0x3220 | 0x43 | DC bus under voltage. |
| 0x4210 | 0x60 | See (bit 1) of Thermal management. |
| | 0x62 | See (bit 2) of Thermal management. |
| 0x4310 | 0x61 | See (bit 0) of Thermal management. |
| 0x5113 | 0x48 | See (bit 1) of Logic voltage error. |
| 0x5114 | 0x49 | See (bit 0) of Logic voltage error |
| 0x5115 | 0x4A | See (bit 2) of Logic voltage error. |
| 0x6320 | 0x91 | At least one of the bits of Parameters serious error is active. |
| | 0x92 | At least one of the bits of Digital IO configuration error is active. |
| | 0x93 | See (bit 10) of Thermal management. |
| 0x7200 | 0xA7 | See (bit 6) of Thermal management. |
| | 0xA8 | See (bit 7) of Thermal management. |
| | 0xA9 | See (bit 8) of Thermal management. |
| 0x8110 | 0x01 | Overwritten message, please check the related overwriting notifications in the parameters CANopenEmcyService (Table 24.7) and CANopenEmcyProcess (Table 24.8). |
| 0x8120 | 0x02 | CAN in error passive state |
| 0x8130 | 0x03 | Life guard error |
| 0x8140 | 0x04 | Recovered from bus-off |
| 0x8210 | 0x10 | PDO RX 1 too short |
| | 0x11 | PDO RX 2 too short |
| | 0x12 | PDO RX 3 too short |
| | 0x13 | PDO RX 4 too short |
| 0x8220 | 0x20 | PDO RX 1 too long |
| | 0x21 | PDO RX 2 too long |
| | 0x22 | PDO RX 3 too long |
| | 0x23 | PDO RX 4 too long |
| 0x8611 | 0x70 | Position following error. |
| 0x8700 | 0x05 | Sync controller error. |
| 0xFF00 | 0xA0 | Real time mode error. |
| 0xFF01 | 0xA6 | User Fault. |
| 0xFF04 | 0xAA | Vedere bit 0 di /STO Management Error. |
| 0xFF05 | 0xAB | Last command requested failed. |

| EEC | MSEC | Description |
|---|---|---|
| 0xFF06 | 0xB0 | See (bit 4) of Internal Error. |

***Table 9.3. Codes for the field Emergency Error Code (EEC) and Manufacturer Specific Error Code (MSEC).***

It is possible to enable or disable this service by operating on the EMCY_CobID parameter.

# 9.3. Synchronization object (SYNC)

SYNC is a periodic broadcast-type message that does not imply any answer. SYNC synchronizes all drives in the CANopen network by using the incoming PDOs and sending the outgoing PDOs, that are synchronized with the SYNC itself.

The SYNC message is sent to the node *producer* and it is addressed to all the other nodes, called *consumer*. Any network node can be service producer but this role is normally carried out by the master. The drives of the series *i*BMD can only be consumer nodes.

The service must be configured in the PRE-OPERATIONAL state before being used. In order to do this it is necessary to write the two parameters SYNC_CobID and CommunicCyclePeriod.

Once the master has enabled the SYNC service, if the time interval between two consecutive SYNC messages is different from the time set by the CommunicCyclePeriod (±50%) parameter, the drive notifies CAN communication error.

**PDO transmission/sending/analysis sequence**
The order with which the messages are transmitted/sent/analized is the following:

- the master sends the PDO(s) RX;
- the synchronism signal is activated. This signal is the SYNC message;
- the drive composes and sends the PDO TX.
- the drive analyzes and executes the operations that are required by the PDO RX.

NOTE

**Example with CANopen drive:**
**if a command is sent to Controlword with the PDO RX, the drive analizes it on the following SYNC and indicates its effect through the Statusword on the next SYNC only.**

# 9.4. Service data object (SDO)

The SDOs are objects whose aim is exchanging data with confirmation and are used to access all the parameters of the vocabulary (Chapter 27, *Parameters vocabulary*). The size of their messages is set at 8 bytes: some are used as control bytes and others for sending data.

The drives of the series *i*BMD support two types of data transfer with this service:

- mode *expedited*: SDO is made up by a single request message and a single answer message, in which 4 bytes are used for the control (type of operation to be run, indexes and subindex). You can transfer up to 4 bytes of effective data;

- mode *normal*: the transfer is an initial negotiation between master and slave where you can find the size of the data to be transferred; the data are later sent through the transmission of 8-byte-messages containing one single control byte and 7 data bytes.

The SDO are appropriate to configure the drive and the PDOs (see Section 9.5, *Process data object (PDO)*), and in general for the low priority communication between the drive and the master.

## 9.5. Process data object (PDO)

The PDOs are used for the exchange of data in real time without any confirmation by the one receiving them; in this way the network is less overloaded.

The PDOs, like the SYNC, are based on the relation *producer - consumer*, in which the producer sends the PDO message and the consumer receives it. The outgoing PDO messages, sent by the producer node, are called PDO TX, while the incoming messages in the consumer node are called PDO RX. The drives of the series *i*BMD imply the possibility of managing up to 4 incoming PDOs (PDO RX) and 4 outgoing PDOs (PDO TX).

Any network node can send a PDO message linked to an ID (COB-ID). Any other network node, that finds a correspondence between the COB-ID of the PDO in the network and one of its PDOs RX, accepts the message and interprets it.

The PDOs must be configured and enabled in the PRE-OPERATIONAL state before being used. Their configuration implies the writing of two parameter groups:

- *Communication parameters*: parameters for the management of the transmission and the receiving of the PDOs (addresses CANopen from 0x1400 to 0x1403 for PDO RX and from 0x1600 to 0x1603 for PDO TX);
- *Mapping parameters*: parameters for the management of the mapping in the PDOs of the mappable parameters (addresses CANopen from 0x1800 to 0x1803 for PDO RX and from 0x1A00 to 0x1A03 for PDO TX).

NOTE

**PDOs are enabled only in the OPERATIONAL state.**

IMPORTANT

**the inhibit time parameter (for example PdoTx1_InhibitTime) is used to inhibit the sending of the related TPDO for a time period equal to inhibit value * 100us. The period starts since when the last related TPDO has been sent, unless there is already one waiting to be transmitted. In this case the period has yet to begin to elapse. The sending inhibition even causes the event loss.**

### 9.5.1. PDO Mapping

Every PDO can contain up to 8 bytes of informations. For that reason a single PDO can be mapped with up to 8 parameters of 1 byte each one, or with a number of parameters that have an overall dimension of at maximum 8 bytes.

The parameters that can be mapped are identified by the written *"YES"* in the *"PDO"* field of the table that describe them (see Section 27.1, *Agreements on the parameters description*)

Both the incoming PDO RX interpretation and the outgoing PDO TX construction have to respect the order in which the parameters are mapped in the PDO, starting from the 1° till, at maximum, the 8°. So it's important to pay attention on the parameters insertion order during the PDO mapping operation.

In particular, to use the PDO RX to execute an axis movement, it's necessary to insert the moving parameters first (e.g. Velocity, target Position, ...) and at last, as last parameter mapped on PDO, the ControlWord to command the movement. (please refer to the PdoRx3_MappingParameters and PdoRx4_MappingParameters default PDO RX).

The full list of all the parameters for the PDOs configuration can be found in Section 27.23, *PDO managed by the port CANopen (10000-11999)*.

NOTE

**The procedures for the PDOs management are in compliant with the CiA-301 specific.**

Chapter 10

# Communication port EtherCAT

NOTE

**About the communication settings with a EtherCAT Master,**
**please refer to Section 8.2, *Communicate with master EtherCAT*.**

NOTE

**All information in this chapter are valid for the drives of version ETC.**

The communication port EtherCAT is the interface of the main bus for the drives of version ETC. The main bus must be connected to the connectors CN2 e CN3. The implemented protocol EtherCAT respects the specifications of the organization EtherCAT Technology Group (ETG). In the drives of the series *i*BMD the interface towards the network EtherCAT is constituted by the ASIC ET1100. To communicate with the drive you can refer to the data sheet of the ET1100.

**EtherCAT state machine**

To check the flow of the messages of the communication port, the nodes EtherCAT are equipped with a state machine.

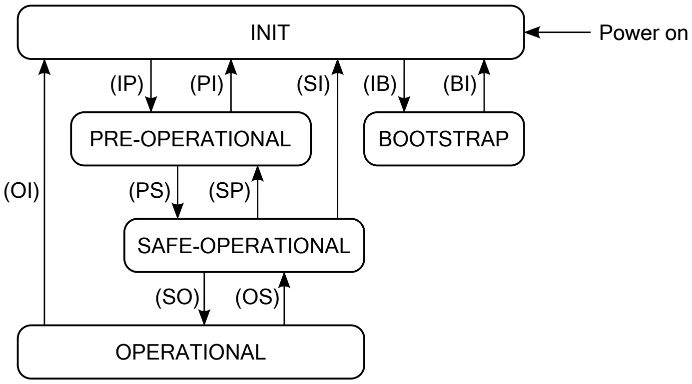



*Figure 10.1. Status diagram of the EtherCAT state machine.*

The states of the EtherCAT state machine have the following meaning:

- INIT: initialization of the drive; no protocol and no service are enabled; to recognize and set the drive the master can have access only to the registers of the ET1100;
- PRE-OPERATIONAL: configuration of the drive and of the PDOs; all communication protocols are enabled but the PDO service is disabled;
- SAFE-OPERATIONAL: all communication protocols are enabled and the PDO service is enabled only during transmission (PDO TX);
- OPERATIONAL: all communication protocols are enabled and the PDO service is completely enabled;
- BOOTSTRAP: only the update of the drive firmware with the protocol File access over EtherCAT is enabled.

After the Power on the drive runs the operations scheduled in the INIT state and remains in such state waiting for the commands coming from the master. The led RUN shows the state of the EtherCAT state machine, according to what is reported in Table 8.8.

| State | Available services | | | |
|---|---|---|---|---|
| | **CoE** | **FoE** | **PDO TX** | **PDO RX** |
| INITIALIZATION | - | - | - | - |
| PRE-OPERATIONAL | *YES* | *YES* | - | - |
| SAFE-OPERATIONAL | *YES* | *YES* | *YES* | - |
| OPERATIONAL | *YES* | *YES* | *YES* | *YES* |
| BOOTSTRAP | - | *YES* | - | - |

*Table 10.1. Available services in the states of the EtherCAT state machine.*

NOTE

**In the data sheet of the ET1100 you can find the available registers used to monitor the status of the communication in all its aspects.**

## Sync manager (SM)

The management of the messages of the communication port EtherCAT is carried out through the Sync manager (SM). In the following table you can find the features of the Sync managers that can be used in the drives of the series *i*BMD.

| SM | Communication mode | Starting address | Dimension (byte) | Available services |
|---|---|---|---|---|
| 0 | Mailbox RX | 0x1000 | 128 | CoE, FoE |
| 1 | Mailbox TX | 0x1080 | 128 | |
| 2 | Buffered RX | 0x1100 | 64 | PDO RX |
| 3 | Buffered TX | 0x1180 | 64 | PDO TX |

***Table 10.2. Features of the managed Sync manager (SM).***

The communication modes of the Sync managers show how the data are exchanged between the master and the drives:

- *Mailbox mode*: mechanism of handshake guaranteeing the complete reading of the message before sending next message; it is used for the communication protocols;
- *Buffered mode*: access to the buffers of the data in a substantial way in any moment; it is used for the PDOs.

The parameters of the Sync managers are described in Section 27.24, *Sync manager and PDOs managed by the port EtherCAT*.

NOTE

**The following paragraphs describe how the functionalities for the communication port EtherCAT have been implemented in the drives *i*BMD.**

## 10.1. Protocol CANopen over EtherCAT (CoE)

The CoE implements in the drives EtherCAT the application layer of the protocol CANopen (see specifications of CiA-301).

The CoE provides the Service data object (SDO) to exchange data with confirmation. The SDOs are used to access all parameters of the vocabulary (Chapter 27, *Parameters vocabulary*). Their messages have the same dimension as the whole mailbox of the protocol CoE (see Table 10.2). The drives of the series *i*BMD support two types of data transfer with the SDOs:

- mode *expedited*: SDO is composed by one message of request and one message of answer; it is possible to transfer up to four bytes of data through this mode.
- mode *normal*: it is used for the transfer of data with a dimension bigger than four bytes.

The SDO are appropriate to configure the drive and the PDOs (see Section 10.3, *Process data object (PDO)*), and in general for the low priority communication between the drive and the master.

The CoE also provides the service *SDO information* to read the information on the parameters of the vocabulary: the whole list of all parameters, the list of the parameters mappable on PDO, information on the single parameters, etc.

## 10.2. Emergency Error Code

In the drive ETC the emergency management is not implemented. Through the ErrorCode parameter the code of the last error is reported. The code contains all the informations that are useful to indentify the error type, and is composed by 8 bytes that are divided in three parts: *Emergency Error Code* (EEC, byte 0-1), ErrorRegister (byte 2) and Reserved (byte 3-7, not used). In the following table the values of the EEC part, according to the detected error, are reported:

| EEC | Description |
|---|---|
| 0x0000 | Reset error or no error. |
| 0x2250 | Power or motor short circuit. |
| 0x2310 | Power or motor over current. |
| 0x2350 | I2T limit reached. |
| 0x3210 | DC bus over voltage. |
| 0x3220 | DC bus under voltage. |
| 0x4210 | Thermal management (one of the following):<br>- Over temperature of logic section (bit 1)<br>- Over temperature of motor (bit 2) |
| 0x4310 | See (bit 0) of Thermal management. |
| 0x5114 | See (bit 0) of Logic voltage error. |
| 0x6320 | At least one of the following situations has occurred:<br>- At least one of the bits of Parameters serious error is active;<br>- At least one of the bits of Digital IO configuration error is active. |
| 0x7200 | Thermal management (one of the following):<br>- Power Temp Sensor hardware failure (bit 6);<br>- Logic Temp Sensor hardware failure (bit 7);<br>- Motor Temp Sensor hardware failure (bit 8). |
| 0x8611 | Position following error. |
| 0x8700 | Sync controller error |
| 0xFF00 | Real time mode error. |
| 0xFF01 | User Fault. |
| 0xFF04 | See (bit 0) of /STO Management Error. |
| 0xFF05 | Last command requested failed. |

***Table 10.3. Codes for the Emergency Error Code (EEC) part.***

## 10.3. Process data object (PDO)

The PDOs are used for the exchange of data in real time without any confirmation by the one receiving them; in this way the network is less overloaded.

The PDOs are based on the relation *producer* - *consumer*, in which the producer sends the PDO message and the consumer receives it. In the network EtherCAT it is always the master who starts the communication and sends the PDOs; depending on the type of PDOs, the drives in the network can be producer and complete the outgoing PDOs, or consumer with the incoming PDOs. The *i*BMD drives offer the possibility to manage up to 4 outgoing PDOs (PDO TX) and 4 incoming PDOs (PDO RX). Every PDO must be assigned to a Sync manager (SM). The association of type of PDO and number of Sync manager (SM) is reported in Table 10.2.

The PDOs must be configured and enabled in the PRE-OPERATIONAL state before being used. Their configuration implies the writing of two parameter groups:

- *Mapping parameters*: parameters used to manage the mapping in the PDOs of the mappable parameters (addresses CANopen from 0x1800 tp 0x1803 for the PDO RX and from 0x1A00 to 0x1A03 for the PDO TX);
- *Sync manager PDO assignment parameters*: parameters to assign the PDOs to the Sync manager (SM) (addresses CANopen from 0x1C10 to 0x1C13).

NOTE

**The PDOs TX are enabled in the SAFE-OPERATIONAL and OPERATIONAL states; the PDOs RX are enabled only in the OPERATIONAL state.**

## 10.3.1. PDO Mapping

The PDOs allows the overall exchange of 64 bytes in reception (for the 4 PDO RX) and others 64 byte in transmission (for the 4 PDO TX). Each PDO can contain up to 8 parameters independently by their dimension. If, for example, 2 PDO TX are mapped with 8 parameter of 4 byte each, will be used all the 64 bytes that are available in the PDOTX reserved exchange area and, therefore, it's not possible to map other PDOs (naturally the same applies for PDO RX).

The parameters that can be mapped are identified by the written *"YES"* in the *"PDO"* field of the table that describe them (see Section 27.1, *Agreements on the parameters description*)

Both the incoming PDO RX interpretation and the outgoing PDO TX construction have to respect the order in which the parameters are mapped in the PDO, starting from the 1° till the last one. So it's important to pay attention on the parameters insertion order during the PDO mapping operation.

In particular, to use the PDO RX to execute an axis movement, it's necessary to insert the moving parameters first (e.g. Velocity, target Position, ...) and at last, as last parameter mapped on PDO, the ControlWord to command the movement. (please refer to the PdoRx3_MappingParameters and PdoRx4_MappingParameters default PDO RX).

The whole list of all useful parameters to manage the PDOs is reported in Section 27.24, *Sync manager and PDOs managed by the port EtherCAT*.

NOTE

**The procedures to manage the PDOs are in compliance with the specifications shown by EtherCAT Technology Group (ETG).**

## 10.3.2. Missing or corrupted PDO RX management

The EtherCAT field bus is not tolerant about the messages that are lost in the network and, on consequence, doesn't manage their automatic re-transmission (as instead happens in the CANopen field bus). That implies that if a PDO RX is corrupted or doesn't arrive in correspondence of the synchronization reference (see Section 10.4, *Synchronization*), the drive immediately generates an alarm and disables the motor. To avoid this inconvenience Bonfiglioli has implemented in the *i*BMD drive series a PDO RX monitoring and management system.

This system has been introduced to avoid that the drive goes in alarm state if the consecutive missing of a certain number of PDO RX is not considered serious (see EtcPdoRxMissingTolerance). Until the loss is lower or equal to this value the movement will proceeds with the last valid received data. In the particular case of *Interpolated Position Mode* the drive cannot command to continue the motion because it needs to receive regularly the velocity and position targets (see Section 22.10, *Interpolated Position Mode*), then it will move the motor by reconstructing the profile coherently with the last received valid data (then referring to the last valid PDO RX), and so continuing the movement that it was making before the PDO RX loss.

This means that the more is high the number of tolerated consecutive and not valid PDO RX, the more long may be the movement that's defined by the previous parameters and not controlled by the master.

NOTE

**The corrupted or missing PDO RX management is active only when the drive is in OPERATIONAL state (see EtherCAT state machine).**

**Exceeded the corrupted or missing PDO tolerance (see EtcPdoRxMissingTolerance) the drive goes in synchronization error (see bit 3 of Table 24.9).**

**The occurrence of this alarm condition implies the transition from the OPERATIONAL to the SAFE-OPERATIONAL state.**

NOTE

**If the PDOs RX arrive too close each other, the alarm is immediately generated independently of the set tolerance (see bit 3 of Table 24.9).**

**For a complete diagnostic see the EtherCAT_Diagnostics parameter group.**

## 10.4. Synchronization

In the drives ETC the synchronization of the PDOs is managed through the Sync manager (SM) by setting the related registers of the ET1100. The related settings can be read in the parameters *Sync manager synchronization* (see Section 27.24, *Sync manager and PDOs managed by the port EtherCAT*).

In the drive of the *i*BMD series have been implemented three synchronization modes:

- Free run;
- Soft sync;
- Hard sync.

**Free run**

The Free run mode does not have any mechanism of synchronization of the PDOs, they are managed at low priority.

**Soft sync**

The Soft sync mode synchronizes the outgoing PDOs TX with the incoming PDOs RX. This synchronization way is useful when the master does not support the synchronization of Hard sync and/or when there is no need for a correction because of the delays of the network EtherCAT (for example on networks of small dimension). To use this mode it is necessary to set the Sync managers of the PDOs in order to get in the parameters SM2_SynchronizationType and SM3_SynchronizationType the values respectively 1 and 34 and it is necessary to set the $T_{SYNC}$ through the parameter CommunicCyclePeriod.

**Hard sync**

The Hard sync mode can be used only with the masters that manage the functionality Distributed clocks. The distributed clocks is used to synchronize the drive more precisely by cancelling any errors generated by propagation times, offset and derive. With the synchronization way Hard sync it is possible to synchronize up to 65535 drives (highest limit allowed by a network EtherCAT). To use this mode it is necessary (for every drive):

- to run with the master the sequence of operations to calculate the corrections of the times for the distributed clocks and to apply them;
- to write the cycle time $T_{SYNC}$ in the registers of the ET1100 for the cyclic generation of the Sync Signal signal;
- to set the registers of the Sync managers of the PDOs in order to get the value 2 in the parameters SM2_SynchronizationType and SM3_SynchronizationType.

NOTE

**For a correct setting of the synchronization it is advisable to use the settings as described in the file EtherCAT Slave Information (ESI) that can be downloaded in the DOWNLOAD AREA of the website http://www.bonfiglioli.com**

**PDO transmission/sending/analysis sequence**

The order with which the messages are transmitted/sent/analized is the following:

- the master sends the PDO(s) RX;
- the synchronism signal is activated. This signal is the SyncSignal if the used synchronism is the Hard sync or the same PDO RX message if the used synchronism is the Soft sync;
- the drive composes and sends the PDO TX.
- the drive analyzes and executes the operations that are required by the PDO RX.

Chapter 11

# Auxiliary communication port Modbus

NOTE

**For the communication settings with a Modbus Master, please refer to Section 8.3, *Communicate with master Modbus RS232 (auxiliary communication port)*.**

The drives of the *i*BMD series provide an auxiliary communication port on which the Modbus protocol is implemented. Through the CN1 connector it's possible to connect to the port, which purpose is the configuration of the drive through MotionDrive. For further details please see Section 8.3, *Communicate with master Modbus RS232 (auxiliary communication port)*.

NOTE

**The Modbus protocol that's implemented in the drives respects the regulations of the Modbus organization: in this section are only reported the implemented functionalities indications.**

**In the protocol only the transmission mode of RTU type has been implemented.**

**Modbus frame**

The protocol Modbus uses a frame composed by many fields, in Figure 11.1 you can find the scheme.

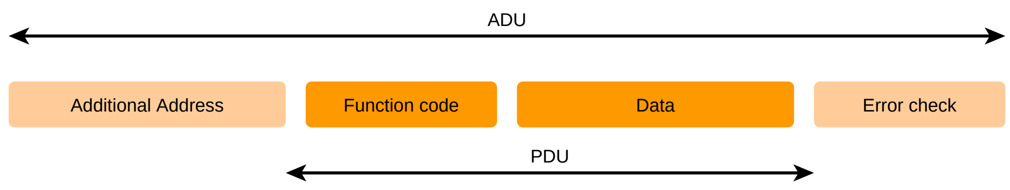



*Figure 11.1. Modbus frame*

The field *Function code* shows which operation the drive must run, once received and checked the whole frame, it checks it is not damaged. This information occupies 1 byte and has a range of valid values from 1 to 127; the codes between 128 and 255 are used for the Exceptions but the value 0 is not accepted. In Table 11.1 you can find all the accepted codes.

| Funct. Code | Name | Description |
|---|---|---|
| 3 | Read Holding Register | Reading one or more parameters (at 16/32 bits) starting from the Modbus address shown in the frame (such as *Read Input Register*). |
| 4 | Read Input Register | Reading one or more parameters (at 16/32 bits) starting from the Modbus address shown in the frame (such as Read Holding Register). |
| 6 | Write Single Register | Writing a parameter at 16 bits near the Modbus address shown in the frame. If the Modbus address refers to a parameter higher than 16 bits the operation is not run and the drive finds an exception. |
| 7 | Diagnostics | The diagnostics is only simulated and it has been implemented only to be compatible with the terminals requesting it. |
| 16 | Write Multiple Register | Writing one or more parameters (at 16/32 bits) starting from the Modbus address shown in the frame. |

***Table 11.1. Function Codes supported by the drives.***

The function codes (3, 4, 6 and 16), described in the previous chart, give full access to all drive parameters through the vocabulary in Chapter 27, *Parameters vocabulary*.

### Exceptions

If the drive receives a message without communication errors, but it cannot run the requested operation or there is an error in the protocol, the drive answers to the request with an exception frame. In Table 11.2 you can find the implemented exception codes.

| Funct. Code | Name | Description |
|---|---|---|
| 1 | Illegal function | Function code not supported. |
| 2 | Illegal data address | Modbus address not accepted. More precisely, the combination of the Modbus address and of the number of data to write / read is not valid (all addresses included in the requested range must be in the vocabulary). |
| 3 | Illegal data value | Data quantity not accepted (too high or equal to 0). |
| 4 | Slave device failure | Error in the running of the requested action. |

***Table 11.2. Exception codes implemented in the drive.***

NOTE

**The details on the error condition can be found in the group of parameters AuxiliaryPortError.**

# Chapter 12 Communicating with MotionDrive

## 12.1. MotionDrive via RS232 (auxiliary communication port)

Connect the PC serial port to CN1. For further information see Section 7.2, *Electrical connections*.

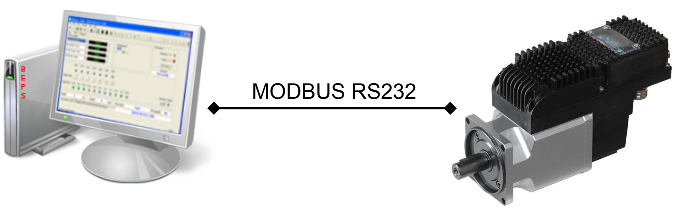
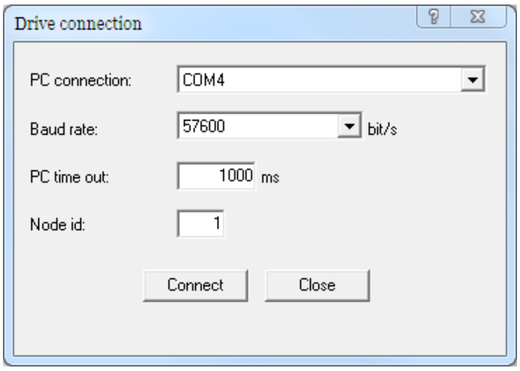



***Figure 12.1. Point-point communication scheme with MotionDrive.***

Start MotionDrive from: **Start menu** > **Programs** > **Bonfiglioli** > **MotionDrive.**

Set the connection parameters in the window *Drive connection.*




If the programme has already been started, run a new connection. Access:

**Main menu** > **File** > **New connection ...**

**Toolbar** >

**Connection parameters**

1. *PC connection*: choosing the connection physical port (COM1, COM2...)
2. *Baud rate*: choosing the communication speed (the drive default value is 57600bit/s).
3. *PC time out*: if the drive does not answer during a longer time period than this value, the communication is interrupted and it is necessary to reconnect (the default value is 500 ms).
4. *Node id*: set the value 1.

ADVICE

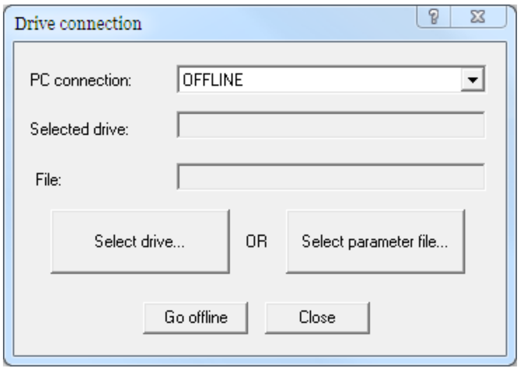
**In case of more connection interruptions it may be necessary to increase the PC time out. It is advisable not to increase the timeout more than 5 seconds.**

# 12.2. Offline mode

Through the Offline mode it is possible to connect to a virtual drive through MotionDrive. To enable this mode, start MotionDrive or request a new connection by pressing .

In the window *Drive connection:*

1. Choose *OFFLINE* in the pull-down menu *PC connection*
2. Select the drive type by pressing [Select drive...] or a parameters file previously saved by pressing [Select parameter file...].
3. Start the Offline mode by pressing [Go offline].




NOTE

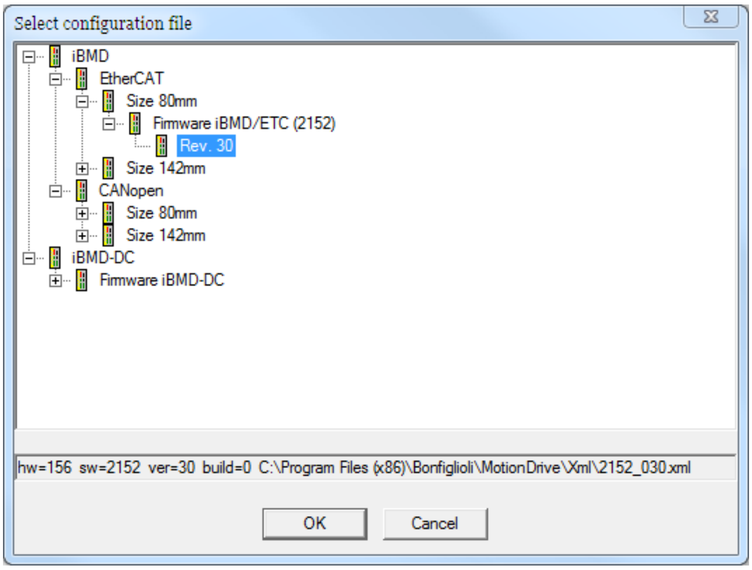
**The Offline mode is the best way to debug the system remotely by analysing the parameters file containing the problem.**

### How to choose the drive

Choose in the window *Select configuration file* the firmware Configuration file and the related version you are going to work with by exploring the stem-and-leaf diagram.

ADVICE

**Always choose the latest available firmware version. After having selected the file, check that the drive data shown in the field below are the wanted ones.**




### What you cannot do in the Offline mode

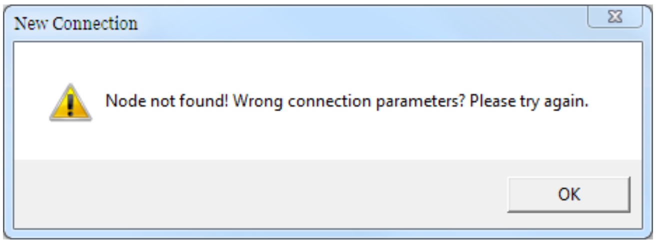
In the Offline mode you can run all the operations foreseen by MotionDrive, except from:

- Tab Main
  - enabling the drive
  - downloading the firmware

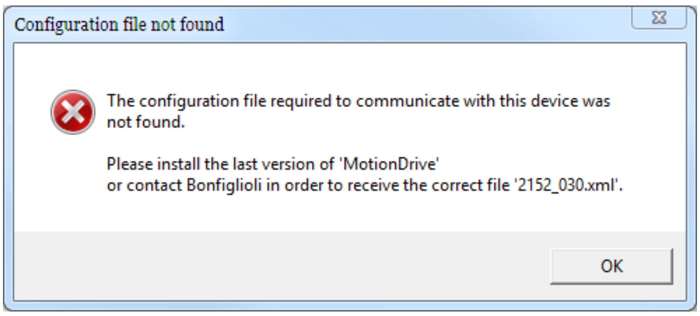
- Tab Program
  - enabling/disabling the integrated PLC
  - downloading/deleting the integrated PLC programme
  - displaying the variables of the programme in the Watch
- Drive setup
  - enabling the capture peripherals
  - running the Tuning commands of the regulation loops
- Oscilloscope
  - enabling a data capture
  - running motions by the Function Generator
  - running Tuning commands of the regulation loops.

## 12.3. Communication errors with MotionDrive

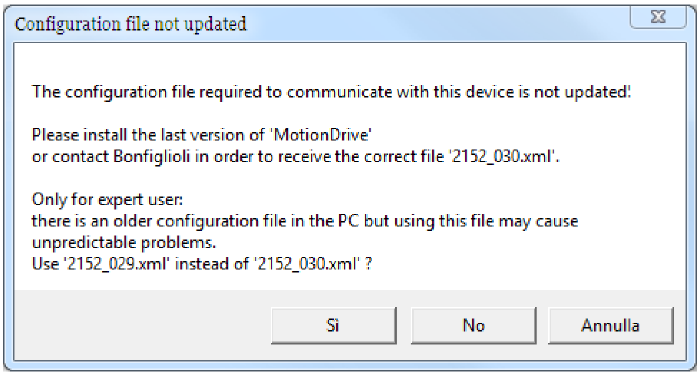
**New connection**
If during the connection the following window appears check carefully the electrical connections, the correctness of the Connection parameters and if the drive is correctly supplied; then try again.




**Configuration file not found**
If the following window appears it is necessary to update the MotionDrive Configuration files according to what is reported in Section 26.5, *Updating the Configuration File*.




**Configuration file not update**
If the following window appears it is advisable to update the MotionDrive Configuration files according to what is reported in Section 26.5, *Updating the Configuration File*.

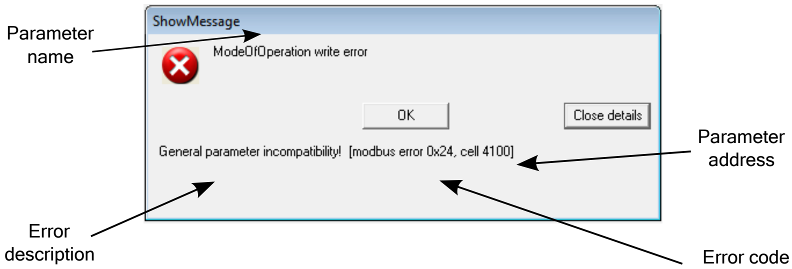



**⚠ ATTENTION**

**In case of urgency and if it is not possible to update the Configuration files, you can connect to the drive by using the Configuration file proposed only to expert users. By using not updated Configuration files, Bonfiglioli Mechatronic Research S.p.a does not guarantee the correct working of MotionDrive.**

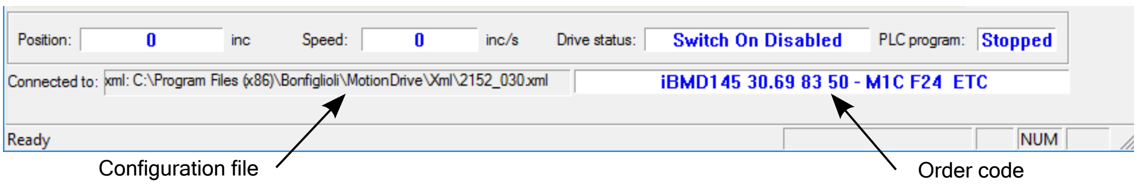
### Generic errors

When you have communication errors, MotionDrive shows some specific messages. To understand the information in the error generic message see the following picture and the Table 8.9.



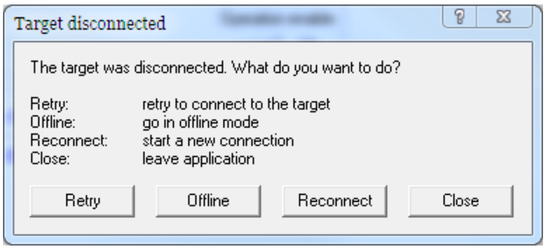

## 12.4. Connection status with MotionDrive

The connection status is made up by the Configuration File and by the ManufacturerDeviceName which can be found in the last line below in the tab Main. For further information see Section 26.5, *Updating the Configuration File* and Section 29.1, *OrderCode*.



# 12.5. Disconnection of MotionDrive

When the connection between MotionDrive and the drive is interrupted you will see the following window.




**Retry** try to connect again with the last Connection parameters used.

**Offline** go to mode Offline.

**Reconnect** open a new connection.

**Close** close the programme.

# 12.6. MotionDrive options

The MotionDrive options refer to the program working mode, particularly with its messages transmission. Access:

**Main menu** > **View** > **Options** > **General options**.

The choices done by the user by interacting with the MotionDrive message service are saved in this page and can be modified in any moment.

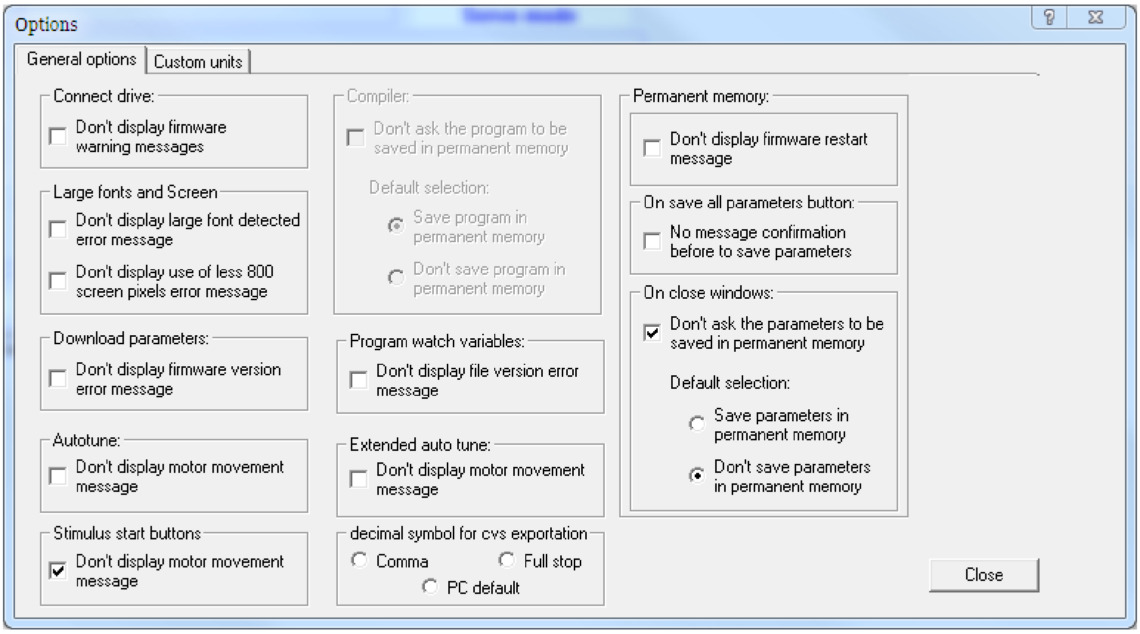



*Figure 12.2. Default configuration of MotionDrive options*

- *Connect drive:* notice of obsolete firmware connection (only for some firmwares)
- *Large fonts and Screen:* notice at the start-up in case some screen graphical options are not compatible with MotionDrive
- *Download parameters:* error notice during the download of the parameters file (only for some firmwares)
- *Autotune:* confirmation for the motion which will be run at the requested tuning command
- *Function Generator start button:* confirmation of the motion you are going to run by the function generator
- *Compiler:* notice during saving of the internal PLC programme in the permanent memory when it is downloaded in the drive (only for some firmwares)
- *Program watch variables:* control of the congruence between the variables file and the programme in the internal PLC
- *Extended autotune:* confirmation for the motion which will be run by the requested tuning command
- *Decimal symbol for cvs exportation:* choice of the separating character to export the oscilloscope data to a file
- *Permanent memory:* notice of firmware reboot when the default parameters in the permanent memory are restored (only for some firmwares)
- *On save all parameters button:* saving confirmation in the permanent memory of the modifications to the parameters
- *Drive setup close window:* automatic saving of the modifications to the parameters in the permanent memory at the Drive setup closure.

# Chapter 13
# Units of measurement

## 13.1. Units of measurement of the parameters

In the drives of the series *i*BMD, the parameters are expressed with the units of measurement of the following table. The first column shows the symbol, the second the name in detail, the third a short description.

| Unit | Name | Description |
|---|---|---|
| ns | nano second | Time unit of measurement |
| µs | micro second | |
| s | second | |
| h | hour | |
| inc | increment | Motor position unit of measurement. A complete revolution of the motor shaft for the rotary motors, or a movement equal to PolePitch for the linear motors, corresponds to the number of increments that is reported in the EncoderIncrements parameter |
| cnt | count | Unit of measurement of the position obtained by counting the number of edges of the encoder phases it is referring to |
| pulse | pulse | Quadrature encoder resolution unit of measurement [1pulse = 4count] |
| deg | degree | Position unit of measurement in sexagesimal degrees |
| inc/s | increment/second | Speed unit of measurement, calculated as incremental ratio expressed in increment |

| Unit | Name | Description |
|---|---|---|
| cnt/s | count/second | Speed unit of measurement, calculated as incremental ratio of a position expressed in count |
| rev/s | revolution/second | Speed unit of measurement, expressed in revolutions per second |
| rad/s | radian/second | Speed unit of measurement, expressed in radiants per second |
| rpm | revolution/minute | Speed unit of measurement, expressed in revolutions per minute |
| mm/s | millimeters/second | Linear velocity unit of measurement, expressed in millimeters per second |
| inc/$s^2$ | increment/$second^2$ | Acceleration unit of measurement, calculated as incremental ratio of a speed expressed in inc/s |
| rev/$s^2$ | revolution/$second^2$ | Acceleration unit of measurement, calculated as incremental ratio of a speed expressed in rev/s |
| rad/$s^2$ | radian/$second^2$ | Acceleration unit of measurement, calculated as incremental ratio of a speed expressed in rad/s |
| %IS | %I Stall | Torque unit of measurement. The 100% corresponds to the motor stall torque, considering the torque constant equal to the TorqueConstant(ForceConstant) parameter value. The sign shows the torque application direction in accordance with the Polarity parameter (it does not show if the couple is supplied or absorbed by the motor) |
| V | Volt | Voltage unit of measurement |
| A | Ampere | Current unit of measurement, values RMS |
| mH | milli Henry | Inductance unit of measurement |
| Ω | Ohm | Unit of measurement of the electric resistance |
| g | gram | Mass unit of measurement |
| g $cm^2$ | gram $cm^2$ | Inertia moment unit of measurement |
| Jm | J motor | Inertia moment unit of measurement. 1Jm corresponds to the motor inertia moment |
| °C | degree Celsius | Temperature unit of measurement |
| bit/s | bit/second | Communication speed unit of measurement |
| - | - | Dimensionless unit |

IMPORTANT

**All the torque values are obtained from a current measure multiplied per TorqueConstant(ForceConstant). If the torque constant does not correspond to the value of the TorqueConstant(ForceConstant) parameter, the obtained torque value is not correct.**

# 13.2. Revolution resolution

The revolution resolution shows the exact inc number for every revolution of the motor shaft. The revolution resolution defines the resolution describing the position, speed and accelerations, expressed respectively in inc, inc/s and inc/$s^2$. To modify the revolution resolution modify the parameter EncoderIncrements. The modification of the parameter EncoderIncrements does not imply the change of the drive performance but only the meaning of the values in which the above-mentioned variables are expressed.

Access with MotionDrive:

**Main menu** > **Drive** > **Drive setup ...** > **Motor and drive**

**Toolbar** > > **Motor and drive**

> ⚠ WARNING
>
> **If the EncoderIncrements is changed it's necessary to execute a new homing procedure and set again the SoftwarePositionLimit.**

## 13.3. Polarity

The polarity shows the direction of the motor shaft rotation in which the values increase. The signs of the speed, acceleration and torque values show if the related parameter is concordant or not with the polarity value. To modify the polarity, modify the parameter Polarity. Changing the parameter Polarity, the value of PositionActualValue does not change.

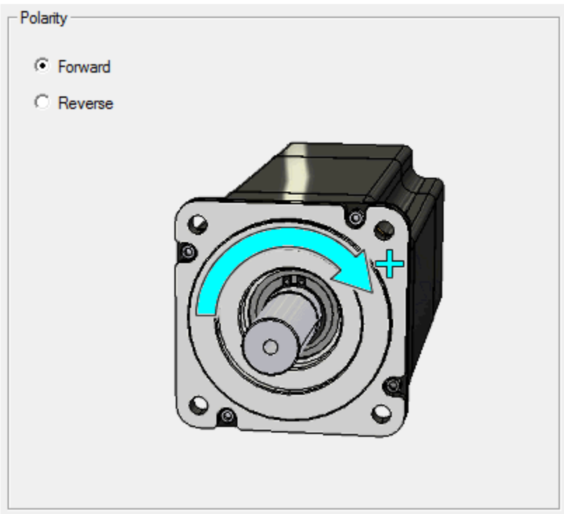
> ⚠ WARNING
>
> **If the Polarity is of Reverse type, the roles of Positive limit switch (FC +) and Negative limit switch (FC -) are reversed: Positive limit switch (FC +) behaves like Negative limit switch (FC -) and Negative limit switch (FC -) behaves like Positive limit switch (FC +). This is true both in the text of this manual and in the MotionDrive.**

> ⚠ ATTENTION
>
> **If the Polarity is changed it will be necessary to re-execute the homing procedure. By selecting the reverse mode, the limits are inverted, and PositionActualValue is consequently modified (see Figure 15.1).**

Access with MotionDrive:

**Main menu** > **Drive** > **Drive setup ...** > **Polarity**

**Toolbar** > > **Polarity**

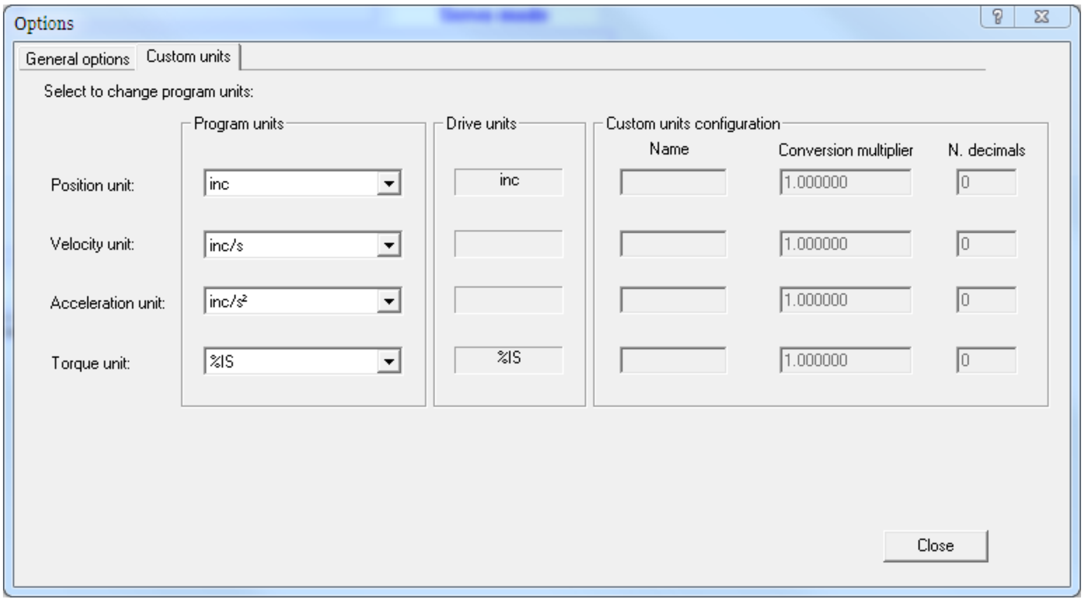


# 13.4. MotionDrive units

It is possible to select the unit of measurement with which some quantities are displayed in MotionDrive. To do this go to the page Custom units.

Access:

**Main menu** > **View** > **Options** > **Custom units**.

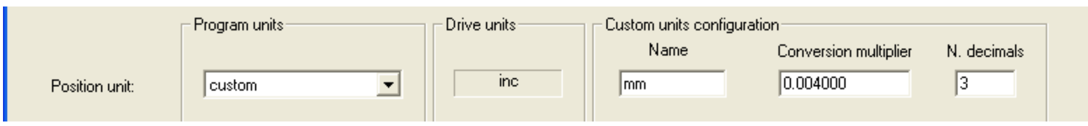



NOTE

**The settings modified in the Custom units page are linked to the MotionDrive program and not to the single drive you are connected to.**

To personalise the display of some MotionDrive variables, choose Custom in the Combo box of the Program units area and define the related fields in the box Custom units configuration. The Conversion multiplier value expresses the multiplicative factor converting the drive units in custom units.

If for example, you must match a value of 32mm to an exact revolution of the motor shaft, when the EncoderIncrements is equal to 8000inc/rev, the Conversion multiplier must be set at 32 / 8000 = 0.004 mm/inc.




***Example 13.1. Enabling the custom units for the position: 1inc = 0.004mm.***

# Chapter 14
# Power configuration

## 14.1. *i*BMD series drives supply: Y topology

The power section of the *i*BMD series drives has been projected in order to be supplied through a continue voltage that can be easily obtained from the network voltage through the AC/DC power supplier named *i*BMD-DC. It is however possible to realize an installation with a different power supplier that respects the Section 14.1.2, *"Supply with a generic power supplier"* requirements.

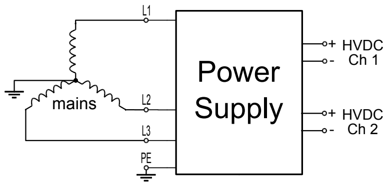



***Figure 14.1. Connection scheme of the i*BMD-DC *to the network***

**Make the connections of the network and of the grounding as required by the regulations in force: on the contrary subsists the electric shock danger.**

**The system voltage (phase-ground voltage) mustn't exceed 300VAC: in particular it may be necessary to insert an appropriate insulation transformer upstream of the drives in case of supply with not grounded networks or with networks that are grounded asymmetrically.**

### 14.1.1. Supply with *i*BMD-DC

In order to supply the *i*BMD series drives it's better to use a power supplier of the *i*BMD-DC series: for further details see the dedicated user manual.

⚠ WARNING

**Make the connection of the power supplier only after the correct sizing of the electrical wiring and the related protections and after having read the user manual of the used devices.**

**The power supply is designed for a fixed connection on a three-phase electric network of TT and TN type. The rated current of short-circuit of the electric line must be < 5kA.**

**Be sure that the protection devices on the *i*BMD-DC input have an adequate interruption capacity.**

To use this power supplier has the following advantages:

- the *i*BMD-DC has been designed to supply the *i*BMD series drives;
- the overcurrent protections are integrated;
- The DC bus voltage levelling characteristics are adequate to the *i*BMD series drives;
- Protections integration:
  - Overcharge on the DC bus outputs;
  - Braking circuit short-circuit;
  - Braking energy overcharge;
  - Charging energy overcharge;
  - Undervoltage / overvoltage / DC bus excessive ripple;
  - Power and control section overtemperature;

The *i*BMD-DC series suppliers are provided of 2 DC bus outputs; it's possible to maintain these outputs separated by making 2 branches or by connecting them in parallel: in this last configuration it's necessary to adequate the cables flow, downstream of the union of the cables to the maximum suppliable current of the supplier.

A semplified scheme for the *i*BMD connection to the DC bus generated by the *i*BMD-DC is reported in Figure 14.2: this scheme represents the *i*BMD-DC output separated connection (HVDC CH1 and HVDC CH2); the DC bus cables can therefore be connected to a terminal box, represented by the “A” and “B” details in figure.

A wiring example of HVDC CH1 to the terminal box is reported in Figure 14.3.

The cable that's identified by the orange colour allows to make in the same way the connection to the DC bus and to the control voltage for all the *i*BMD drives; furthermore this cable allows to connec to the *i*BMD drive 1 input channel for the STO inputs control and the IN9 input.

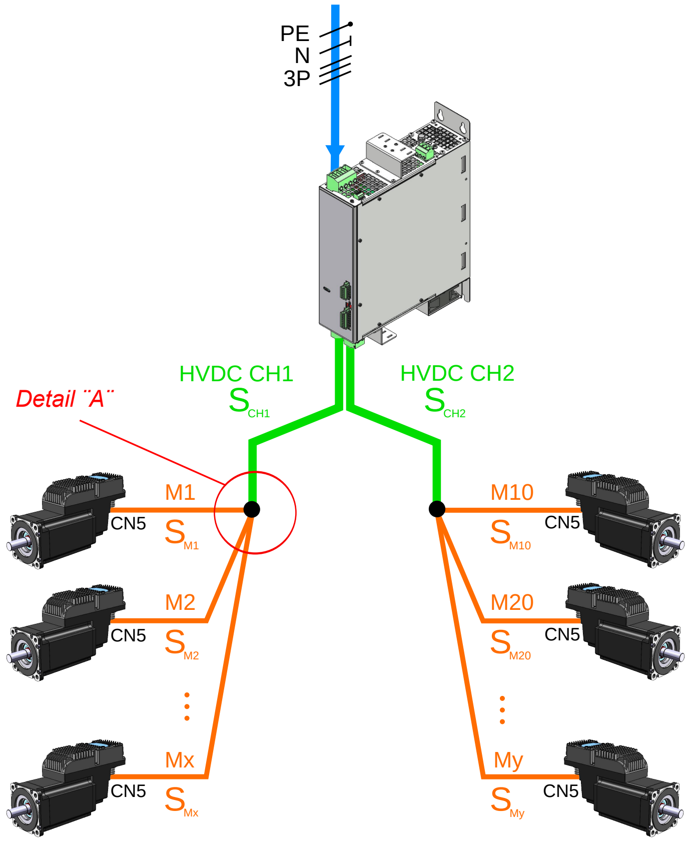



***Figure 14.2. Scheme of the** i*BMD-DC ***conection to** i*BMD ***drives***

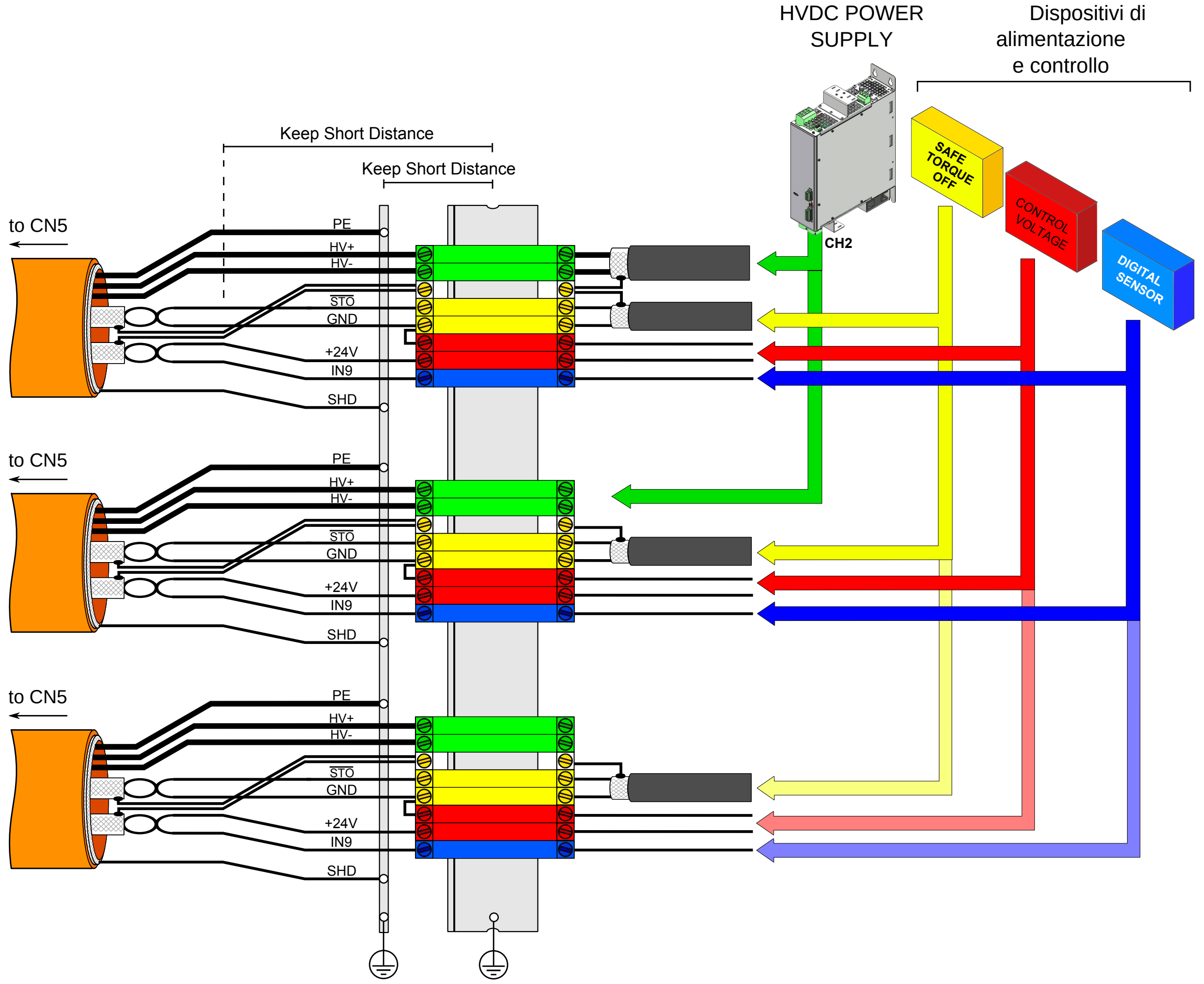



***Figure 14.3. Scheme of the*** *i*BMD-DC ***conection to*** *i*BMD ***- particular A***

The duct drives supply network topology and the cables selection must be done according to the prescriptions contained in the *i***BMD-DC** manual and by respecting the current regulations. In particular it's necessary the conformity to the IEC 60364-5-52 (Low-voltage electrical installations – Part 5-52: Selection and erection of electrical equipment – Wiring systems) and the IEC 60364-4-43 (Low-voltage electrical installations – Part 4-43: Protection for safety – Protection against overcurrent).

It's important that during the realization of the duct the maximum drives connection cables length is respected: for further informations see Section 7.2.5, *“Cables for i*BMD”.

## 14.1.2. Supply with a generic power supplier

The drives can be supplied by a DC bus generated by a generic supplier, as indicated in Figure 14.4: in this case it's necessary to make the complete sizing of the ducts and of the protections.

⚠ WARNING

**Make the connection to the network only after the correct sizing of the electrical wiring and the related protections and after having read the user manual of the used devices.**

**The drive is designed to be used with a supplier connected to an electric network of three-phase or single-phase TT or TN type (depending on the performances required to the motor).**

In particular, to supply the drives with a generic supplier, over the requirements of Section 14.1.1, "*Supply with i*BMD-DC", it's necessary to guarantee the conformity to the requirements reported in Table 14.1.

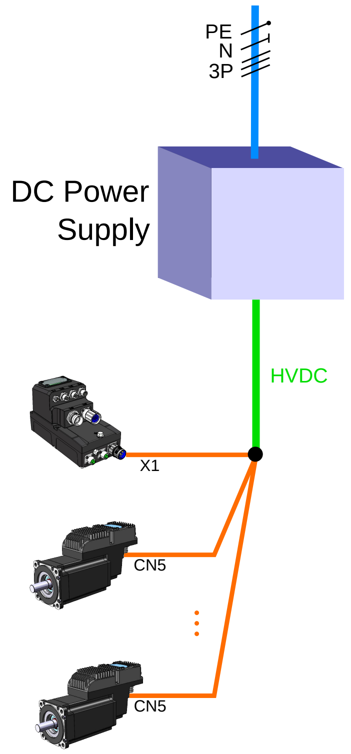



*Figure 14.4. Scheme of connection to a generic supplier to i*BMD *drives*

### Supplier electrical requirements

| Supplier features | Symbol | Requirement |
|---|---|---|
| Output voltage range | Vdc | Minimum 275VDC; Maximum 730VDC[a] |
| Output voltage ripple | DVdc | The supplier must guarantee a levelling of the output voltage adequate to the performance requirements[b]. |
| Maximum output voltage while braking | $Vdc_{max}$ | 785VDC[c] |
| Soft start | - | It's necessary a soft start system through which the supplier limits the capacitors charging current at the turn on so that to avoid overcurrent and overvoltage peaks on the DC bus. |
| Output current | Idc | The supplier must provide a rated and peak currents adequate to the type and the absorption of each single drive[d] and to the coincidence factor |
| Protection from overcurrent and short-circuit of the output. | - | The supplier must be provided of internal protections, adequate to the installation, against the short-circuit and the overvoltage on the DC bus. Otherwise protect the cables of the DC bus with external devices (es. fuses) adequate to the load, to the installation electrical features and to the requirements of the current regulations. |

[a]The supplier must be installed in a TT or TN system network. The system voltage must be equal or lower than 300 VAC. The output voltage of the DC bus must be adequate to the electrical features of the drives that are connected

in order to guarantee to reach the required performances of velocity and torque. Tipically a decreasing of the DC bus voltage determines a proportional reduction of the motor rotation.

[b]A decreasing of the DC bus voltage determines a proportional reduction of the motor rotation velocity.

[c]The maximum voltage must be adequate to the electrical features of the drives that are connected: in particular it's necessary to guarantee a margin to avoid the DC bus overvoltage Fault.

[d]The maximum absorption of each drive is related to the I2T parameter (for further information see Section 14.6, *I2T*).

***Table 14.1. Supplier electrical requirements for compatibility with i*****BMD *drives***

## 14.2. Supply voltages

The drives of the *i*BMD series have two separated sections, control and power, that must be separately supplied with direct voltages (galvanic isolation). Check that the values of the voltage, that are reported in the Chapter 5, *Technical features*, are respected.

While choosing the voltage of the DC bus (power section supply) you need to consider:

1. any possible voltage changes in order to avoid any unwanted notifications of Faults or Warnings
2. the drive cannot dissipate the energy of regeneration (see Section 14.3, *Regeneration*)
3. the drop in the motor performances, decreasing the supply voltage

ADVICE

**Supplying the power section with the rated voltage value.**

When the supply voltage of the control section decreases below the lowest threshold, the drive is disabled. In the previous chart you can find the value of this threshold.

There is a threshold, on brake-equipped motors, causing the drive Fault when the supply voltage of the control section is not sufficient to ensure the safe brake release. In the previous chart you can find the value of this threshold.

NOTE

**DC bus under voltage it can be of self-restoring type. Furthermore you can choose if enabling or not the Fault in case of Logic voltage error. Fur further details please see Chapter 24, *Fault and Warning*.**

## 14.3. Regeneration

The regeneration is a drive working phase in which the drive brings energy to the DC bus during the motor deceleration. If this energy is not absorbed or dissipated, the DC bus voltage can increase and cause the drive Fault. The drives *i*BMD are not enabled to dissipate this energy internally. In order to dissipate the regeneration energy, it is necessary to supply the DC bus with a bidirectional power supply or with brake resistances, which can limit the DC bus voltage and let the drive work normally also during the motor deceleration.

ADVICE

**To evaluate the level of the drive regeneration, it is necessary to take into account the peak of kinetic energy generated by the motor during its deceleration and the total ener-**

**gy continuously generated. These data are fundamental in order to choose the DC power supply. It is advisable to read the manual and the power supply technical documents.**

## 14.4. Drive currents

The drives of the series *i*BMD regulate the motor current depending on the torque requests and speed. The parameters related to the drive currents can be found in the following table:

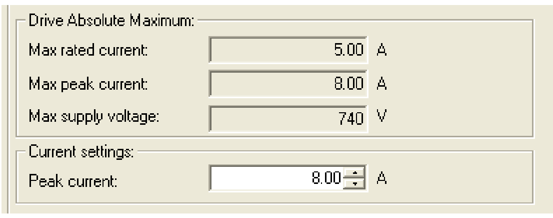
| Parameter | Description |
|---|---|
| MotorStallCurrent | Motor stall current |
| MotorPeakCurrent | Motor peak current |
| MaxRatedCurrent | Drive nominal current, power section |
| MaxPeakCurrent | Drive peak current, power section |
| UserPeakCurrent | Peak current set by the user to limit the current supply to the motor |
| **NominalCurrent** | Real nominal current: lower value between MotorStallCurrent and MaxRatedCurrent |
| **PeakCurrent** | Real peak current: lower value than MotorPeakCurrent, MaxPeakCurrent and UserPeakCurrent |
| ActualMotorCurrent | Actual motor current |
| ActualFieldCurrent | Actual motor current, field component |
| ActualTorqueCurrent | Actual motor current, torque component |
| OverCurrentAValue | Current of the motor A phase in condition of Power or motor over current |
| OverCurrentBValue | Current of the motor B phase in condition of Power or motor over current |
| OverCurrentCValue | Current of the motor C phase in condition of Power or motor over current |
| RMSMotorCurrent | Motor RMS current |
| RMSMotorCurrentFilter | Filtering time to get the motor RMS current |

The only writable parameter of the previous chart (after RMSMotorCurrentFilter) is UserPeakCurrent and it is used to limit the current supplied to the motor (see Section 19.1, *Electricity limit*).

Access with MotionDrive:

**Main menu** > **Drive** > **Drive setup ...** > **Motor and drive**

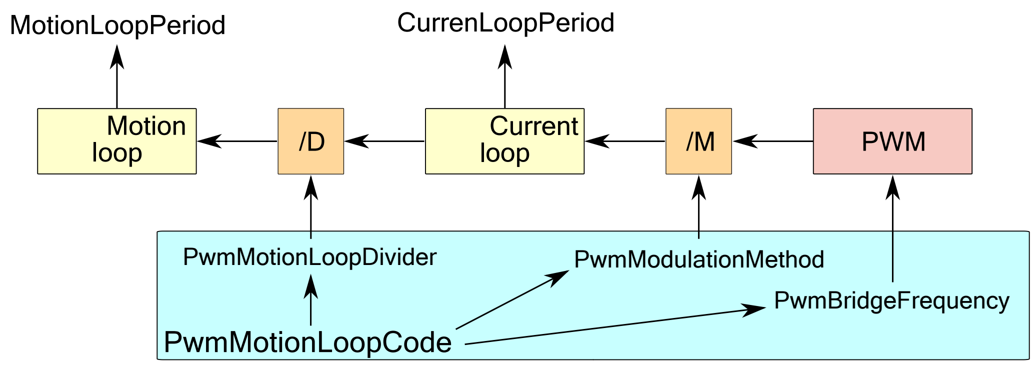
**Toolbar** > > **Motor and drive**




## 14.5. Power PWM

In the drives of the series *i*BMD it is possible to modify the sampling frequency of the three-phase bridge steering the motor currents and the loops sampling period Increasing the sampling frequency of the three-phase bridge can increase the drive dynamic performances as well as the losses in the power section and the power section heating. Vice versa, decreasing the sampling frequency of the three-phase bridge can damage the drive dynamic performances but decreases the power section heating. In order to modify these variables, use the parameters in the following table:

| Parameter | Description |
|---|---|
| PwmMotionLoopCode | Unique code to set the three-phase bridge frequency and the loops period (it automatically sets the PwmBridgeFrequency, PwmModulationMethod and PwmMotionLoopDivider parameters values) |
| MotionLoopPeriod | Motion loop period |
| CurrentLoopPeriod | Current loop period |




The current loop period can be obtained through the following expression:

$$\text{CurrentLoopPeriod [s]} = \frac{1}{\text{PwmBridgeFrequency} \times \text{PwmModulationMethod}}$$

while the motion period can be obtained with the following expression:

$$\text{MotionLoopPeriod [s]} = \text{CurrentLoopPeriod [s]} \times \text{PwmMotionLoopDivider}$$

ADVICE

**Modify these parameters only if strictly needed. Do not go below 100µs with MotionLoopPeriod. After the modification of these parameters it is necessary to tare the loops again.**

## 14.6. I2T

For a limited time period, the current supplied to the motor can be more than *NominalCurrent* (**overload**). To protect the drive motor and power section during the overload periods, the drive controls the energy transferred to the motor and can limit the current. The parameter I2TValue shows the level of the energy transfer according to the following table:

| I2TValue | Drive energy status. |
|---|---|
| 0 | The drive is not overloaded |
| >0 and <100 | The drive is overloaded |

| I2TValue | Drive energy status. |
|---|---|
| >50 | The drive is overloaded and too much exploited: application in case of working critical conditions |
| 100 | The drive has reached the highest level of overload and the current falls at the value NominalCurrent (only if the limitation does not cause any Fault) |

The maximum energy that the drive can supply in overload condition can be found in the parameter UserMaxI2T. The value is limited by the parameter DriveMaxI2T.

The value UserMaxI2T is directly connected to the product between UserPeakCurrent and I2TTime. So for example it is possible to oversupply a motor with 20A for 1s or with 10A for 4s, by keeping limited the value of UserMaxI2T .

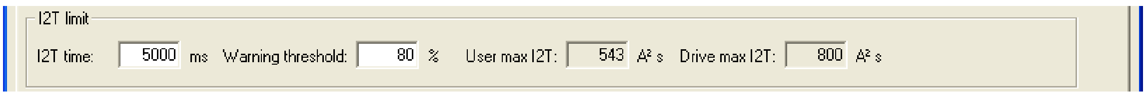
To set correctly the parameters of the I2T follow these instructions:

1. choose the value of UserPeakCurrent as current limit used to overload the motor and the drive (with MotionDrive, "Motor and drive" page of Drive Setup)
2. choose the value of I2TTime as maximum current overload time PeakCurrent
3. check that UserMaxI2T is lower than DriveMaxI2T; if it is not so decrease UserPeakCurrent and/or I2TTime
4. choose I2TWarningThreshold equal to the level of I2TValue in which you wish to be warned through the I2T Warning threshold reached Warning
5. consider if enable the Fault I2T limit reached when I2TValue reaches the 100% (with MotionDrive, Errors page of Drive Setup).

Access with MotionDrive:

**Main menu** > **Drive** > **Drive setup ...** > **Limit and windows settings**

**Toolbar** >  > **Limit and windows settings**




| Active errors | Led L4 drive *i*BMD | I2TValue | ActualMotorCurrent |
|---|---|---|---|
| - | GREEN, 1 FLASH | 0 | 0 |
| | GREEN, ON | | > 0 and ≤ NominalCurrent |
| | ORANGE, ON | > 0 and < I2TWarningThreshold | ≤ PeakCurrent |
| I2T Warning threshold reached | ORANGE, BLINKING | ≥ I2TWarningThreshold and < 100 | |
| I2T limit reached | RED, ON | ≥ 90 | ≤ NominalCurrent |

***Table 14.2. Status of the I2T.***

⚠ ATTENTION

**If the Fault for I2T limit reached is active, the current is limited but the motor motion stops and the drive enters the Fault status.**

IMPORTANT

**In the boot and firmware start-up phases of the *i*BMD drives, the above description of the led is no more valid. Please refer to Chapter 26, *Software updating***

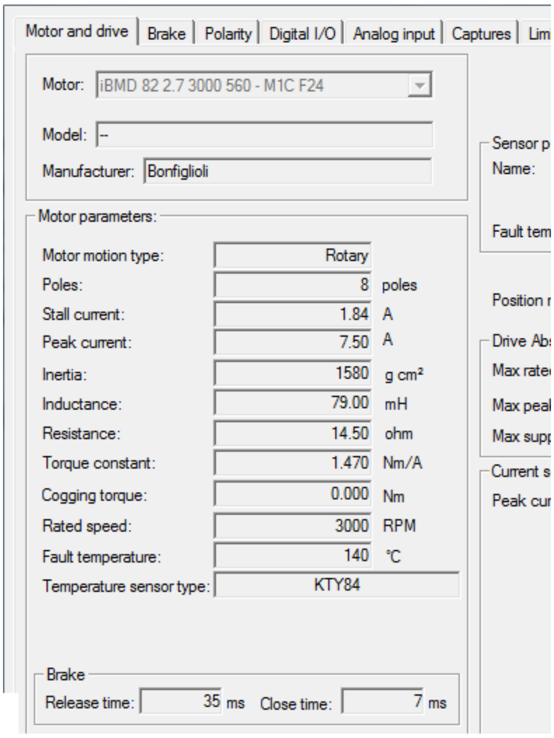
# Chapter 15
# Configuring motor, sensors and brake

## 15.1. Motor parametrization

To parametrize the motor follow the order of the tabs that are presents in the *Drive Setup* window

Access with MotionDrive:

> **Main menu** > **Drive** > **Drive setup ...**
>
> **Toolbar** > 




NOTE

**In the *i*BMD drives the motor is already selected and cannot be modified.**

The parametrization consists of the following operations:

1. if necessary, define the parameters in the *Motor and drive* page;
2. continue with the parameterisation of the drive functionalities in the other pages;
3. execute the loop tuning (see Chapter 20, *System tuning*);
4. Save the parameters in the internal memory or in a parameters file (see Chapter 21, *Saving, restoring or cloning the drive configuration.*).

## 15.2. Sensor of the feedback position

NOTE

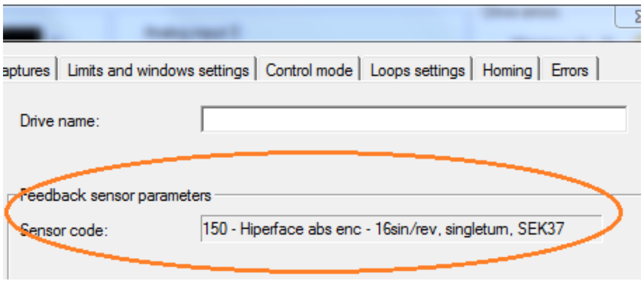
**In the *i*BMD drives the feedback sensor is already selected and cannot be changed.**

The code that selects the installed feedback sensor type is in the FeedbackSensorCode parameter. The order codes of the available feedback sensors are listed in the Table 29.1. The feedback sensor type is reported in the *Motor and Drive* tab of the *Drive Setup*.

Access with MotionDrive:

**Main menu** > **Drive** > **Drive setup ...** > **Motor and drive**

**Toolbar** >  > **Motor and drive**




**Phasing types**
Four types of phasing are defined:

- *Phasing type 0 (phased)*: of this cathegory are part the sensors that are installed on the motor with a mechanical phasing that's defined and recognized by the drive. For these sensors it's not necessary to execute the phasing;
- *Phasing type 1 (pre-phased)*: of this cathegory are part the sensors that are installed on the motor with a mechanical phasing that's defined but not recognized by the drive. For these sensors it's necessary to execute the phasing at least one time; The phasing informations can be exported to other drives through the parameters file without having to repeat the phasing procedure, but it has to be installed the same sensor type and phasing type must be the same. The correct phasing must be guaranteed and verified during the feedback sensor mechanical installation;
- *Phasing type 2 (not pre-phased)*: of this cathegory are part the sensors that are installed on the motor without a defined mechanical phasing. For these sensors it's necessary to execute the phasing on each single motor at least one time. Once the phasing has been executed the phasing informations can be saved in the drive, without having to repeat it at every startup;
- *Phasing type 3 (not savable)*: of this cathegory are part the sensors that are installed on the motor without a defined mechanical phasing and that don't provide any absolute information about the shaft position related to the motor revolution. For these sensors it's necessary to execute the phasing on each single motor at every drive startup.

NOTE

**In the motors that mount Hiperface sensors the phasing is saved directly by the encoder, so, once it has been executed, it is no more necessary to repeat it even if that motor is connected to another drive of the same series.**

## 15.3. Feedback sensor phasing check.

To check the feedback sensor phasing, even in case of sensor anomalies, follow the instructions contained in the *Test feedback sensor phasing* field.

Access with MotionDrive:

**Main menu** > **Drive** > **Drive setup ...** > **Phasing**

**Toolbar** >  > **Phasing**

- through the FeedbackSensorPhasingAngleTest parameter the angle position on which the test will be executed is specified;
- through the Start button the test starts (to execute the test through System Manager use the command 5310);
- through the field on step 4 (that returns the value contained in the FeedbackSensorPhasingAngleError parameter) the phasing angle error is read. If the value of this parameter exceeds a drive internally defined threshold, it is signaled the error 5301 of the SysMngError.

ADVICE

**Repeat the operation by trying different values of FeedbackSensorPhasingAngleTest.**

**Check the phasing through System Manager**

In order to manage the phasing test from controller through the System Manager commands, it is sufficient to perform the following procedure:

- Start: write the value of FeedbackSensorPhasingAngleTest first and then execute the command 5310 of the SysMngCommand (Test phasing of feedback position sensor)
- Save: execute the command 5320 of the SysMngCommand (Save phasing of feedback position sensor)
- Disable: as every operative mode according to the CiA-402 specifications

ADVICE

**Repeat the operation by trying different values of FeedbackSensorPhasingAngleTest.**

## 15.4. Absolute feedback position sensor

IMPORTANT

**if a MULTITURN absolute encoder is set, the FeedbackSensorAbsMode parameter is set to 1 by default, consequently the drive will effectively manage it as an absolute sensor. Otherwise if a SINGLETURN absolute encoder is set, the FeedbackSensorAbsMode parameter is set to 0 by default, consequently the drive will manage it as an incremental sensor. To use the sensor as absolute, set to 1 the FeedbackSensorAbsMode parameter.**

The provided absolute encoders are listed in the following table.

| Description | Functioning range | Accuracy[a] | Precision (repeatability)[b] |
|---|---|---|---|
| Absolute encoder Hiperface multiturn SKM36 128sin/rev. | 4096rev | 1/10000 rev | 1/40000 rev |
| Absolute encoder Hiperface singleturn SKS36 128sin/rev. | 1rev | 1/10000 rev | 1/40000 rev |
| Absolute encoder Hiperface multiturn SEL37 16sin/rev. | 4096rev | 1/4000 rev | 1/10000 rev |

| Description | Functioning range | Accuracy[a] | Precision (repeatability)[b] |
|---|---|---|---|
| Absolute encoder Hiperface singleturn SEK37 16sin/rev. | 1rev | 1/4000 rev | 1/10000 rev |

[a] *"Accuracy"* is the proximity of measurement results to the true value.
[b] *"Precision"* is the repeatability or reproducibility of the measurement in the same conditions.

***Table 15.1. Supported absoluted feedback sensors***

To align the absolute sensor position with a specific mechanical position it's necessary to execute a homing procedure (see Section 22.19, *Homing Mode*).

The absolute position sensors keep the position value even if the drive is turned off. The maintenance of the position at the turn on of the drive is guaranteed only within the operating range of the sensor.

⚠ WARNING

**If the sensor works as an incremental encoder (parameter FeedbackSensorAbsMode equal to 0), after the EncoderIncrements and/or Polarity modification, it will be necessary to re-execute the homing procedure.**

**Functioning of the absolute sensor: EncoderIncrements, Polarity and homing**

In the Figure 15.1 is shown an example of how the homing procedure (with HomingAbsRangeMode parameter equal to 0) and the Polarity act on the feedback absolute sensor position (the FeedbackSensorAbsMode parameter must be equal to 1 in order to work in this way). In the example an offset is set between the sensor physical position and the one that's read by the drive.

- On the top the machine physical range is represented (red reference) compared with the position sensor range (blue triangle)
- In the first image the read position coincides with the sensor physical position. In this condition the machine physical range is not completely included in the sensor range, therefore the absolute position is not guaranteed
- In the second image it's shown how the sensor range is shifted after the homing procedure, of 350000[inc] in the specific case (this offset is saved in the drive permanent memory and it will no longer be necessary to execute the homing procedure). Through this operation the machine physical range is completely included in the sensor range, therefore the absolute position is always guaranteed.
- In the third image it's shown how the Reverse function (see Polarity) reverses the position reference between the 0 and the sensor range maximum allowed value

⚠ ATTENTION

**If during the functioning the motor exceeds, in positive or negative direction, the sensor working range, the drive is able to correctly reconstruct the position, but if the drive is turned off while the position read by the sensor is out of its working range, the position that has been read on the turn on of the drive will not be consistent with the executed homing and it will be necessary to re-execute the homing procedure.**

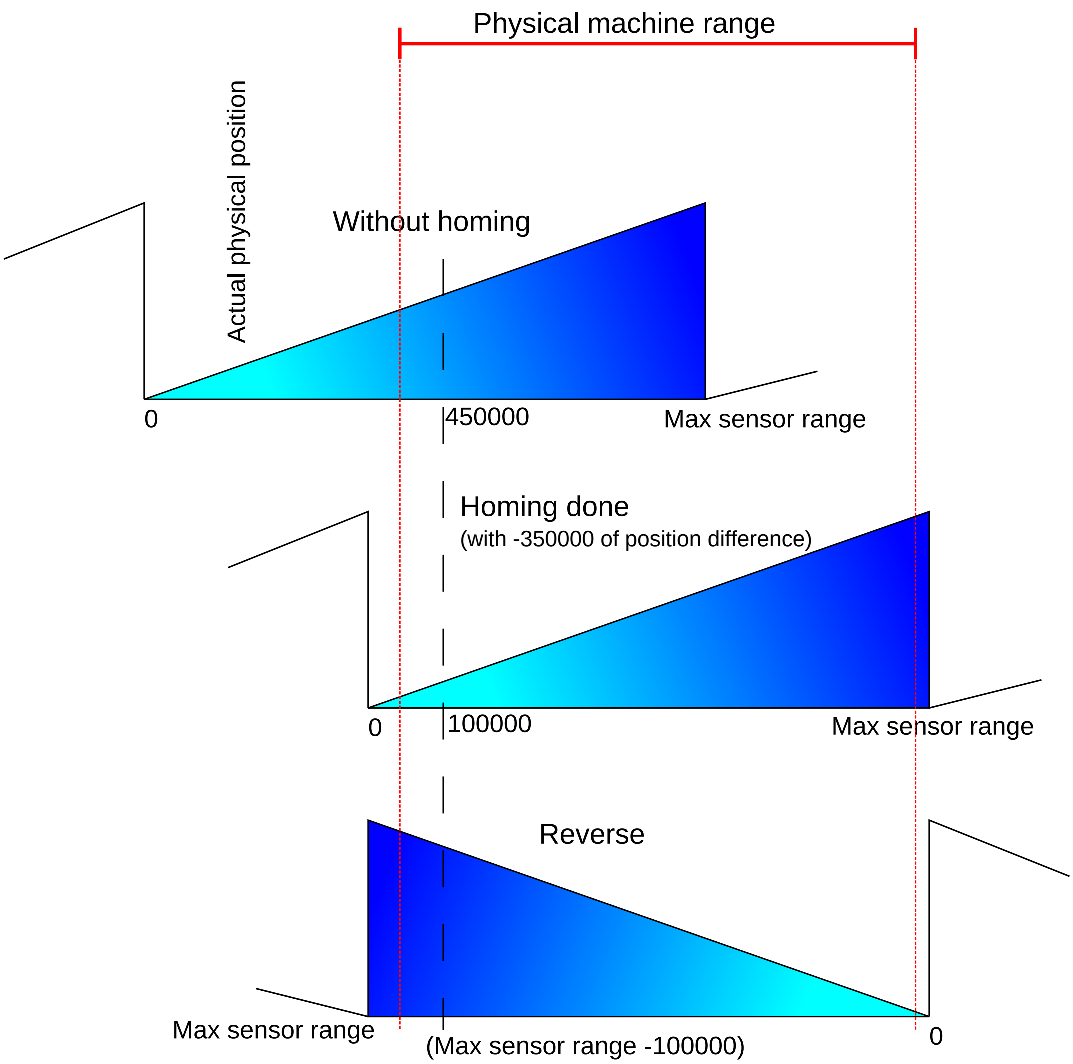



*Figure 15.1. Functioning range of absolute sensors, position preset and reverse.*

# 15.5. Auxiliary position sensor

The drive can acquire an external physical encoder that can be used for example to control the position of another axis or in relation to the motion function. The incremental encoder is the only managed auxiliary position sensor type.

The drive internally provides even a virtual (simulated) encoder that generates a position that varies with a constant velocity that can be configured by the user.

If you want to use a physical encoder, this must be connected as follows (for further information about wiring please see Chapter 7, *Electrical connections, leds and dip switches*):

| Encoder signal | iBMD HW input | Notes |
|---|---|---|
| Quadrature Input ChA (Ch A) | I/O 0 (connector CN4) | Compulsory |
| Quadrature Input ChB (Ch B) | I/O 1 (connector CN4) | Compulsory |
| Quadrature Input Index (Idx) | I/O 2 (connector CN4) | Optional |

To acquire the physical auxiliary position sensor, the digital input must be configured as described in Chapter 16, *Digital inputs and outputs*.

Here follows the list of the parameters that are provided to manage the master encoder functionalities:

**Physical master encoder:**

- RealEncoderPosition
- RealEncoderVelocity
- RealEncoderPolarity

**Virtual master encoder:**

- VirtualEncoderPosition
- VirtualEncoderVelocity
- VirtualEncoderRunStop

**Selected master encoder:**

- AuxiliaryEncoderPosition
- AuxiliaryEncoderSpeed
- AuxiliaryEncoderSelector

NOTE

**If a physical master encoder is used and RealEncoderPolarity value is 0 (it's set on forward), RealEncoderPosition increments when Quadrature Input ChA (Ch A) anticipates Quadrature Input ChB (Ch B).**

## 15.6. Brake

The brake installed in the *i*BMD drives is a holding brake. When the brake is configured, it's automatically managed by the drive, contemporary to the enable/disable operation. The brake parameters are in *Drive setup*.

| Brake electrical features | |
|---|---|
| Rated voltage | 24Vdc |
| Current absorbed by the control section | 1.5A with control voltage = 24Vdc<br>0.85A with control voltage = 48Vdc |
| Internal fuse | 2A-T not replaceable |
| Protections | Protection from short-circuit |

***Table 15.2. Brake features***

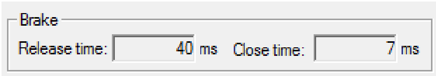
NOTE

**In the *i*BMD drives the brake is already configured therefore no parametrization is needed.**

Access with MotionDrive:

**Main menu** > **Drive** > **Drive setup ...** > **Motor and drive**

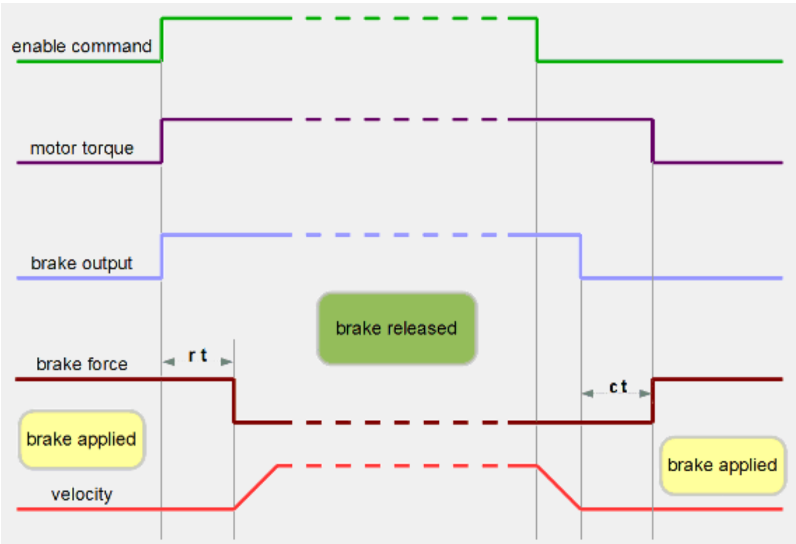
**Toolbar** >  > **Motor and drive**



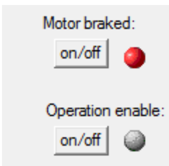

The parameter BrakeReleaseTime shows the time elapsed between the command to delay (disabling) the brake, occurring together with the motor enabling, and the moment when the drive accepts any motion commands. During this time interval the brake is exercising its braking power and the shaft motion could damage the brake.

The BrakeCloseTime parameter shows the time between the brake blocking command (activation), that happens after the motor disabling command, and the moment in which the drive cuts off the current to the drive. During this time, the brake is not exercising its brake power; if during this time the motor runs out of power the position could not be guaranteed.

See the timing in the following figure.




The brake can be manually commanded only when the motor is disabled. If the motor is enabled, it's not possible to activate the brake. The command can be reached through the BrakeStatus parameter  or through the dedicated button that's present in the *Main* Tab of MotionDrive.




NOTE

**The brake is active when it is not powered, so when it produces braking force (red led is on).**

Chapter 16

# Digital inputs and outputs

## 16.1. Digital inputs and outputs of the *i*BMD drives

In the *i*BMD drives are provided the following optoisolated digital inputs and outputs:
On the CN4 connector (M23 male 19 poles) there are:

- 5 PNP digital inputs (24Vdc)
- 3 PNP digital outputs (24Vdc; max 300mA)
- 1 PNP bidirectional digital [with configurable direction]

On the supply connector CN5 ( M23 male 8 poles) there is:

- 1 PNP digital input (24Vdc)

*It's therefore possible to have up to 7 inputs[1] and up to 4 PNP digital outputs[2] 24Vdc.*
On the CN4 connector there are also:

- 3 differential digital IN/OUT (type +5V/Line driver), isolated from the power section, [usable for example as master encoder input or step-dir input].

[1]In this case the PNP digital outputs number is 3.
[2]In this case the PNP digital inputs number is 6.

**Provided digital I/O for *i*BMD drive**

| Name | Type of resource / logic | Details | Default |
|---|---|---|---|
| I/O 0 | Bidirectional, differential, Line Driver | Configurable functionality, connections: pin 13 (I/O+) and pin 12 (I/O-) of CN4 | GPIN |
| I/O 1 | Bidirectional, differential, Line Driver | Configurable functionality, connections: pin 14 (I/O+) and pin 1 (I/O-) of CN4 | GPIN |
| I/O 2 | Bidirectional, differential, Line Driver | Configurable functionality, connections: pin 5 (I/O+) and pin 2 (I/O-) of CN4 | GPIN |
| I/O 3 | Bidirectional, PNP, 24V | Configurable functionality, connection: pin 10 of CN4 | GPIN |
| In 4 | Input, PNP, 24V | Configurable functionality, connection: pin 15 of CN4 | GPIN |
| In 5 | Input, PNP, 24V | Configurable functionality, connection: pin 19 of CN4 | GPIN |
| In 6 | Input, PNP, 24V | Configurable functionality, connection: pin 18 of CN4 | GPIN |
| In 7 | Input, PNP, 24V | Configurable functionality, connection: pin 17 of CN4 | GPIN |
| In 8 | Input, PNP, 24V | Configurable functionality, connection: pin 8 of CN4 | GPIN |
| In 9 | Input, PNP, 24V | Configurable functionality, connection: pin C of CN5 | GPIN |
| Out 4 | Output, PNP, 24V | Configurable functionality, connection: pin 16 of CN4 | GPOUT |
| Out 5 | Output, PNP, 24V | Configurable functionality, connection: pin 9 of CN4 | GPOUT |
| Out 6 | Output, PNP, 24V | Configurable functionality, connection: pin 17 of CN4 | GPOUT |
| /STO | Input, PNP, 24V | Not configurable. Connection: pin A of CN5 | /STO |

***Table 16.1. Description of the iBMD drive digital I/O***

**Description of the terms used in the previous chart**

*Bidirectional:* The resource cannot be configured to work as an input or output.

*Input:* The resource works as input but not as output.

*Output:* The resource works as output but not as input.

*Differential:* The status of the resource is linked to the difference of potential between two pins (Section 7.2.4, *iBMD drive connectors*).

*PNP:* The status of the resource is linked to the current voltage value as to the common mass (Section 7.2.4, *iBMD drive connectors*).

*Configurable functionality:* The resource can be configured to work as "Generic Input" (GPIN) or "Generic Output" (GPOUT), so the status of the resource can be read or written by the user through the DigitalInputs and PhysicalOutputs parameters, or it can be configured to run some special functions (see Section 16.4, *Functionalities*), so its status is directly managed by the drive.

*/STO :* See Chapter 6, *STO safety function: Safe Torque Off* .

**Inputs (Table 16.2), differential IN/OUTs (Table 16.3) and outputs (Table 16.5) electrical characteristics.**

| 24V PNP DIGITAL INPUTS CHARACTERISTICS | |
|---|---|
| Inputs maximum n° | 7 |
| Galvanic isolation | YES, through optoisolators |
| **In/Out3, In4, In5, In6, In7, In8, In9** | |

| 24V PNP DIGITAL INPUTS CHARACTERISTICS | |
|---|---|
| Input type | PNP |
| Input voltage | • Nominal : +24Vdc<br>• LOW signal (physical status 0) : -30 ÷ +3Vdc<br>• HIGH signal (physical status 1) : +15 ÷ +30Vdc |
| Input current (typical) with Vin = 24Vdc | • 3,3 mA (IN8 and IN9 excluded)<br>• 7 mA (IN8 and IN9) |
| Maximum allowed current on IN5 if configured as SGND | 2A |
| HW propagation delay (IN8 and IN9 capture inputs)[a] | • Typical : rising edge=6,8μs , falling edge=1μs<br>• Minimum : rising edge=3,6μs , falling edge=1μs<br>• Maximum: rising edge=12μs , falling edge=1μs |
| Jitter on the SW acquiring of IN8 and IN9 capture inputs | max 1μs |
| HW propagation delay (generic input) | Typical : rising edge=15μs , falling edge=170μs |

[a]24V amplitude step signal

***Table 16.2. Digital inputs electrical characteristics***

| DIGITAL IN/OUT CHARACTERISTICS (differential line drive type) | |
|---|---|
| Type | Differential IN-OUT (line driver/line receiver) |
| Galvanic isolation | YES, towards the power section |
| Maximum outputs n. | 3 |
| Maximum inputs n. | 3 |
| **In/Out0, In/Out1, In/Out2** | |
| *OUTPUT characteristics* [a] | |
| Differential output voltage | min: 2V; max: 3,3V (with a 50Ω load) |
| *INPUT characteristics* [b] | |
| Differential input voltage | max: -5V ÷ +5V (with enabled termination) |
| Common-mode voltage | -7V ÷ +12V |
| Typical input resistance | 125kΩ (when the termination resistance is disabled) |
| Typical input current | 110μA |
| Differential threshold voltage $V_{thd}$ (input) | 0,2V |
| Integrated termination resistance[c] | typ: 120Ω |
| Input maximum frequency (duty cycle: 40%÷60%) | 300KHz |

[a]If they are configured as outputs.
[b]If they are configured as inputs.
[c]Can be activated by software command.

***Table 16.3. Digital IN/OUT electrical characteristics***

⚠ NOTICE

**The In/Out0, In/Out1, In/Out2 inputs are differential and must NOT be connected with "24V" level signals. Please see the applicative diagrams of the CN4 connector. It's recommended to respect the maximum differential voltage and to report this voltage to the “GND_5V” ground [pin 6 of CN4].**

NOTE

**When the voltage that's applied to the differential input is greater than the differential threshold voltage $V_{thd}$ (see Table 16.3), then the related physical status in the *i*BMD is 1. When instead the voltage that's applied to the differential input is minor than $-V_{thd}$, then the related physical status in the *i*BMD is 0. For values included in the $\pm V_{thd}$ range the physical status of the input is not guaranteed.**

| /STO INPUT FEATURES | |
|---|---|
| /STO input | See /STO input electric features. |

***Table 16.4. /STO input electric features***

| DIGITAL OUTPUTS CHARACTERISTICS | |
|---|---|
| **In/Out3[a], Out4, Out5, Out6** | |
| Output type | PNP |
| Output maximum n° | 4 |
| Galvanic isolation | YES, through optoisolators |
| Supply voltage | 24V (internally obtained from the 24V that are presents on CN5) |
| Maximum output current (for each output)[b] | 300mA |
| Voltage with OFF output | <1V |

[a]If configured as output.

[b]That limit is true even if the output is configured as S24V (simulated 24V)

***Table 16.5. Digital outputs electrical characteristics***

NOTE

**The maximum output current declared on Table 16.5 is referred to each output, therefore it's possible to have a maximum overall absorption of about 24V (with all 4 outputs enabled and with the maximum connected load) equal to 1200mA. Inside the system a protection useful in case of the overcoming of this absorption limit is implemented. This protection disable all the outputs (even if the greater part of the absorption is due to only one of these). Therefore pay particular attention because the overcurrent on a single output may cause a Fault that causes the switch off of the others outputs too.**

## 16.1.1. I/O functionality for *i*BMD drive

Here follows the functionalities given to the resources of I/O of the drive *i*BMD. Some functionalities can be given to more I/O at the same time, others can be given to only one resource per time.

| Functionalities | Given to... |
|---|---|
| Generic Input (I/O X - In X) | I/O 0, I/O 1, I/O 2, I/O 3, In 4, In 5, In 6, In 7, In 8, In 9 |
| Generic Output (I/O X - Out X) | I/O 0, I/O 1, I/O 2, I/O 3, Out 4, Out 5, Out6 |
| Fault (Fault) | I/O 0, I/O 1, I/O 2, I/O 3, Out 4, Out 5, Out6 |
| Home | I/O 0, I/O 1, I/O 2, I/O 3, In 4, In 5, In 6, In 7, In 8, In 9 |
| Step | I/O 0 |
| Dir | I/O 1 |
| Positive limit switch (FC +) | I/O 0, I/O 1, I/O 2, I/O 3, In 4, In 5, In 6, In 7, In 8, In 9 |
| Negative limit switch (FC -) | I/O 0, I/O 1, I/O 2, I/O 3, In 4, In 5, In 6, In 7, In 8, In 9 |
| Enable | I/O 3, In 4, In 5, In 6, In 7, In 8, In 9 |
| Quadrature Input ChA (Ch A) | I/O 0 |
| Quadrature Input ChB (Ch B) | I/O 1 |
| Quadrature Input Index (Idx) | I/O 2 |
| Pwm out (Pwm O) | I/O 0, I/O 1, I/O 2 |
| Motor Fan (M. Fan) | I/O 0, I/O 1, I/O 2, I/O 3, Out 4, Out 5, Out6 |

| Functionalities | Given to... |
|---|---|
| Drive Fan (D. Fan) | I/O 0, I/O 1, I/O 2, I/O 3, Out 4, Out 5, Out6 |
| Drive Ok (Drv OK) | I/O 0, I/O 1, I/O 2, I/O 3, Out 4, Out 5, Out6 |
| Simulated 24V Out (S24V) | I/O 3, Out 4, Out 5, Out6 |
| Simulated GND (SGND) | In5 |

***Table 16.6. Functionalities given to the resources of I/O***

NOTE

**Each functionality has a code that can be assigned to the inputs and outputs. These codes are listed in the Table 27.21**

The following table lists the parameters fo the selection of the functionality related to each single I/O. The procedure for their use is described in Section 16.3, *Configuring the I/O by using the parameters*.

| Resource | Parameter |
|---|---|
| I/O 0 | IO_0_Function |
| I/O 1 | IO_1_Function |
| I/O 2 | IO_2_Function |
| I/O 3 | IO_3_Function |
| In 4 | In_4_Function |
| In 5 | In_5_Function |
| In 6 | In_6_Function |
| In 7 | In_7_Function |
| In 8 | In_8_Function |
| In 9 | In_9_Function |
| Out 4 | Out_4_Function |
| Out 5 | Out_5_Function |
| Out 6 | Out_6_Function |

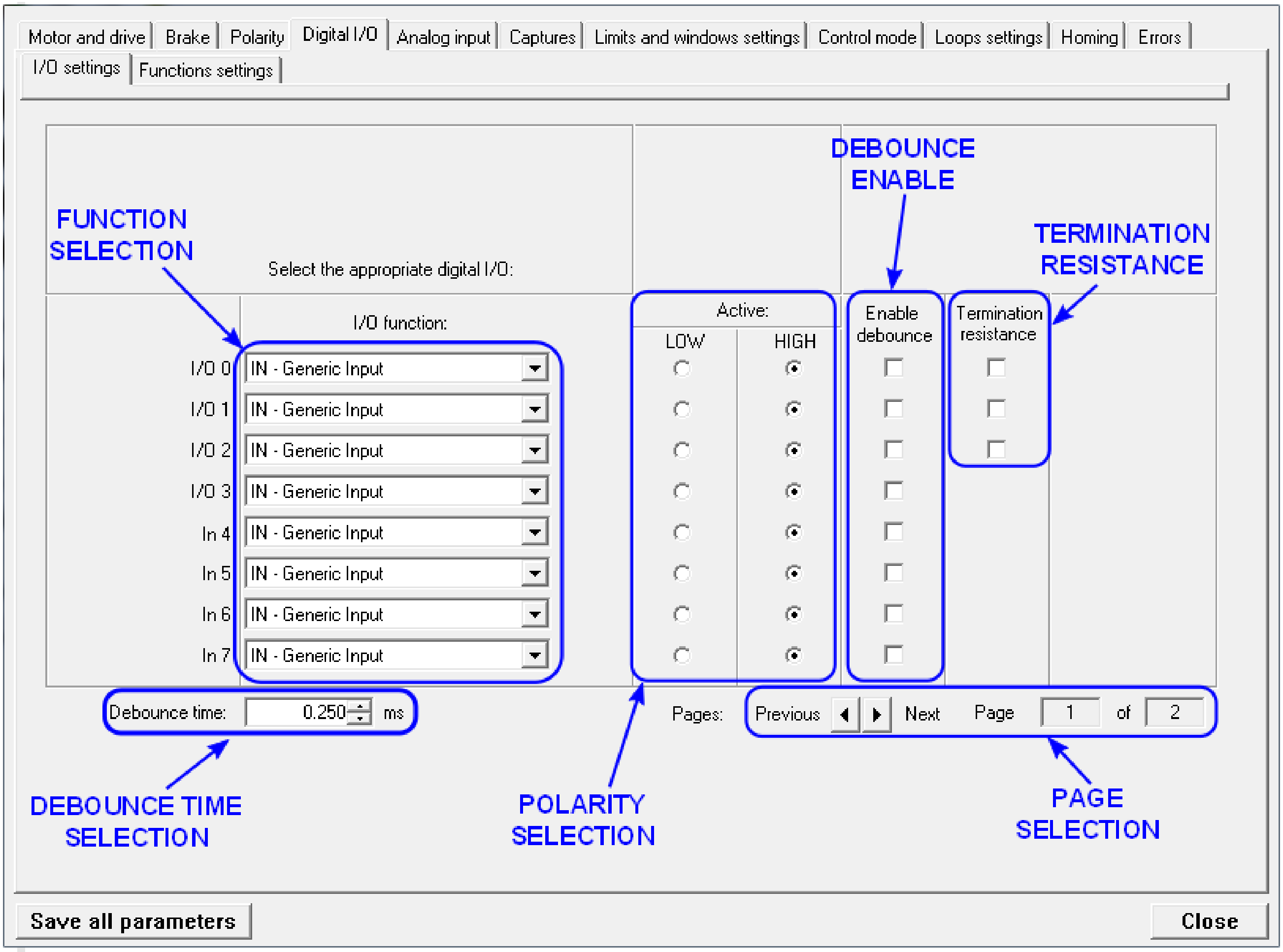
***Table 16.7. List of the configuration parameters of the functionalities supported by every resource in the iBMD drives***

## 16.2. Configuring the I/O through MotionDrive

All the settings about I/O are available from MotionDrive. Access:

**Main menu** > **Drive** > **Drive setup ...** > **Digital I/O** > **I/O settings**

**Toolbar** >  > **Digital I/O** > **I/O settings**

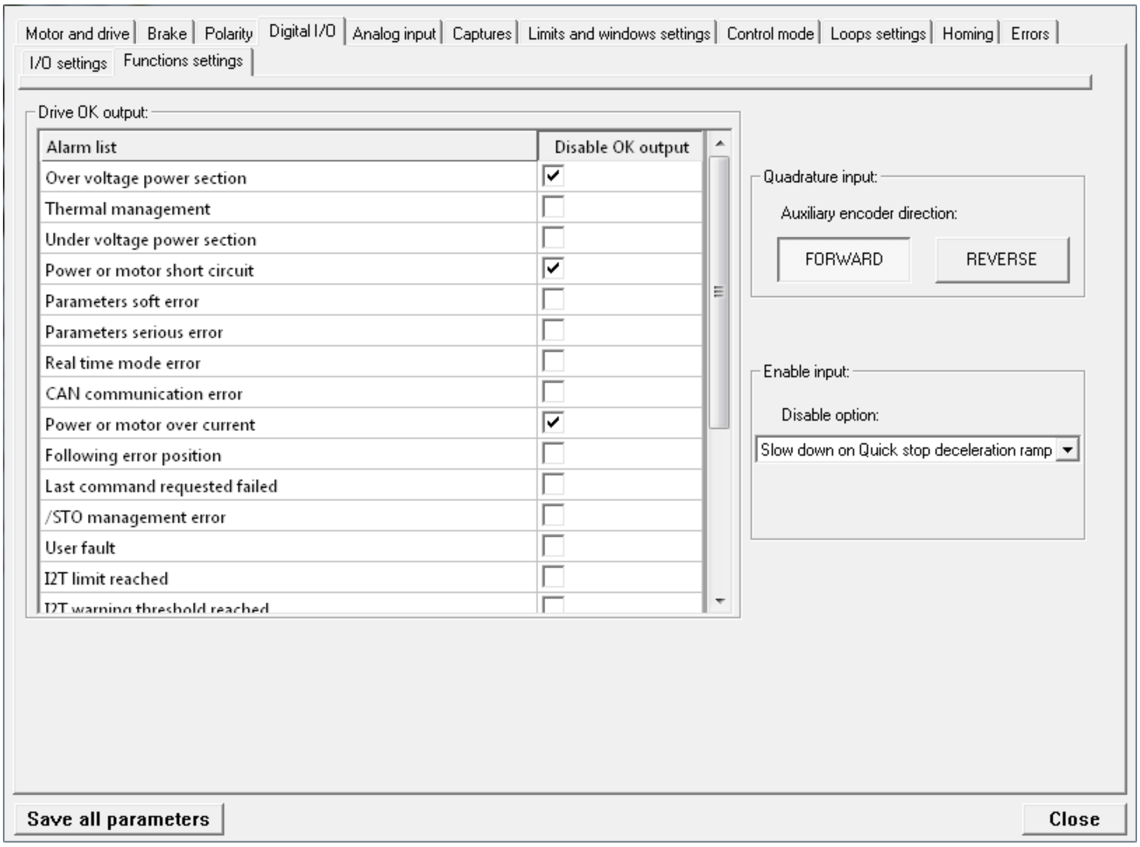



The mask of the Faults related to the functioning of the Drive Ok (Drv OK) output, the selection of the working mode of the Enable functionality and the positive increment of the counting of the master encoder can be selected through MotionDrive. Access:

**Main menu** > **Drive** > **Drive setup ...** > **Digital I/O** > **Function settings**

**Toolbar** > > **Digital I/O** >**Function settings**

| IMPORTANT |
|---|
| **If one of the three differential I/O (I/O 0, I/O 1 e I/O 2) is reprogrammed, even the other two will be reconfigured. This may cause, even if the set functionality will not be lost, some "glitches", that are involuntary short transitions of the digital signal.** |




In this page it is possible to select the Faults that causes the switching-off of the output programmed as Drive Ok (Drv OK).

It is furthermore possible to set if the disabling, commanded by the digital input configured as Enable, must be anticipated by the motor stopping with the QuickStop ramp (see Section 22.3, *Carrying out a stop by using the master*).

## 16.3. Configuring the I/O by using the parameters

To configure the Digital I/O by writing directly the related parameters, follow these instructions:

1. Run the command of the System Manager 6200 to start the configuration procedure.
2. Select the functionalities through the parameters listed in Table 16.7; the codes of the functionalities are listed in the description of the parameters.
3. Configuring the polarity (PolarityInputValue).
4. Run the command of the System Manager 620 to end the configuration procedure.
5. Check any possible error.

The running of the settings related to the filtering and the termination resistance do not require any particular modes or commands of the System Manager. Refer to Section 16.5, *Filters, polarities and terminations*.

## 16.4. Functionalities

**Generic Input (I/O X - In X)**
The resource works as input for general use. The physical status of the input can be read through the parameter DigitalInputs. The parameter LogicalDigitalInputStatus shows the input status after the application of the polarity (see Selection of the polarity of the digital inputs).

**Generic Output (I/O X - Out X)**
The resource works as output for general use. The output status can be read and written through the parameter PhysicalOutputs. To prevent the accidental modification of one or more bits of the parameter PhysicalOutputs it is possible to block the writing, every bit through the parameter DigitalOutputsBitMask.

**Fault (Fault)**
The *Fault* functionality enables the output when a retentive Fault is active. When the Fault is reset, the status of the digital output is set to zero. See Chapter 24, *Fault and Warning*.

**Home**
Input used to carry out the homing of the shaft. (See Section 22.19, *Homing Mode*). Through the parameter HomeStatus it is possible to read the status of the limit switch, irrespective of the resource used for such functionality.

**Step**[3]
The functionality *Step* is used to get information on position and speed, gaining them from the frequency of the applied signal. This functionality can be used with the functionality Dir. In this way it is possible to link the drive to third part controllers.

**Dir**[4]
The functionality *Dir* is used to get the direction of the reference gained through the functionality Step. This functionality can be used with the functionality Step. In this way it is possible to link the drive to third part controllers.

**Positive limit switch (FC +)**
Input of positive limit switch. The sensors used as limit switches must work when the contact is normally closed. The drive, for safety reasons, automatically selects the polarity of the input configured as limit switch, so that it can correctly work with this kind of sensor (see Selection of the polarity of the digital inputs). Through the parameter FcStatus it is

[3]This functionality is not yet available
[4]This functionality is not yet available

possible to read the status of the limit switch, irrespective of the resource used for such functionality.

WARNING

**If the Polarity is of Reverse type, the roles of Positive limit switch (FC +) and Negative limit switch (FC -) are reversed: Positive limit switch (FC +) behaves like Negative limit switch (FC -) and Negative limit switch (FC -) behaves like Positive limit switch (FC +). This is true both in the text of this manual and in the MotionDrive.**

### Negative limit switch (FC -)

Input of negative limit switch. The sensors used as limit switches must work when the contact is normally closed. The drive, for safety reasons, automatically selects the polarity of the input configured as limit switch, so that it can correctly work with this kind of sensor (see Selection of the polarity of the digital inputs). Through the parameter FcStatus it is possible to read the status of the limit switch, irrespective of the resource used for such functionality.

### Enable

The functionality Enable is a consent to enable the drive. When the functionality Enable is associated to one of the digital inputs, such input must be at the logical status '1' in order to take the drive to the Operation enable state. The consent to enable the drive can be read through the parameter EnableInputStatus irrespective of which input has been used for the Enable function. When EnableInputStatus value is 0, the CiA402 State Machine cannot be taken to the *Switched On* and Operation enable states (see Section 8.5, *CiA402 state machine*).

If the drive is in the Operation enable state and the Enable input is disabled then the following sequence takes place:

1. deceleration of the motor according to the settings of the parameter DisableOption;
2. waiting for the motor stopping and enabling of the brake, if present;
3. the CiA402 State Machine enters the *Switch On Disabled* state (see Section 8.5, *CiA402 state machine*);
4. activation of the Drive is in disable state, since the enable input is or has been in not active state Warning.

IMPORTANT

**For some operating modes, when the Enable input is enabled, the drive goes automatically to the Operation enable state, unless the drive is in Fault. This function is called Automatic Enable; the operating modes with automatic Enable are specified in Table 22.1.**

### Quadrature Input ChA (Ch A)

The functionality *Quadrature Input ChA* is used, with Quadrature Input ChB (Ch B), to get a quadrature signal, typically used by the incremental encoders.

To reverse the sense of positive rotation of the master encoder without modifying the electrical connections you can work on the RealEncoderPolarity parameter.

If you select this functionality the parameter PolarityInputValue does not have any effect.

> ⚠ NOTICE
>
> **To read the master encoder position or to capture its position, it's necessary to program this functionality (together with Quadrature Input ChB (Ch B)) in the digital inputs. See Section 15.5, *Auxiliary position sensor***

### Quadrature Input ChB (Ch B)

Through the functionality *Quadrature Input ChB* it is possible, together with Quadrature Input ChA (Ch A), to get a quadrature signal, typically used on the incremental encoders.

To reverse the sense of positive rotation of the master encoder without modifying the electrical connections you can work on the RealEncoderPolarity parameter.

If you select this functionality the parameter PolarityInputValue does not have any effect.

> ⚠ NOTICE
>
> **To read the master encoder position or to capture its position, it's necessary to program this functionality (together with Quadrature Input ChA (Ch A)) in the digital inputs. See Section 15.5, *Auxiliary position sensor***

### Quadrature Input Index (Idx)

The *Quadrature Input Index* functionality is used to get the Index pulse of an incremental encoder. It must be used together with Quadrature Input ChA (Ch A) and Quadrature Input ChB (Ch B).

To reverse the sense of positive rotation of the master encoder without modifying the electrical connections you can work on the RealEncoderPolarity parameter.

If you select this functionality the parameter PolarityInputValue does not have any effect.

> ⚠ NOTICE
>
> **To use the Index as the capture event to capture the master encoder position it's necessary to program this functionality in the digital input 2 (see IO_2_Function).**

### Pwm out (Pwm O)

The functionality *Pwm out* runs the output by creating a frequency square wave and duty cycles to be set. The configuration parameters depend on which resource is used to run this functionality; you can find a list on the following chart:

| Resource | Parameter for frequency setting | Parameter for Duty Cycle setting |
|---|---|---|
| I/O 0 | PwmHwFrequencyIO0 | PwmHwDutyCycleIO0 |
| I/O 1 | PwmHwFrequencyIO1 | PwmHwDutyCycleIO1 |
| I/O 2 | PwmHwFrequencyIO2 | PwmHwDutyCycleIO2 |

### Motor Fan (M. Fan)

The functionality *Motor Fan* is used to run a fan to cool the motor. The output is automatically enabled when the motor temperature exceeds the warning threshold (see Table 24.3). When the temperature of the motor is lower than this threshold, the output remains enabled for one minute and then it switches off.

### Drive Fan (D. Fan)

The functionality *Drive Fan* is used to run a fan to cool the drive. The output is automatically enabled when the temperature of the control section or the one of the power section exceeds the Warning threshold (see Table 24.3). When both temperatures are lower than this threshold, the output remains enabled for one minute and then it switches off.

### Drive Ok (Drv OK)

The functionality *Drive Ok* enables the output when the drive has finished the start-up phase and it is ready to get any command. This output is switched off because of the presence of Faults, since the drive is no longer operative. The Faults switching this output off can be selected through the DisableOkOutput parameter.

### Simulated 24V Out (S24V)

The *Simulated 24V Out* functionality transforms the selected resource in a 24V supply output. If used together with Simulated GND (SGND), it can be used to supply external devices.

IMPORTANT

**This functionality is available on drives with hardware revision 17 or higher.**

### Simulated GND (SGND)

The functionality *Simulated GND* transforms the selected resource into a ground terminal. It is useful to close the circuit of the digital outputs. If used together with Simulated 24V Out (S24V), it can be used to supply external devices.

⚠ NOTICE

**Simulated GND (SGND) input is not protected against overcurrent.**

IMPORTANT

**This functionality is available on drives with hardware revision 17 or higher.**

## 16.5. Filters, polarities and terminations

**Filtering of the digital inputs**
The status of the digital inputs is updated every 250 µs.

If it is necessary to filter the digital inputs, it is possible to set a time interval, every 250 µs, during which the input status must remain stable to be validated (debounce time).

For example if, due to noises, on the inputs some unwanted status changing happens for a shorter time than the debounce time, these noises are filtered and the input status is unaltered.

After this debounce time the image stored in the digital inputs is updated with the new status. The highest filter value is 65 ms.

The debounce configuration takes place through two parameters:

- DebounceTime: time during which the input status must be stable to be validated. It can be set every 250 µs.
- EnableDebounce: Mask used to select on which digital inputs you can apply the filtering.

The possibility to enable the debounce depends on the functionality given to the digital input; the functionalities used to enable the filtering are:

- Generic Input (I/O X - In X)
- Positive limit switch (FC +)
- Negative limit switch (FC -)
- Home.

**Selection of the polarity of the digital inputs**
For the inputs of PNP type the enabled physical status is reached when the applied voltage (referred to the ground signal) surpasses the activation threshold declared on the Table 16.3. For the line-driver differential ones the enabled status is reached when the voltage difference between the positive and negative inputs is greater than $+V_{thd}$, and the not enabled status is reached when the voltage difference is lower than $-V_{thd}$ (see Table 16.3).

the inputs logical status (1 or 0 in the inputs image) depends on both their physical status and the polarity that's set through the PolarityInputValue parameter. If the polarity value is 0 then the logical status coincides with the physical one, otherwise if the polarity value is 1 then the logical status is inverted respect to the physical one. The PolarityInputValue parameter must be written during the Digital I/O configuration procedure (see Section 16.3, *Configuring the I/O by using the parameters*).

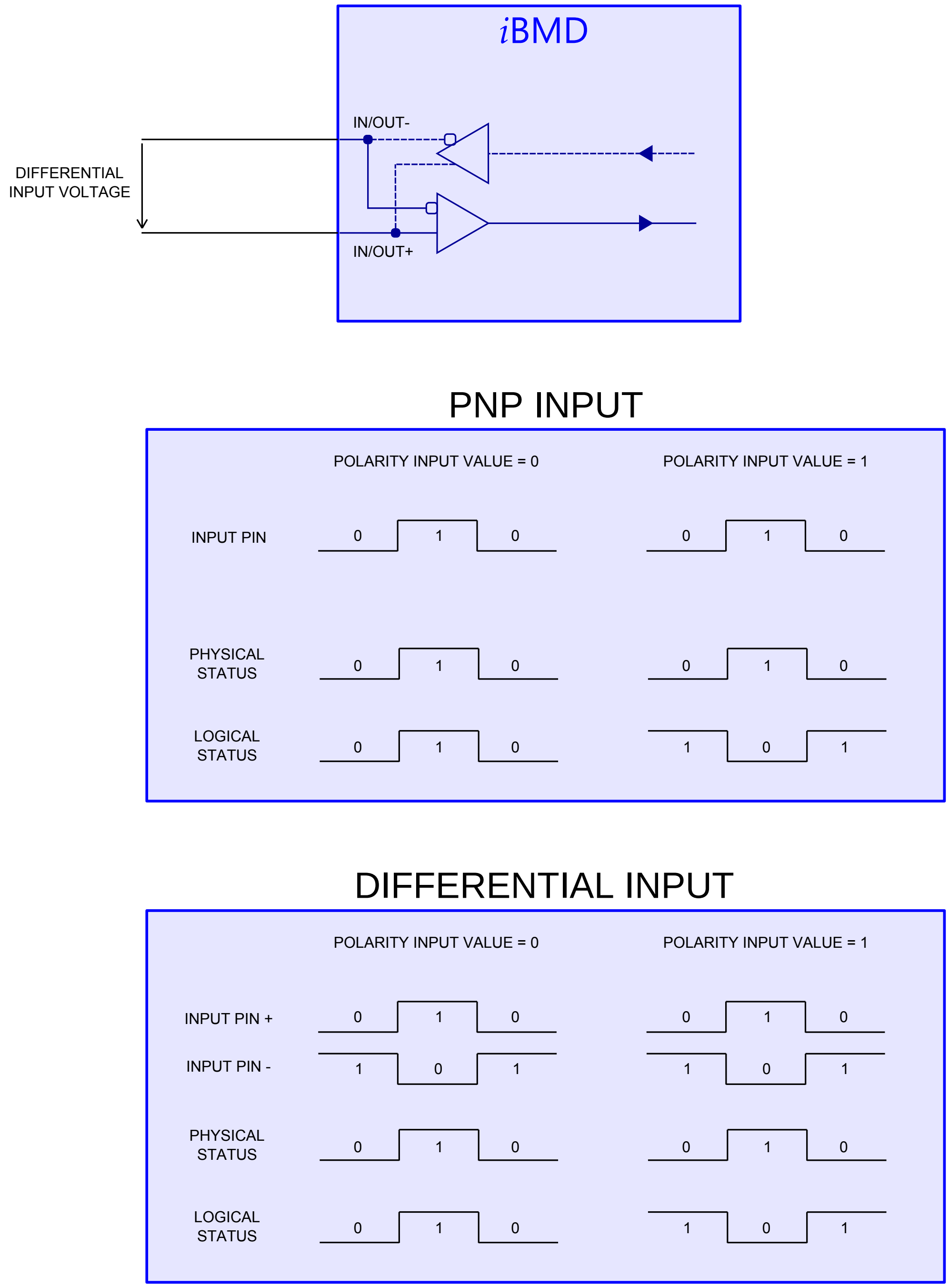



*Figure 16.1. Inputs logical status in relation to the polarity.*

The functionalities Positive limit switch (FC +) and Negative limit switch (FC -) force to zero the bits corresponding to the parameter PolarityInputValue.

**Termination resistances**
For the resources with logic of differential kind (Table 16.1) it is possible to enable the termination resistance through the parameter TerminationResistance.

# Chapter 17 Analog input

The features of the analog input are summarized in the following table:

| Analog input 0 | Details |
|---|---|
| Electric features | Range ±10V; Precision ±10mV |
| Connections | Pin 3 and 4 of CN4 |
| Updating time AI0Voltage | CurrentLoopPeriod |
| Updating time AI0FilteredVoltage | MotionLoopPeriod |

***Table 17.1. Features of AI0***

The features of the analog input, related to the CN4 connector, are summarized in the following table:

| ANALOG INPUT FEATURES | |
|---|---|
| Maximum operating differential voltage | ±10V |
| Maximum absolute differential voltage | ±15V |
| Maximum common-mode voltage[a] | • with Vin=+10V → -18,9V < Vcm < +7,7V<br>• with Vin=-10V → -2,3V < Vcm < +27,7V |
| Input detection delay | max 300µs |
| Resolution | ±50mV |
| Differential input resistance | > 150kΩ |

[a]relative to the system power ground.

***Table 17.2. Analog input electrical features on CN4***

The parameters of the analog input are summarized in the following table:

| Analog input 0 | Parameter | Description |
|---|---|---|
| Capture | AI0Voltage | Not filtered value |
| | AI0FilteredVoltage | Filtered value |
| Calibration | AI0CalibrationStatus | Calibration status |
| | AI0CalibrationOffset | Calibration offset |
| | AI0CalibrationGain | Calibration gain |
| | AI0CalibrationVoltage | Calibration voltage |
| Filter | AI0FilterFrequency | Filter frequency |
| | AI0FilterType | Filter type |
| | AI0FilterQFactor | Filter Q factor |
| Conversion | AI0VSettings | Set-up of the voltage for the conversion |
| | AI0RSettings | Set-up of the conversion reference |
| | AI0VPolarity | Polarity of the voltage for conversion |
| | AI0RPolarity | Polarity of the conversion reference |
| | AI0V0Zone | Half amplitude of the dead zone in the conversion |
| | AI0VRefLevel | Voltage value to define the conversion |
| | AI0TRefValue | Torque value to define the conversion |
| | AI0WRefValue | Speed value to define the conversion |

NOTE

**About the electrical connections please see the Analog input data in the related section.**

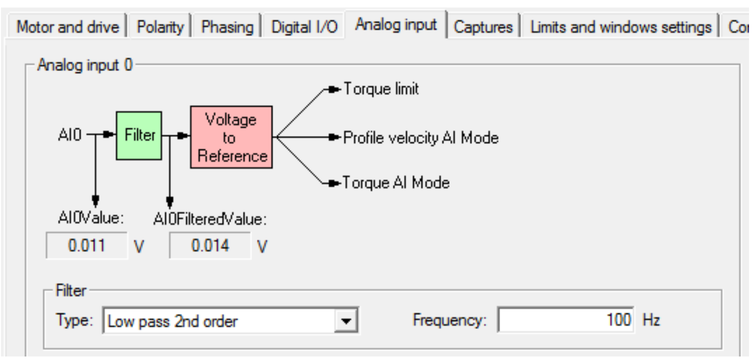
# 17.1. Capture

The analog input is sampled every CurrentLoopPeriod and can be read in the parameter AI0Voltage. The filtered value of AI0Voltage is updated every MotionLoopPeriod and can be read in the parameter AI0FilteredVoltage.

Access with MotionDrive:

**Main menu** > **Drive** > **Drive setup ...** > **Analog input**

**Toolbar** >  > **Analog input**



## 17.2. Calibration

The analog input calibration is made on every single drive by Bonfiglioli.

ADVICE

**Carry out the analog input calibration only after having precisely checked that the drive does not have a correct voltage value. It is possible to carry out the calibration to adapt the values of the analog input to the voltage generated by a generic source.**

To carry out the calibration of the analog input follow these instructions:

- Step 1: analysis
  - switch off all circuits that can influence the reading accuracy of the analog input;
  - apply a direct constant voltage to the analog input;
  - make use of a voltmeter previously calibrated and enough precise;
  - check the reading accuracy of the analog input by making reference to the electric features in Table 17.1. If accuracy is observed, it is not necessary to carry out the calibration; if accuracy is not observed and you think you shall carry out the calibration, go to step 2;
- Step 2: Offset calibration
  - apply a voltage 0V to the analog input (or firmly short-circuit the analog input);
  - run the command of System Manager 7200;
  - check if the value of *AI0CalibrationOffset* is between -10 e +10; if the value of AI0CalibrationOffset is in the specified interval then go to step 3, otherwise repeat more precisely the step 2 or contact Bonfiglioli;
- Step 3: Gain calibration
  - apply a direct voltage between +4 e +10V to the analog input;
  - measure the applied voltage through a voltmeter previously calibrated and enough precise and write its value in the parameter *AI0CalibrationVoltage* ;
  - run the command of System Manager 7201;
  - check if the value of *AI0CalibrationGain* is between 4950 e 5050; if the value of AI0CalibrationGain is in the specified interval then go to step 4, otherwise repeat more precisely the step 3 or contact Bonfiglioli;
- Step 4: checking
  - apply different voltage values to the analog input and check if the voltmeter and the parameter AI0Voltage give the same results according to the accuracy specified in the Table 17.1; if all the comparisons give a positive result, go to step 5, otherwise repeat the calibration from the beginning or contact Bonfiglioli;
- Step 5: data storage
  - the calibration parameters of the analog input are of ES-type and they can be saved in the permanent memory by running the command of System Manager 2001.

  IMPORTANT

  **By restoring the default values of the parameters, the calibration data of the analog input are overwritten with their related default values.**

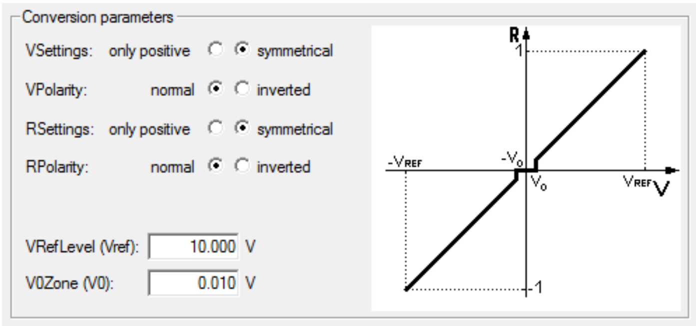
## 17.3. Conversion

The value of the filtered analog input (AI0FilteredVoltage) can be used as torque limit or torque or speed reference depending on the value of TorqueLimitSelector and of ModesOfOperation. To convert the voltage values in torque or speed values, the parameters AI0ConversionParameters are used. In order to define the various conversion options, use MotionDrive, considering the reported cenversion diagrams.

**Parameters to define the conversion function**
In the following MotionDrive page, the parameters used to convert the input voltage can be set. Access:

**Main menu** > **Drive** > **Drive setup ...** > **Analog input**

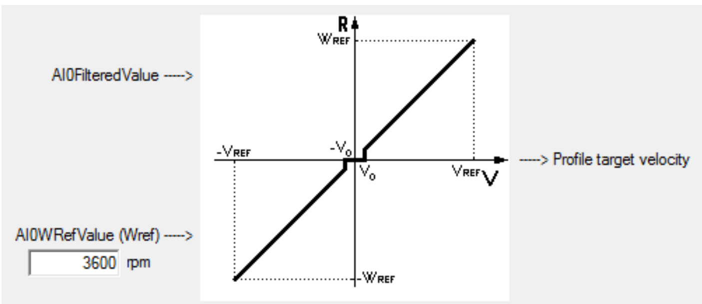
**Toolbar** >  > **Analog input**

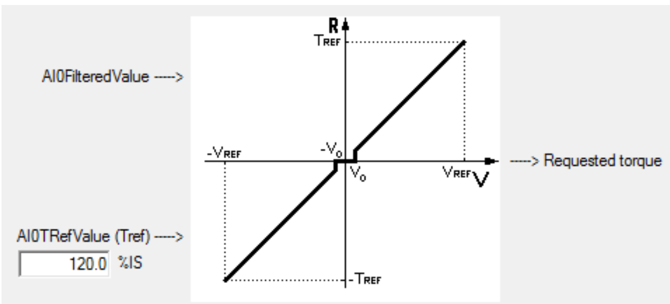



**References related parameters**
In the following page of MotionDrive it's possible to choose the parameters related to the references (the AI0TRefValue parameter is even used by the torque limit). Access:

**Main menu** > **Drive** > **Drive setup ...** > **Control mode**

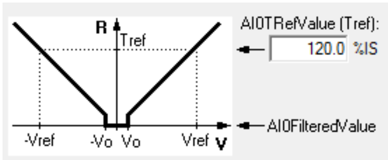
**Toolbar** >  > **Control mode**



**Torque limit related parameters**
In the following page of MotionDrive it's possible to choose the parameters related to the torque limit (the AI0TRefValue parameter is even used by the reference conversion). Access:

> **Main menu** > **Drive** > **Drive setup ...** > **Limits and windows settings**
>
> **Toolbar** >  > **Limits and windows settings**



# Chapter 18
# Capture Peripherals

The drives of *i*BMD series are equipped with two capture peripherals which allow to capture a maximum of 3 quantities each and which are driven by a trigger signal that causes the capture. This signal can be linked to a digital input or can be controlled by the user, through a parameter, to force the capture via software.

Furthermore the drive provides some particular filters and algorithms to detect and validate the capture events. In this way, through the capture peripherals, it is possible to carry out some complex functionalities or avoid undesired captures.

The currently available algorithms are:

- Filter on CaptureSource0_A and CaptureSource0_B (typically the position).

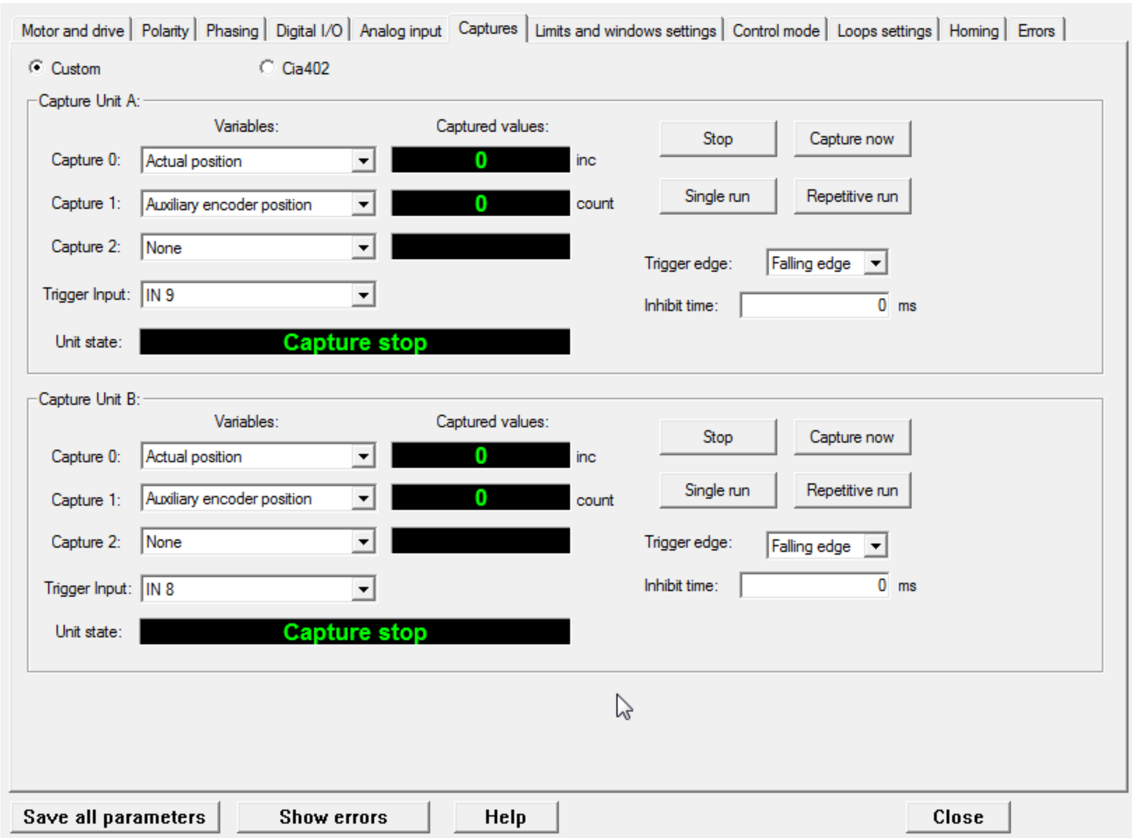
From now onwards we are going to call the two capture peripherals **Capture A** e **Capture B**.

> ⚠ NOTICE
>
> **If the desired capture is the one with the master encoder Index, remember to program the digital input 2 (see IO_2_Function) with the Quadrature Input Index (Idx) functionality.**

## 18.1. Configuring the capture by using MotionDrive

In the MotionDrive it's possible to make the capture configuration through the related page in the DriveSetup. Access:

> **Main menu** > **Drive** > **Drive setup ...** > **Captures**
>
> **Toolbar** > > **Captures**




## 18.2. Configuration interface selection

In the *i*BMD series drives, 2 configuration interfaces have been integrated: CUSTOM interface and CiA-402 interface, that can be selected through the CaptureInterfaceMode parameter.

> IMPORTANT
>
> **It's not possible to change the interface if the capture functionality is active:**
> **- in case CaptureInterfaceMode value is 0 (CUSTOM interface): if at least one of the CaptureUnitState_A and CaptureUnitState_B parameters value is 1**
> **- in case CaptureInterfaceMode value is 1 (CiA-402 interface): if both the bits 0 and/or both the bits 8 of TouchProbeFunction and of TouchProbeStatus are set to 1.**

> ⚠ NOTICE
>
> **These two interfaces cannot be contemporary used because the contained informations are not consistent.**

In fact these two capture interfaces differ in some details. Here follows the list of the main limits and differences:

- If case of access to the not selected interface, in reading operation the answer is 0, while in writing operation an ABORT is reported. More than this, the "Capture setup setting by using disabled parameters interface (look at parameter 'CaptureInterfaceMode')" warning bit is set, in the ParamSoftError parameter. This Warning doesn't automatically reset, it has to be reset by the user.
- When the interface is changed, the new one is re-initialized with the values that are memorized in the EEprom, if they are compatibles with the selected interface, or with the default values.
- The filter in space (CaptureSource0_A and CaptureSource0_B) is only available for the CUSTOM interface and cannot be used with the double side capture.
- In both the interfaces, the settings of the repetitive mode and of the trigger cannot be made with the capture enabled. If these selections are made, then the "Filter or trigger on both edges not allowed on selected trigger input" Warning bit is set in the ParamSoftError parameter. In the CUSTOM mode the enable operation is not allowed, in the CiA-402 mode, considering that the setting and the enable operations are contemporary (because both are done through the same parameter), the parameter reports an Abort.

## 18.3. Configure the capture by using the CUSTOM interface parameters

The configuration of the capture peripheral must be carried out when the peripheral is disabled, otherwise the configuration parameters will not be writable.

The two capture peripherals available on the drives of *i*BMD series are identical. Here follow the configuration parameters for each peripheral:

| Configuration | Capture A | Capture B |
|---|---|---|
| Trigger signal | CaptureTriggerInput_A | CaptureTriggerInput_B |
| First quantity to capture | CaptureSource0_A | CaptureSource0_B |
| Second quantity to capture | CaptureSource1_A | CaptureSource1_B |
| Third quantity to capture | CaptureSource2_A | CaptureSource2_B |
| Capture edge | CaptureTriggerEdge_A | CaptureTriggerEdge_B |
| Inhibit time | CaptureInhibitTime_A | CaptureInhibitTime_B |
| State of the capture peripheral | CaptureUnitState_A | CaptureUnitState_B |
| Capture peripheral control | CaptureUnitCommand_A | CaptureUnitCommand_B |
| Captured value, first quantity | CapturedValue0_A | CapturedValue0_B |
| Captured value, second quantity | CapturedValue1_A | CapturedValue1_B |
| Captured value, third quantity | CapturedValue2_A | CapturedValue2_B |

Once trigger (CaptureTriggerInput_A), values to capture (CaptureSource0_A, CaptureSource1_A, CaptureSource2_A), capture edge (CaptureTriggerEdge_A) and inhibit time (CaptureInhibitTime_A) are configured, you can start the capture peripheral by properly writing the CaptureUnitCommand_A parameter.

Now, the capture state has to be verified through the CaptureUnitState_A parameter, and when it indicates that the capture has happened, the results can be read through the CapturedValue0_A, CapturedValue1_A and CapturedValue2_A parameters.

To optimize the space, if the capture results are mapped on PDO, you can use some parameters having a different length, that you can use depending on the needs. In the following chart you can find the table:

| Configuration | Capture A (word) | Capture A (byte) | Capture B (word) | Capture B (byte) |
|---|---|---|---|---|
| Captured value, first quantity | CapturedValue0_Word_A | CapturedValue0_Byte_A | CapturedValue0_Word_B | CapturedValue0_Byte_B |
| Captured value, second quantity | CapturedValue1_Word_A | CapturedValue1_Byte_A | CapturedValue1_Word_B | CapturedValue1_Byte_B |
| Captured value, third quantity | CapturedValue2_Word_A | CapturedValue2_Byte_A | CapturedValue2_Word_B | CapturedValue2_Byte_B |

## 18.4. Filter on CaptureSource0_A and CaptureSource0_B

This kind of filter validates the capture if the trigger signal is enabled, until the value of the first quantity to capture evolves into a quantity set by the user. The capture is carried out on the edge of the trigger signal which has been programmed (CaptureTriggerInput_A, CaptureTriggerInput_B), but the capture peripheral reports that the capture has been carried out and shows the results only after the validation.

At this point the trigger signal is on the active state. To go back to the "inactive" state and to carry out a new capture, the trigger signal undergoes the same validation. The filtering values for the enabled capture edge and for the restore edge can be different.

This kind of filter is normally used by selecting PositionActualValue on the first quantity to capture; in this way it is possible to apply a proportional filtering to the motor speed, in this case the capture is validated only if the trigger signal is enabled for a certain position range. The functioning of the filter is shown in Figure 18.1.

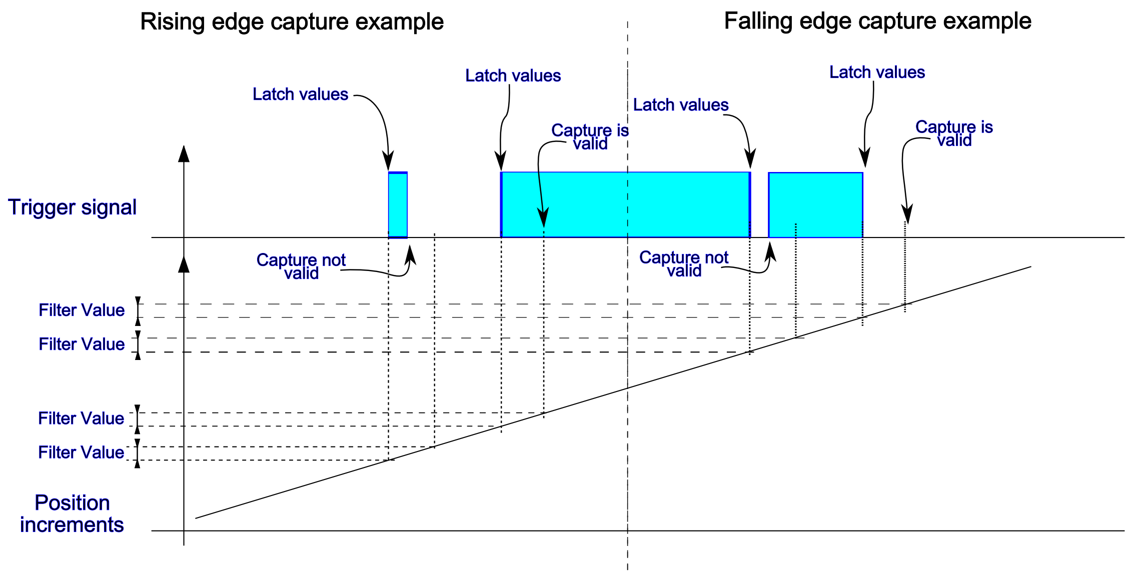



*Figure 18.1. Example of filtering of the trigger signal.*

**Configuration of the filtering**

The configuration of this function is very easy, since it is just necessary to specify the value of the filtering (parameters CaptureRestoreSlopeValidationFilter_A and CaptureActiveSlopeValidationFilter_A for Capture A, and parameters CaptureRestoreSlopeValidationFilter_B and CaptureActiveSlopeValidationFilter_B for Capture B), and the mode of the symmetric/asymmetric filtering (parameters CaptureValidationFilterMode_A for Capture A, and CaptureValidationFilterMode_B for Capture B).

The value according to which the parameters are set is related to the first quantity to capture, so it has the same unit of measurement.

Chapter 19

# Motion limits

## 19.1. Electricity limit

To reduce the motor electricity you need to write the parameter UserPeakCurrent. Reducing the motor electricity means reducing the motion performances, so reducing the highest supplied torque (ActualTorqueLimitP).

Access with MotionDrive:

> **Main menu** > **Drive** > **Drive setup ...** > **Motor and drive**
>
> **Toolbar** >  > **Motor and drive**


## 19.2. Limit I2T

The I2T limit reduces the electric power which is transferred to the motor during the overload periods. For further details see Section 14.6, *I2T*.

# 19.3. Torque limit

To reduce the highest supplied torque write the parameter TorqueLimitSelector so that you can select the source where the torque limit can be obtained. The torque limit can be limited by:

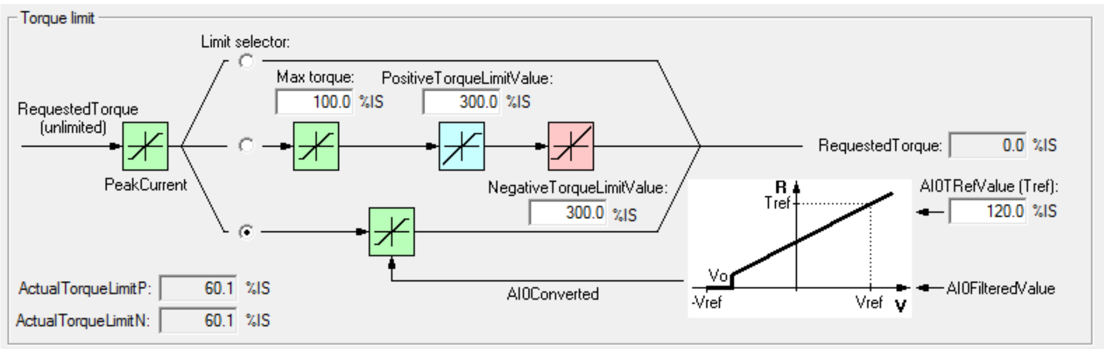
- the PeakCurrent parameter value only
- the combination between the MaxTorque, PositiveTorqueLimitValue and Negative-TorqueLimitValue parameters
- the value that has been obtained from the analog input voltage conversion, according to what's reported in Section 17.3, *Conversion*

In any case, in the ActualTorqueLimitP and ActualTorqueLimitN parameters you can read the value of the torque limits that are really applied to the motor. When a torque limit intervenes, the Limit reached Warning activates with the Torque limit reached detail.

Access with MotionDrive:

**Main menu** > **Drive** > **Drive setup ...** > **Limit and windows settings**

**Toolbar** > > **Limit and windows settings**




**Example of torque limit use**

In order to execute a homing with mechanical stop, it is necessary to activate the torque limit. For example, by setting the mode -1 in the HomingMethod parameter, the mode 2 in the TorqueLimitSelector parameter, it has to be set the conversion of the limit from the analog input, as described in the Section 17.3, *Conversion*.

**Torque values calculation procedure**

In order to limit the torque to the value $T_{Lim}$, continue as follows:

1. read the torque constant $K_T$ TorqueConstant(ForceConstant)
2. read the stall current $I_S$ MotorStallCurrent
3. calculate the stall torque $T_S$ [Nm] as $K_T$ [Nm/A] * $I_S$ [A]
4. calculate the $T_{Lim}$ torque limit expressed in [%$I_S$] as ($T_{Lim}$[Nm] / $T_S$) * 100 or as ($I_{Lim}$[A] / $I_S$) * 100
5. multiply the obtained value per 10 to obtain the $T_{Lim}$ torque limit expressed in [%$I_S$10] and insert this value in the desired torque parameter

**Calculation example of PositiveTorqueLimitValue**
With a *i*BMD with a 2.8 Nm motor, we want to limit the torque, in positive direction, to 1Nm and to not limit the torque in the negative direction. by following the above described procedure we obtain:

$K_T$ = 1.6 Nm/A
$I_S$ = 1.75 A
PeakCurrent = 5 A
$T_S = K_T$ [Nm/A] * $I_S$ [A] = 1.6 Nm/A * 1.75 A = 2.8 Nm
$T_{LimP}$ [%$I_S$] = ($T_{Lim}$ [Nm] / $T_S$ [Nm]) * 100 = (1 Nm / 2.8 Nm) * 100 = 35.7 %$I_S$
$T_{LimN}$ [%$I_S$] = ($I_{Lim}$ [A] / $I_S$ [A]) * 100 = ( 5 A / 1.75 A) * 100 = 285.7 %$I_S$
PositiveTorqueLimitValue = $T_{LimP}$ [%$I_S$] * 10 = 35.7 %$I_S$ * 10 = 357 %$I_S$10
MaxTorque = NegativeTorqueLimitValue = $T_{LimN}$ [%$I_S$] * 10 = 285.7 %$I_S$ * 10 = 2857 %$I_S$10

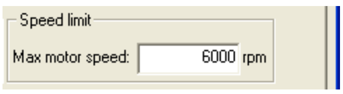
## 19.4. Speed limit

To set the speed limit you need to write the parameter MaxMotorSpeed. The speed limit is an absolute value and it works in a symmetric way on the speed request of the drive. It works only with the operative modes speed or position. When a speed limit intervenes, the Warning Limit reached activates with the Max motor speed limit reached detail.

Access with MotionDrive:

**Main menu** > **Drive** > **Drive setup ...** > **Limit and windows settings**

**Toolbar** >  > **Limit and windows settings**




## 19.5. Limits of hardware position

To enable the limits of hardware position you need to set the functionalities Positive limit switch (FC +) and Negative limit switch (FC -) on two digital inputs of the drive (see Chapter 16, *Digital inputs and outputs*).

> ⚠ WARNING
>
> **If the Polarity is of Reverse type, the roles of Positive limit switch (FC +) and Negative limit switch (FC -) are reversed: Positive limit switch (FC +) behaves like Negative limit switch (FC -) and Negative limit switch (FC -) behaves like Positive limit switch (FC +). This is true both in the text of this manual and in the MotionDrive.**

When the drive is in Operation enable, RequestedSpeed (TargetTorque for torque modes) is higher than 0 and the Positive limit switch (FC +) input enables, or RequestedSpeed (TargetTorque for torque modes) is lower than 0 and the Negative limit switch (FC -) input is enabled, the motor stops with a deceleration ramp equal to QuickStopDeceleration.

During the deceleration ramp, the CiA402 State Machine goes to the *Quick Stop Active* state and, once the motor has stopped, it goes back to the Operation enable state (see Section

8.5, *CiA402 state machine*). When one of the hardware position limits is enabled, the Limit reached Warning activates with the related detail (Positive hardware position limit reached or Negative hardware position limit reached), and remains active until the limitation stops.

**ATTENTION**

**When the drive is in Operation enable, RequestedSpeed (TargetTorque for torque modes) is higher than 0 and the Negative limit switch (FC -) input enables, or RequestedSpeed (TargetTorque for torque modes) is lower than 0 and the Positive limit switch (FC +) input enables, the motor does not stop and there are no further signals.**

## 19.6. Limits of software position

To enable the limits of software position you need to write 1 in the parameter PositionLimitEnable and write the two limits, the positive and the negative one, in the group SoftwarePositionLimit.

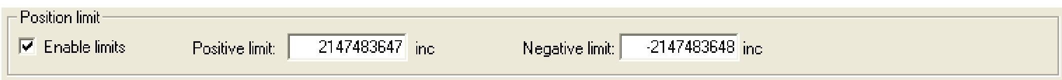
When the drive is in Operation enable, RequestedSpeed (TargetTorque for torque modes) is higher than 0 and PositionActualValue is greater than PositionLimitPositive, the motor stops with a deceleration ramp equal to QuickStopDeceleration. When the drive is in Operation enable, RequestedSpeed (TargetTorque for torque modes) is lower than 0 and PositionActualValue is lower than PositionLimitNegative, the motor stops with a deceleration ramp equal to QuickStopDeceleration.

During the deceleration ramp, the CiA402 State Machine goes to the *Quick Stop Active* state and, once the motor has stopped, it goes back to the Operation enable state (see Section 8.5, *CiA402 state machine*). When one of the software position limits is enabled, the Limit reached Warning activates with the related detail (Positive software position limit reached or Negative software position limit reached), and remains active until the limitation stops.

Access with MotionDrive:

**Main menu** > **Drive** > **Drive setup ...** > **Limit and windows settings**

**Toolbar** >  > **Limit and windows settings**




**NOTE**

**If you set PositionLimitPositive lower than PositionLimitNegative the error Parameters soft error enables with detail Software position limits incompatibility.**

## 19.7. Profiles limit

The speed and acceleration parameters of the operating modes, aiming at running a profile to run a motion, are limited by the following parameters:

- MaxProfileVelocity and MaxMotorSpeed: the lowest value between these two parameters, it reduces all speed parameters of the profilers.
- MaxAcceleration:limit of all acceleration parameters of the profilers.

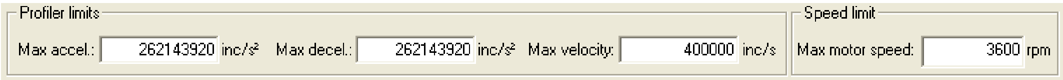
- MaxDeceleration:limit of all the deceleration parameters of the profilers.

When you have a limitation on at least one parameter of the profilers, the Motion parameter limited Warning enables. In Table 24.16 you can find the error detail with the limited parameters.

Access with MotionDrive:

**Main menu** > **Drive** > **Drive setup ...** > **Limit and windows settings**

**Toolbar** >  > **Limit and windows settings**



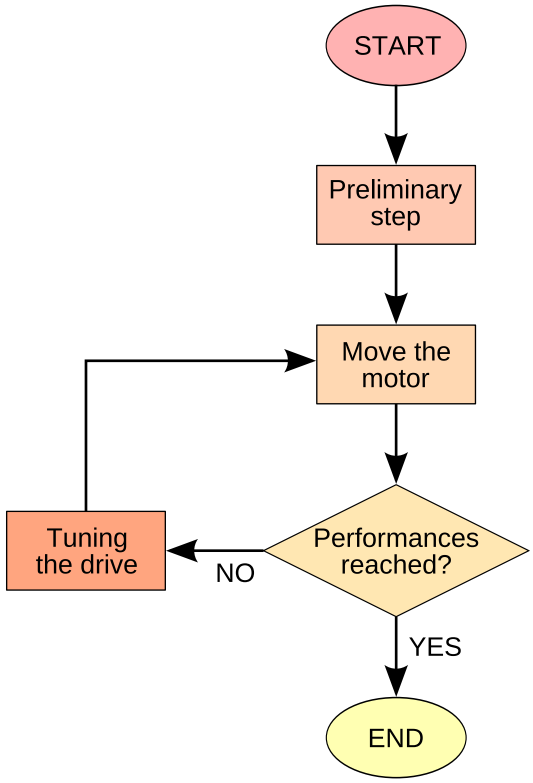

NOTE

**The MaxMotorSpeed sets also the speed limit for all drive motions in the operating modes speed or position (see Section 19.4, *Speed limit*).**

# Chapter 20
# System tuning

## 20.1. How to determine the tuning criterion

To tune the drives of the *i*BMD series follow the instructions in this and in the next paragraphs. The operations for adjusting the drive are summarized in the next flow chart.




**1. Preliminary step A: system data**
Before starting the tuning process it is necessary to gather all the data of the mechanical load; in particular you need to analyse the transmission typology and quality, the stiffness of the machine structure and size of the inertia moments, frictions, elasticity and backlashes. The more information you have, the easier the next tuning phases will be. Check if the requested performances are compatible with the system mechatronic features. Check if the electric connections are correctly connected and the mechanical transmission is perfectly working.

**2. Preliminary step B: requested performances**
Define the motion performances to be reached. Without these data, the tuning has no sense. Defining precisely the requested performances by including all evaluation criteria may simplify the tuning validation. You can include in the technical specifications also the non-scientific criteria as for example getting a visibly fluid motion with no bothering noise. Every parameter can be analysed for the tuning validation.

**3. Move the motor**
To check the basic performances it is necessary to move the motor with realistic motions in the foreseen load conditions. You can start with some easy and slow motions and then move to the motions the machine is designed for. The motion must be carried out by starting from low working speeds till over the requested limit in order to check the system solidity. In presence of variable load the motion must be tested in the different configurations and in particular in the extreme and more demanding ones. To move the motor you can start

with the internal references generator (Section 20.6, *Function Generator*) and then use the motion controller which produces the working cycle the machine is designed for.

> ⚠ WARNING
>
> **Before moving the motor check if you can stop it in safety. If some errors or anomalies happen during the tuning, the motor can quickly reach a non-controlled speed, reaching quickly the position limit and hitting violently some other mechanical parts. To avoid such inconveniences enable all the necessary precautions and configure precisely all drive limits (Chapter 19, *Motion limits*).**

**4. Performance reached?**

To answer to this question you need to get the specifications about the requested motion performances; it is easier to understand when one motion is not acceptable than trying to understand the exact point where one acceptable motion becomes unacceptable. Most of the objective criteria are based on the numerical analysis of parameters as PositionFollowingError and SpeedFollowingError, in some particular points of the working cycle. For example: PositionFollowingError lower than X increment after Y milliseconds at the end of the acceleration ramp; SpeedFollowingError limited in % during a motion at constant speed; ActualTorque never higher than X % as to the selected limit. It is important to concentrate on those criteria that can guarantee the system reliable performances.

**5. Tuning the drive**

The tuning of the drives of the *i*BMD series must be done by using MotionDrive. To tune the loops use one of the following criteria:

- Section 20.3, *Fast tuning guide*
- Section 20.5, *Detailed tuning guide*

The loops that have to be tuned depending on the operative modes are listed in the following table:

| ModesOfOperationDisplay | CurrentLoop | VelocityLoop | PositionLoop |
|---|---|---|---|
| Torque | *YES* | *YES* | - |
| Speed | *YES* | *YES* | - |
| Position | *YES* | *YES* | *YES* |
| Homing | *YES* | *YES* | *YES* |

***Table 20.1. Loop that have to be tuned depending on the operative mode, reported in the .***

> ⚠ WARNING
>
> **If the motor is controlled in one of the torque modes, it's necessary to tune the velocity loop, because the "halt" and "quick stop" commands (see Section 22.3, *Carrying out a stop by using the master*) and the non fatal Fault reactions execute a deceleration ramp, controlled in velocity, to stop the motor.**

> ⚠ ATTENTION
>
> **Whatever is the selected ModesOfOperation, the Safety profile executes a movement that's controlled in position. To use the Security profile it's always necessary to tune the position loop.**

## 20.2. Reset the tuning

To take the tuning configuration back to a known condition, choose among the System Manager commands that are listed in the following table:

| Name | SysMng-Command | Button | Description |
|---|---|---|---|
| Set all loops, tuning and estimated parameters at default | 1101 | Set all loops, tuning and estimated parameters at default | Set the loop parameters (CurrentLoop, VelocityLoop, PositionLoop) of the TuningConfigurations and of the estimated parameters (InertiaEstimator and RLEstimator) to the default values |
| Parameter recalculation of all loops | 1102 | Recalculate all loops | Recalculation of the loop parameters in relation to the TuningConfigurations and to the estimated parameters |
| Parameter recalculation of motion loops | 1103 | - | Recalculation of the motion loop parameters in relation to the TuningConfigurations, to the estimated parameters and to the CurrentLoopEstimatedBandwidth |
| Parameter recalculation of current loop | 1110 | | Recalculation of the CurrentLoop parameters in relation to the TuningConfigurations and to the estimated parameters |
| Parameter recalculation of speed loop | 1120 | Recalculate | Recalculation of the VelocityLoop parameters in relation to the TuningConfigurations, to the estimated parameters and to CurrentLoopEstimatedBandwidth |
| Parameter recalculation of position loop | 1130 | | Recalculation of the PositionLoop parameters in relation to the TuningConfigurations, to the estimated parameters and to VelocityLoopEstimatedBandwidth |
| Permanent memory: reload value of loops parameters and tuning configuration | 2301 | Load loops parameters from permanent memory | Updating the loops parameters and the TuningConfigurations with the values in the permanent memory |

## 20.3. Fast tuning guide

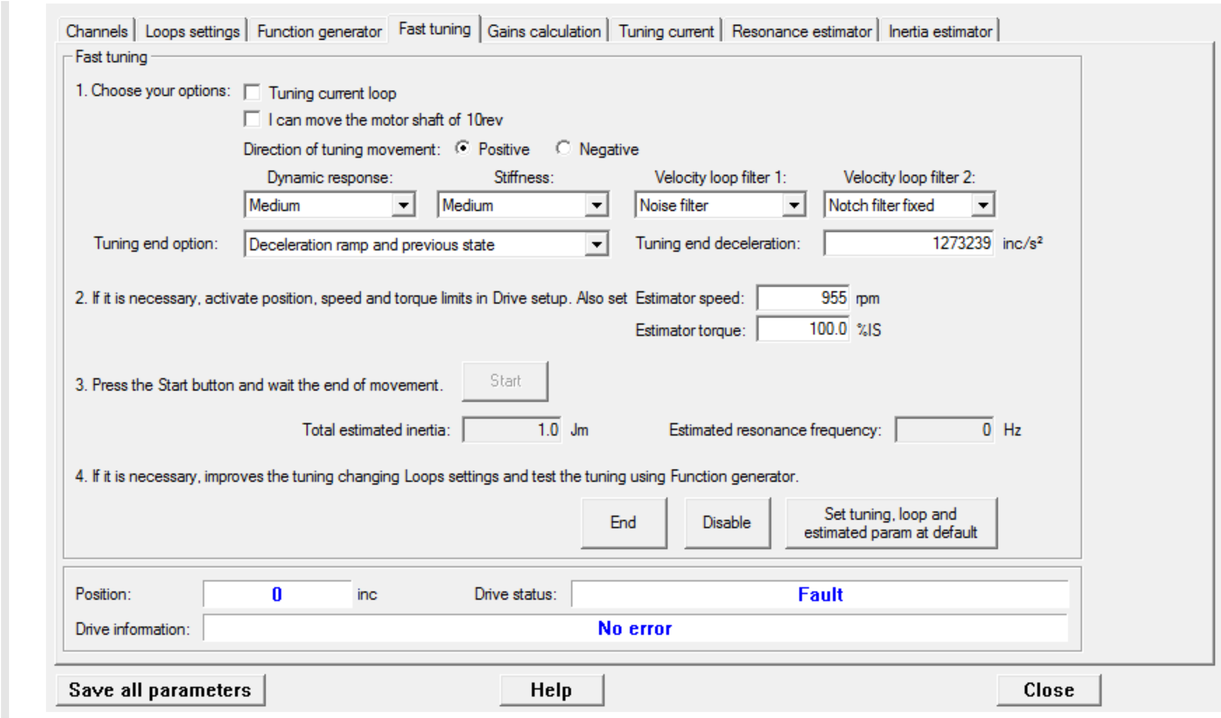
This guide contains the quick criterion to tune the drives of the *i*BMD series. The tuning must be run together with MotionDrive from the *Fast Tuning* Tab by following step by step the instructions reported in Section 20.1, *How to determine the tuning criterion*. Access:

**Main menu** > **Drive** > **Tuning...** > **Tab Fast tuning**

**Toolbar** >  > **Tab Fast tuning**.

ADVICE

**In case of problems or unexpected situations please see the Section 25.8, *Tuning problems*.**




NOTE

**The End button stops the tuning command according to Tuning end option settings.**

**The Disable button ends the tuning command by stopping the motor with maximum deceleration and resetting RequestedSpeed and then the drive enters the Switch On Disabled status.**

## 0. Setting of the default configuration

The first operation to do is taking the drive in the default tuning configuration: push the button Set tuning, loop and estimated param at default. In this configuration the motor can move with lowest dynamic performances; if the motion satisfies the requested performances, it is not necessary to run further tuning operations. If the performances were not satisfactory, continue with the following points.

## 1. Options selection

- *Tuning current loop* must be activated to improve the current loop performances (usually it is not necessary). For further details please see Section 20.7, *RL estimator*
- *I can move the motor shaft of 10rev* must be activated if the system mechanics allows the motor to safely move for 10 revolutions. This option requires more accurate the extimation of EstimatedInertia. If the motor can move for no more than 1 revolution, then it is however possible to proceed with the *Fast tuning* without selecting this

option. If, on the contrary, it is not possible to move the motor, it is necessary to abort the *Fast tuning* procedure and to execute the parameters calculation as reported in Section 20.4, *Gains calculation*.

- *Direction of tuning movement* allows to select the motor movement direction during the tuning operation.

- DynamicResponse influences the motor dynamic performances. When the requested dynamic answer increases, the motor answering times and the following errors are reduced. As against, while the dynamic response increases, the stability margins are reduced and eventual electrical or mechanical resonances are amplified. These resonances are not always eliminated by using the filters but it has to be willing to accept a more limited dynamic answer.

- Stiffness modifies the motor stiffness when it is stopped in torque. When the stiffness increases, the motor is more able to stay steady; on the contrary the stability margins are reduced and any possible electrical and mechanical resonances amplified as happening for the option DynamicResponse.

- The VelocityLoopFilter1 works on the first filter of VelocityLoop and on the sensor filter and can take the following values:
  – User: the tuning commands don't modify the filter parameters
  – Soft filter: the filters are modified to make a *sweet* filtering action of the noise that's present in the loop.
  – Noise filter: the filters are modified to make a *heavy* filtering action of the noise that's present in the loop.
  – Disable: the filtering action for the noise that's present in the VelocityLoop is deleted. In this case it can be obtained a faster dynamic response.

- The VelocityLoopFilter2 works on the 2 and 3 filters of the VelocityLoop and can take the following values:
  – User: the filters parameters are not modified.
  – Resonance filter: a single filter is inserted to eliminate the constant frequency mechanical resonances
  – Double resonance filter: 2 filters are inserted to *strongly* eliminate the constant frequency mechanical resonances,
  – Debounce filter: a single Low-pass filter of the first order is inserted, in order to limit the unwanted effects of the mechanical transmissions if it is not stiff or if it has backlashes.
  – Notch filter fixed: a Band-eliminating filter is inserted and set to 350Hz.
  – Disable: the filtering action is deleted.

- TuningEndOption and TuningEndDeceleration define the operation that are executed when the *End* button (command 100 of the System Manager) is pressed or at the end of the tuning movement.

**2. Movement limits selection**

If necessary, set the movement limits according to what's reported in Chapter 19, *Motion limits* and insert the values of InertiaEstimatorVelocity and EstimatorTorque that will be used during the tuning movement. In most cases it is not necessary to modify the default values of these two parameters.

**3. Tuning movement**
By pressing the *Start* button the tuning procedure starts, and automatically stops when the progress bar disappears.

**4. Results verification**
When the tuning movement is concluded, check that there are no anomalies reported in the *Drive information* field. The oscilloscope shows the step response of the *Velocity loop*. Evaluate the response and, eventually, modify the tuning, according to what's reported in Section 20.4, *Gains calculation*.
To better calculate the total inertia moment as to the motor shaft, see what's reported in Section 20.9, *Inertia estimator*.
To better calculate the mechanical resonances as to the motor shaft, see what's reported in Section 20.8, *Resonance estimator*.

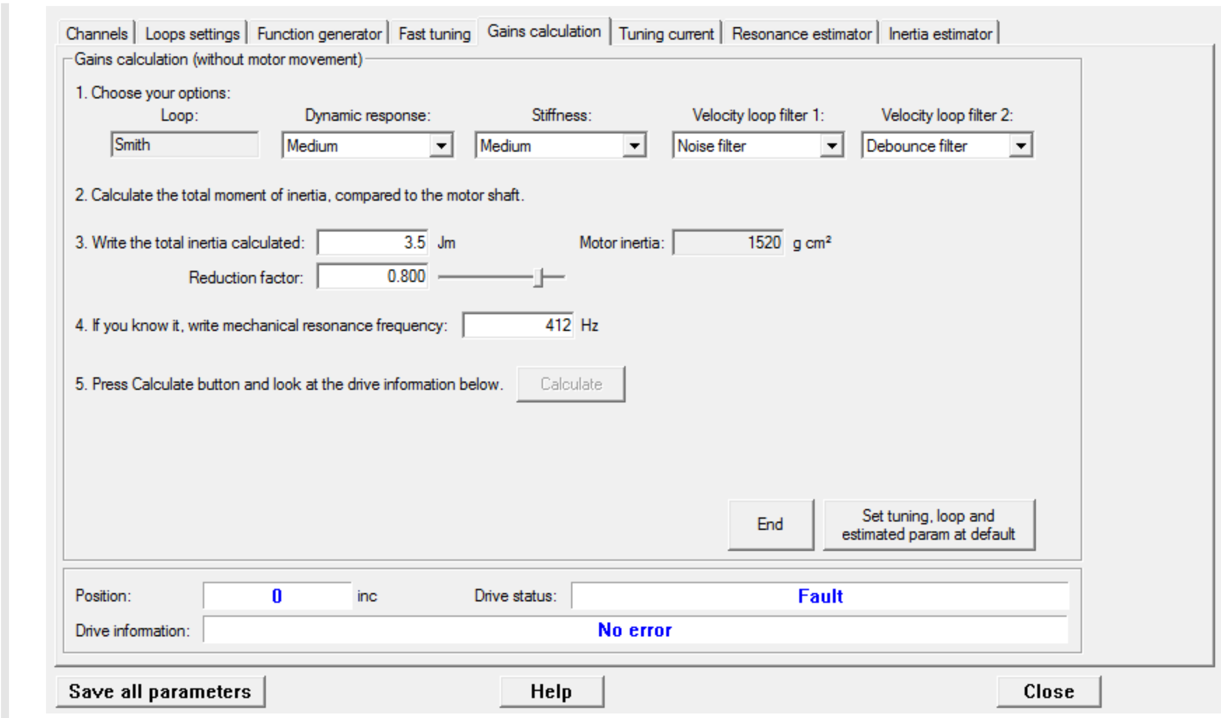
## 20.4. Gains calculation

This functionality allows to tune the loops parameters without to move the motor. Follow the instructions reported in the *Gains calculation* Tab.

Access with MotionDrive:

**Main menu** > **Drive** > **Tuning ...** > **Tab Gains calculation**

**Toolbar** > > **Tab Gains calculation**



**1. Options selection**
Select the desired configuration, selecting between the available options. The option selection criteria are reported on the point 1 of the Section 20.3, *Fast tuning guide*.

**2. Total inertia moment calculation**

NOTE

**If the "Fast Tuning" has already been executed, it's possible to directly jump to the next point.**

EstimatedInertia is the total inertia moment and includes the motor, the brake, the mechanical transmission and the load. A precise analytical estimate of the inertia moment is often very complex: you can accept even a gross estimate but probable. Bear in mind the rules of the inertia moment calculation, in particular the conversions to do between linear motion and rotatory motion, the conversions in presence of reducers and mechanical connections in general and the formulae to calculate the inertia moment of the more common solid objects. If the mechanical allows the motor movement, it is advisable to execute the calculation of EstimatedInertia through the dedicated extimator (see Section 20.9, *Inertia estimator*).

**3. Settings of the total inertia moment**
Write the calculated inertia value in the dedicated field, and if one of the following cases happens:

- loads with inertia moment greater than 5 Jm, without friction or dumping
- Not rigid mechanical transmission
- consistent backlashes and tolerance in the mechanical transmission

that cannot be solved through the filters options, it's necessary to tune the motor as if the total inertia moment is lower than the estimated value. To do this it's necessary to decrease the value of the InertiaReductionFactor parameter (try with 0.8, 0.5, 0.3). Too low values of InertiaReductionFactor cut down the motor dynamic performances.

**4. Mechanical resonance**

NOTE

**If the "Fast Tuning" has already been executed, it's possible to directly jump to the next point.**

EstimatedResonanceFrequency is the mechanical resonance frequency estimated value. If there are mechanical resonances it's advisable to pay attention to the contents of Section 20.8, *Resonance estimator*.

**5. Parameters calculation**
Execute the parameters calculation by pressing the *Calculate* button and verify that in the *Drive information* field there are no anomalies reported. If the desired performances have

not been reached, restart from the point 1 and select different options, or execute a tuning according to what's described in Section 20.5, *Detailed tuning guide*.

## 20.5. Detailed tuning guide

This guide contains the detailed criterion described about the drive tuning of the *i*BMD series. The tuning must be carried out together with MotionDrive according to the instructions in the Section 20.1, *How to determine the tuning criterion*: each tuning operation must be carried out together with the check of the performances through the motor motion.

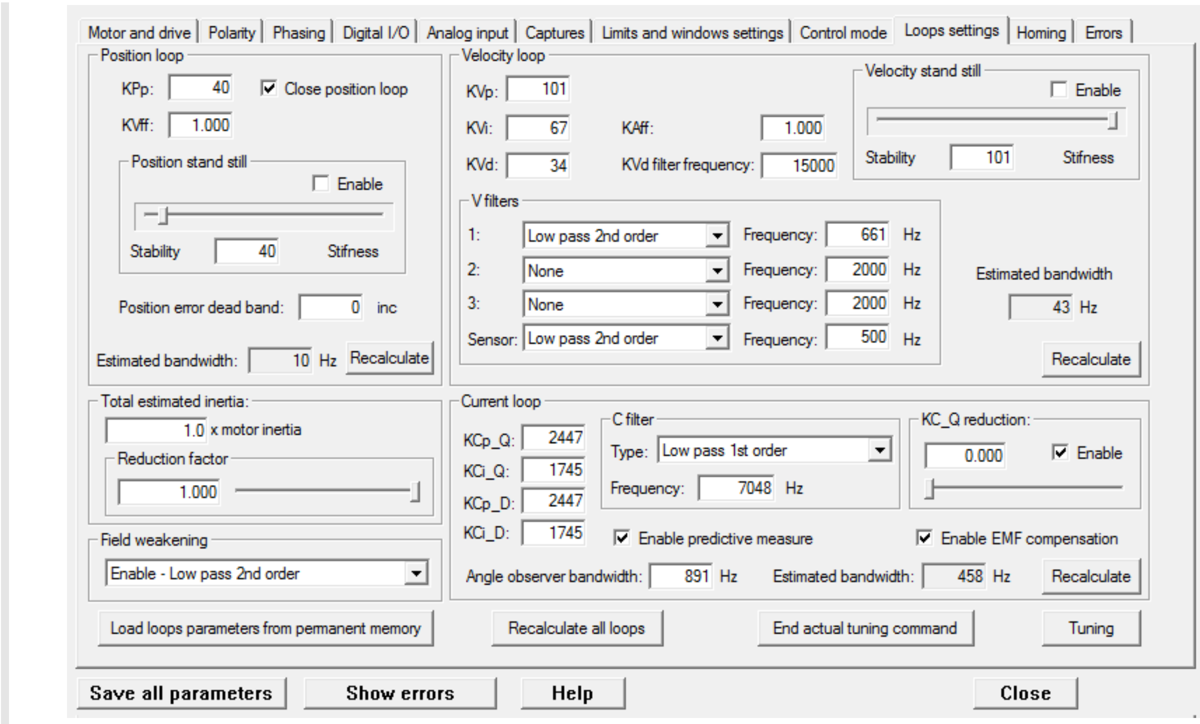
ADVICE

**In case of problems or unexpected situations please see the Section 25.8, *Tuning problems*.**

Unless differently specified the operations can be run from the tab Loops settings of MotionDrive. Access:

**Main menu** > **Drive** > **Tuning ...** > **Tab Loops settings**

**Toolbar** > > **Tab Loops settings**




If you cannot find a parameter in the page, use the parameters vocabulary (Section 27.10, *Loop (1500-1599)*).

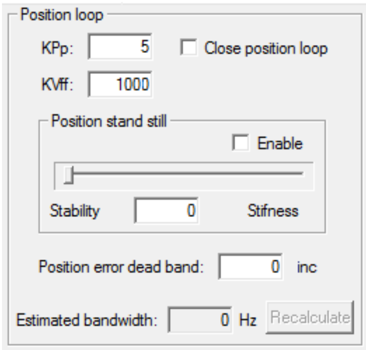
### 1. Parameters estimation

The first operation to be executed is the parameters estimation according to the procedure described in Section 20.7, *RL estimator*, Section 20.8, *Resonance estimator* and Section 20.9, *Inertia estimator*.

**2. PositionLoop inhibition**

During this step the focus is on the *VelocityLoop* tuning. The *PositionLoop* must be strongly inhibited so that it doesn't influence the dynamic of the *VelocityLoop*. During this step it is accepted that PositionFollowingError is slowly controlled. Proceed in this way.

- disable the EnablePositionStandStill option
- set KPp equal to 4÷5 units when you have EstimatedInertia lower than 8 Jm
- set KPp equal to 2÷3 units when you have EstimatedInertia higher than 8 Jm
- prefer lower values than KPp in case of increase of EstimatedInertia
- KPp can be reset, if it is not important to hold the position or you are going to control the motor only in the velocity modes;




**3. System stabilisation**

If, on the motor enabling or during a low velocity movement, the system is not stable, it's necessary to stabilize it before to proceed with the tuning. There may be many causes for the instability:

- Presence of a mechanical risonance with a constant frequency. In this case through the *Resonance estimator* it must be obtained the EstimatedResonanceFrequency and it is necessary to cut it off by using the *Resonance filter* or *Double resonance filter* options, in the *Gains calculation* Tab.

  IMPORTANT

  **The Resonance filter or Double resonance filter options use is useful only if the EstimatedResonanceFrequency is constant (see Section 20.8, *Resonance estimator*).**

- Too high gains, compared to the load mechanical characteristics (loads with an inertia moment greater than 5 Jm without friction or damping, not rigid mechanical transmission, presence of heavy backlash or tolerance on the mechanical transmission, ...).
  In this case it's necessary to decrease the required dynamic performances by lowering the DynamicResponse option and/or decreasing the InertiaReductionFactor parameter value (try with 0.8, 0.5, 0.3; too low values of InertiaReductionFactor cut down the motor dynamic performances) and/or by selecting the *Debounce filter* option in the *Gains calculation* Tab.

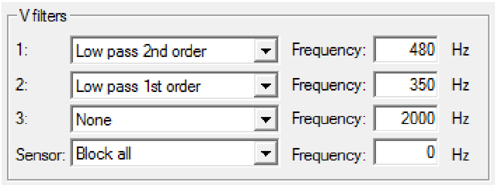
If the operations that have been made in the *Gains calculation* Tab are not sufficient to stabilize the system, pay attention to what's reported in 4. Filters.

IMPORTANT

**On every action performed in the Gains calculation Tab, repeat the point 2(PositionLoop inhibition).**

**4. Filters**

The filters of the VelocityLoop should be used only if strictly necessary in order to cut off any eventual regulation noise or resonance. To evaluate the need of a filter, try to disable it or, in case it is a low pass filter, try to increase its cutoff frequency. The filters right configuration depends on the load and transmission mechanical characteristics. There is no systematic regulation method, but it is advisable to act on the three regulator filters and then on the feedback sensor filter.




Proceed with some trials and progressively define the strategy to improve the performances. Test the following strategies (some may not be effective):

- remove the Band-eliminating filter; select *Type None*
- insert a Low-pass filter of the second order as first filter and increase or decrease the frequency with steps of 50-100-200 Hz; if there are improvements with frequencies higher than 1800 Hz, maybe it's possible to remove the filter, by selecting *Type None*
- insert a Low-pass filter of the first order instead of Low-pass filter of the second order as first filter; look again for an optimal filter frequency
- enable the other two filters to increase and modify the filtering action
- increase or decrease the sensor filter frequency with steps of 50 Hz
- insert a Low-pass filter of the first order in place of Low-pass filter of the second order as feedback sensor filter
- search again for the filter frequency on the feedback sensor
- if a Band-eliminating filter is used, check its performance and modify its selectivity through the quality factor Q (for example, for the filter 2, by using the VFilter2QFactor parameter).

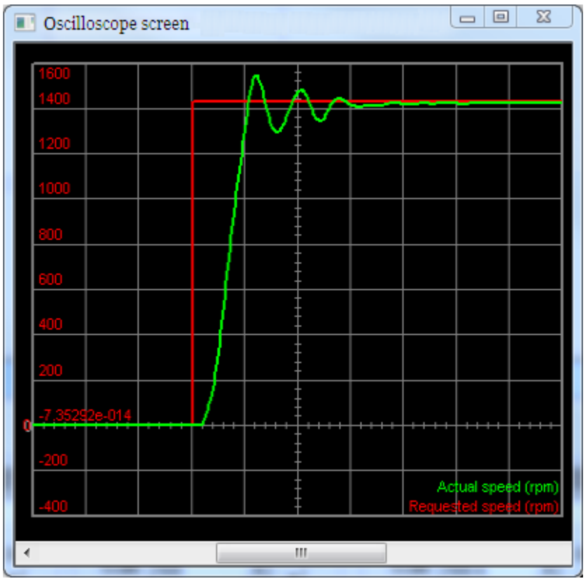


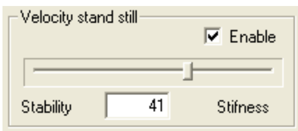

*Figure 20.1. Response to the step where can be observed the start of a resonance, then damped.*

ADVICE

**In order to reduce the velocity ripple, in presence of sin-cos feedback sensors with a limited sinusoids per revolution number, or in presence of resolvers, it is advised to decrease the AngleObserverBandwidth parameter of 50-100Hz at a time even until reach values near 50Hz, for very slow dynamics. If VFilterSensorType is a low pass filter and AngleObserverBandwidth is lower than VFilterSensorFrequency, try to disable VFilterSensorType because it probably hasn't any effect on the system. Keep in mind that decrease AngleObserverBandwidth causes an increasing of the velocity overshoot. Try to find a compromise between the velocity ripple limitation and the overshoots increasing.**

**5. Velocity stand still**

Activate the EnableVelocityStandStill option and, by moving the motor with low velocity (with low velocity is meant velocities that are lower than the 30% of the HighSpeed parameter), move the VelocityStandStill bar as much as possible toward *Stiffness*, in order to increase the motor quickness and stiffness. Moving the bar towards *Stability* eventual noises or resonances are attenuated. Don't move the bar to values lower than 20 units, because the quickness of the motor will be deteriorated.




**6. Special parameters**

If the resonances persist, modify the following parameters (not all of them are reported in the *Loops settings* Tab), while continuing to test the motor with low speed. Please refer to Figure 27.1:

- modify KVd, even up to set it to 0. Also try to modify only the filtering action through the KVdFilterFrequency parameter
- decrease WVd and WVp even up to set them to 0

- progressively increase KVc in order to increase the damping effect, try with steps of 20-50-100 units.

**7. Stopped motor**

Run some stability tests when the motor is stopped in torque. If possible, disturb the mechanical load from outside with the motor stopped in torque to test the motor ability to absorb and dampen the resonances. In case of unwanted effects, try to modify the filters or the VelocityStandStill parameter (verify that the EnableVelocityStandStill option is enabled).

**8. Quick decelerations**

When the deceleration increases, the possibility to get resonances increases when the motor ends the deceleration ramp. Run some tests with the requested decelerations, in case of unwanted effects readjust the filters or the parameter VelocityStandStill. If the resonances at the end of the ramp persist, it's necessary to limit the required working decelerations.

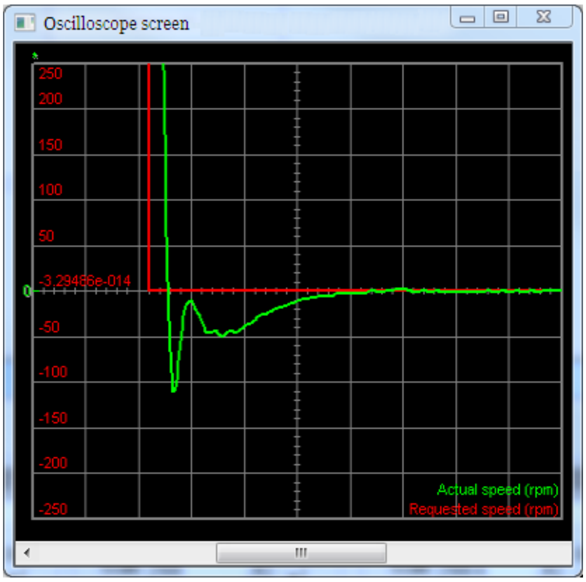


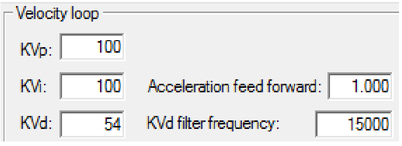

*Figure 20.2. Step answer in deceleration with limited overshoot without resonances and prompt error resetting.*

**9. Working speed**

Proceed with tests with greater velocities, but never greater than the limits; start with a velocity equal to 50% of the HighSpeed parameter and increase the velocity over the required working velocity. The speed profile to generate can be the one the machine is designed for. In these tests modify the parameters KVp and KVi, with the following criteria:

- Increase KVp and KVi to make the system more quick, try with steps of 20% till the system becames unstable. These parameters have greater effect for speeds higher than HighSpeed if the EnableVelocityStandStill option is enabled.
- decrease KVp and KVi to make the system more stable and eliminate the resonances, proceed with decreases of 20% until the system becomes stable. If the option EnableVelocityStandStill is enabled, these parameters have less effect for speeds low-

er than HighSpeed. If there are some resonances for speeds much lower than HighSpeed, readjust VelocityStandStill and the filters.




NOTE

**If the movement noise level or the noise that's overlapped to VelocityActualValue don't have reached the desired levels, it's necessary to work again on the regulator filters and parameters. Check that the noise cause is not due to the noise present in the velocity reference (RequestedSpeed).**

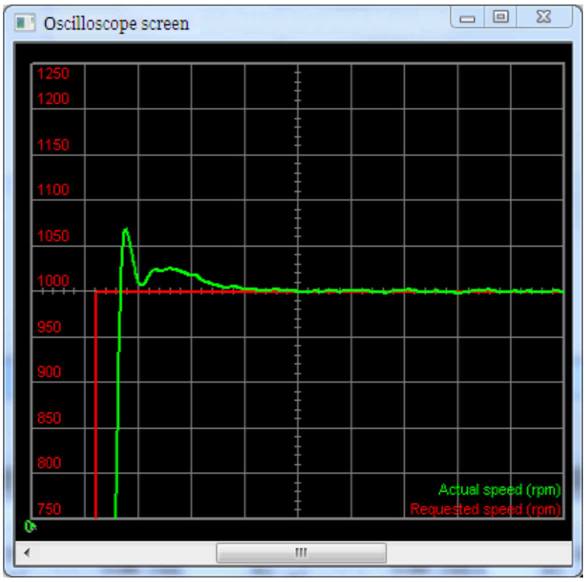



*Figure 20.3. Step answer with acceptable overshoot without resonances and prompt error resetting.*

### 10. Feed forward acceleration

To adjust the parameter KAff move the motor by commanding accelerations and decelerations similar to the machine working ones. Increase or decrease KAff in order to minimize SpeedFollowingError during the acceleration and deceleration ramps. If FeedForwardAcceleration is noisy, it may be useful to reset to zero KAff in order to reduce the noise that enters in the loop. If it's not necessary to tune the *PositionLoop*, then the tuning is finished.

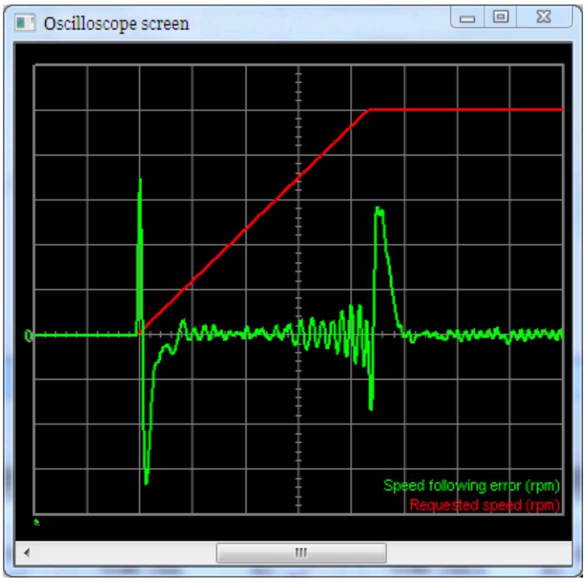
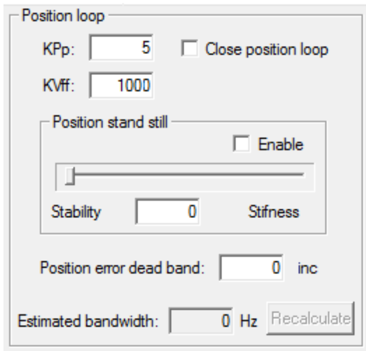



*Figure 20.4. Sign change of SpeedFollowingError at the ramp beginning: KAff too high.*

**11. PositionLoop (low velocities)**

When the *VelocityLoop* is tuned in the best possible way, the *PositionLoop* tuning becomes very easy. Execute some movement with position reference with speeds lower than HighSpeed and with the motor stopped in torque, in order to check the following tuning operations:

- Push the *Recalculate* button in the *PositionLoop* area
- activate the EnablePositionStandStill option and modify PositionStandStill by valuing the effects on the PositionFollowingError. Increase its value to increase the resetting speed of the PositionFollowingError; decrease its value to eliminate not damped oscillations at low speeds. With the bar at 0, PositionFollowingError is not controlled




**12. PositionLoop (high velocities)**

Proceed with tests with greater velocities, but never greater than the limits; start with a velocity equal to 50% of the HighSpeed parameter and increase the velocity over the required working velocity. The position profile that has to be generated should be the one the machine is made for. Check the following tuning operations:

- set the value of KPp equal to the value of PositionStandStill

- modify KPp by valuing the effects on the PositionFollowingError. The effect of this parameter is greater for velocities higher than HighSpeed. Increase its value in order to maintain low the PositionFollowingError; decrease its value if vibrations or resonances appear
- check if PositionLoopEstimatedBandwidth is lower at least 0.7 times VelocityLoopEstimatedBandwidth.

## 20.6. Function Generator

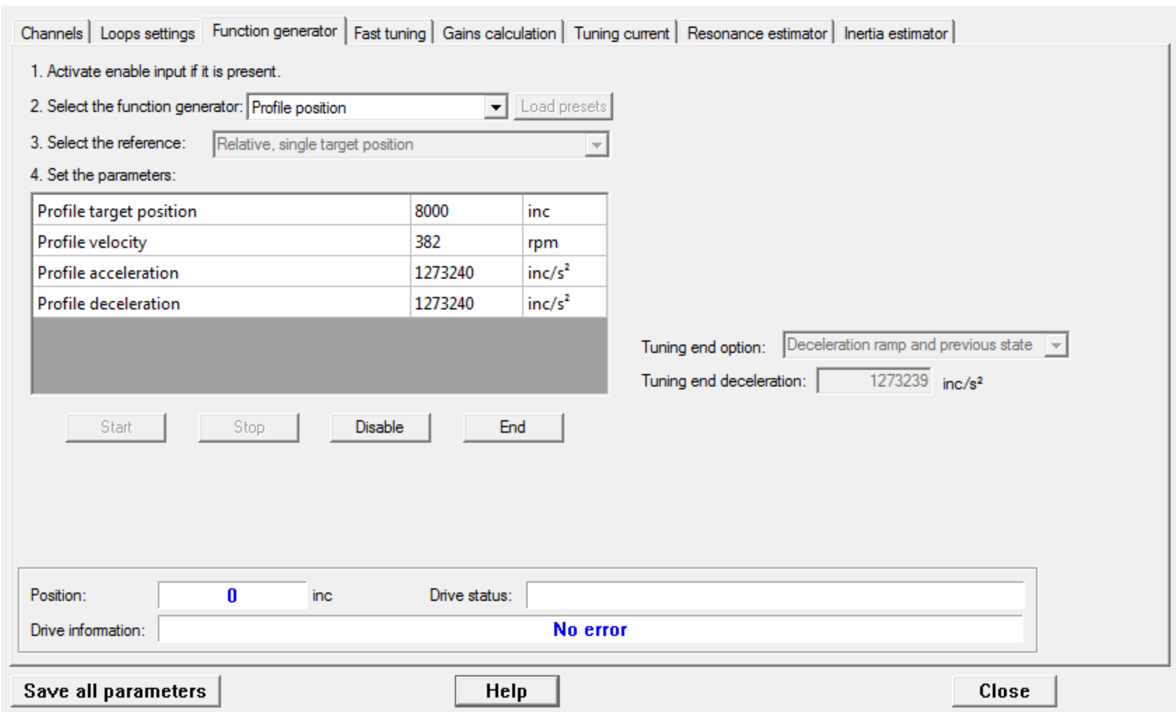
> ⚠ WARNING
>
> **Before enabling the Function Generator when the drive is not adjusted yet, check if it is possible to stop the motor in safety. To avoid unwanted motions or collisions, take all the necessary precautions and configure precisely the drive limits (Chapter 19, *Motion limits*).**

The Function Generator is an integrated functionality of the MotionDrive oscilloscope applying some particular references to the control loops. When a Function Generator command is requested, the oscilloscope capture is enabled to value the drive performances through the analysis of the progress of some particular parameters.

Access:

**Main menu** > **Drive** > **Loops settings and tuning** > **Function Generator Tab**

**Toolbar** >  > **Tab Function Generator**




In the following table you can find the Function Generator functionalities:

| Functionalities | Description | |
|---|---|---|
| Select the Function Generator | Choose the Function Generator type | CurrentLoop D applies to RequestedField the generated reference |
| | | CurrentLoop Q applies to RequestedTorqueCurrent the generated reference |
| | | Speed loop applies to RequestedSpeed the generated reference |
| | | Profile velocity generates a speed motion with linear acceleration ramps |
| | | Profile position runs some positionings with linear acceleration ramps |
| Load presets | Set some default values for the selected reference and the oscilloscope | |
| Select the reference | Select the reference type | Stop |
| | | Step |
| | | Step (time limited) |
| | | Square wave |
| | | Square wave (time limited) |
| | | Sinusoidal wave |
| | | Sinusoidal wave (time limited) |
| | | Profile velocity unlimited standard |
| | | Profile velocity time limited |
| | | Profile velocity time limited, forward and backward, single sequence |
| | | Profile velocity time limited, forward, multiple sequence |
| | | Profile velocity time limited, forward and backward, multiple sequence |
| | | Profile position, absolute target position |
| | | Profile position, relative, single target position |
| | | Profile position, relative, forward and backward, single sequence |
| | | Profile position, relative, forward, multiple sequence |
| | | Profile position, relative, forward and backward, multiple sequence |
| Start | Start the reference | |
| Stop | Stop the reference and keep the drive enabled | |
| Disable | It ends the Function Generator by stopping the motor with maximum deceleration and resetting RequestedSpeed and then the drive enters the *Switch On Disabled* status | |
| End | End the Function Generator following Tuning end option | |
| TuningEndOption | Options for the Function Generator end command | Immediately disable, the motor is stopped with maximum deceleration resetting RequestedSpeed to zero and then the drive has the *Switch On Disabled* state |
| | | Declaration ramp, the motor is stopped with deceleration equal to Tuning end deceleration |
| | | Zero speed, the motor is stopped with maximum deceleration by resetting RequestedSpeed |
| TuningEndDeceleration | Deceleration for the Function Generator end command | |
| Drive status | Drive status (Statusword) | |
| Drive information | Function Generator status (SysMngError) | |

When using the Function Generator, bear in mind as follows:

1. it is not possible to change Function Generator without using the End command
2. to modify the reference type or the reference parameters, when a reference is started, you have to use a command of Stop, End or Disable

3. before starting a reference, set its parameters and Tuning end option and Tuning end deceleration

4. at the references end, that end after a given period, the drive is always enabled

5. the references of time limited type end after a time period equal to the parameter Duration

6. the references of sequence type generate some profiles which are separated among them by a time period equal to the parameter Profile interval

7. when a reference is started and you close the window Oscilloscope, the End command is run

8. if the motor can run motions only in one direction, set properly the reference parameters

9. if the requested reference ends before the end of the answer transient, increase properly the reference parameters to increase its duration

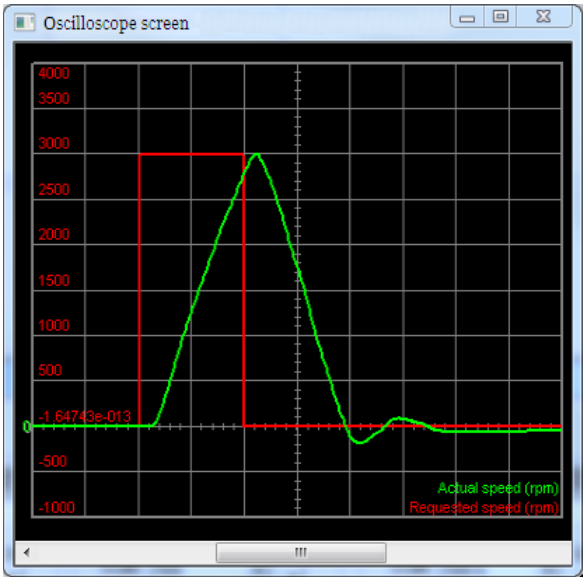



*Figure 20.5. Example of reference concluding before the end of the speed transient.*

10. if the oscilloscope capture ends before the Function Generator has finished and it does not show all the reference and answer progress, increase the oscilloscope sampling time.

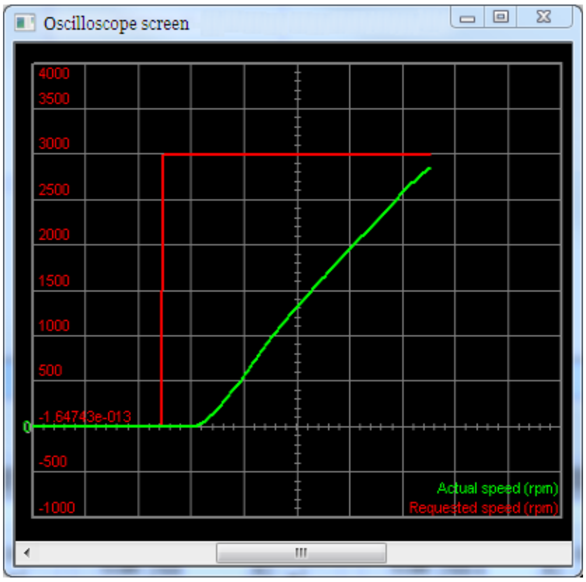



*Figure 20.6. Example of too short capture ending before the Function Generator has finished.*

## 20.7. RL estimator

To optimize the current loop performances it's necessary to estimate the effective value of the phase resistance and of the synchronous motor inductance. *RL Estimator* executes an offline estimation of these parameters through the application of some ramps and pulses of current to the motor phases. During the RL estimation command the drive may move the motor shaft up to 1 polar step. According to the estimated inductance value (see RLEstimator group parameters), the parameters of the speed and position loops are calculated again.

⚠ NOTICE

**During the RL estimation, the motor is free to move. Therefore, in case of vertical load or relevant disturbing forces on the load, the estimation should be executed unmounting the motor and without mechanics and not with the motor connected to the machine mechanics.**

IMPORTANT

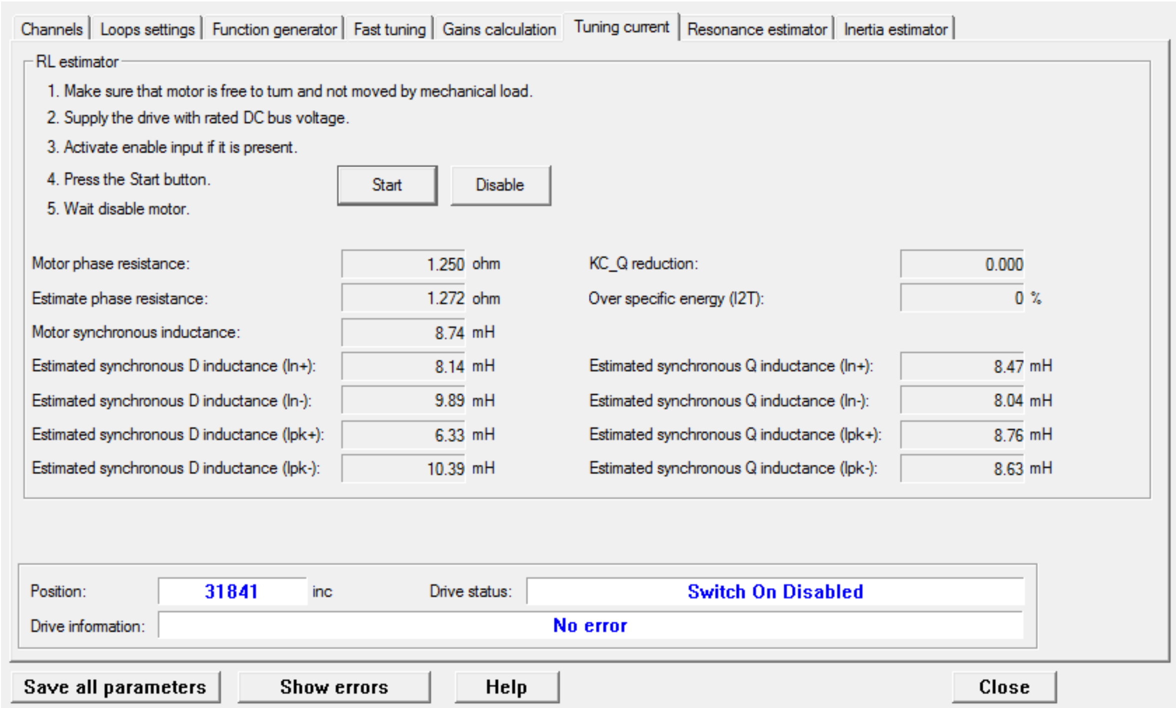
**The EstimatedPhaseResistance, EstimatedLDNominalP, EstimatedLDNominalN, EstimatedLDPeakP, EstimatedLDPeakN, EstimatedLQNominalP, EstimatedLQNominalN, EstimatedLQPeakP, EstimatedLQPeakN parameters cannot be downloaded through the parameters file because they are specific for every axis. On the contrary, the parameters file download reset them to the default value.**

To estimate RL follow the instructions reported in the *RL Estimator* area in the *Tuning current* Tab.

Access with MotionDrive:

**Main menu** > **Drive** > **Tuning ...** > **Tab Tuning current**

**Toolbar** >  > **Tab Tuning current**




**⚠ NOTICE**

**During the RL estimation the I2TValue value, that can be read in the "Over specific energy (I2T)" box of the same page, increases. Always wait that it takes the 0 value.**

**The end of the RL estimation command, coincides with the automatic disabling of the motor. Always wait its disable.**

**IMPORTANT**

**Considering that the current pulses reach MotorPeakCurrent, be sure that the voltage supply DCBusVoltage(+HV) remains stable during the command.**

**ADVICE**

**To correctly estimate RL follow the instruction list in the Tuning current Tab.**

**NOTE**

**The oscilloscope is not activated because a video diagnostics is not necessary. The results are reported in the Tuning current Tab.**

## 20.8. Resonance estimator

The *Resonance Estimator* serves to evaluate if some mechanical resonances with *constant frequency* are present. Through the application of a variable torque, with null average, with maximum amplitude of EstimatorTorque, in order to generate a controlled vibration, the drive obtains and shows the frequency response of the mechenical load and determines EstimatedResonanceFrequency. The main chart reading purpose is to point out the eventual resonance peaks, their frequency and their band width. The resonance peak can be eliminated by setting a band-stop filter with the peak central frequency.

IMPORTANT

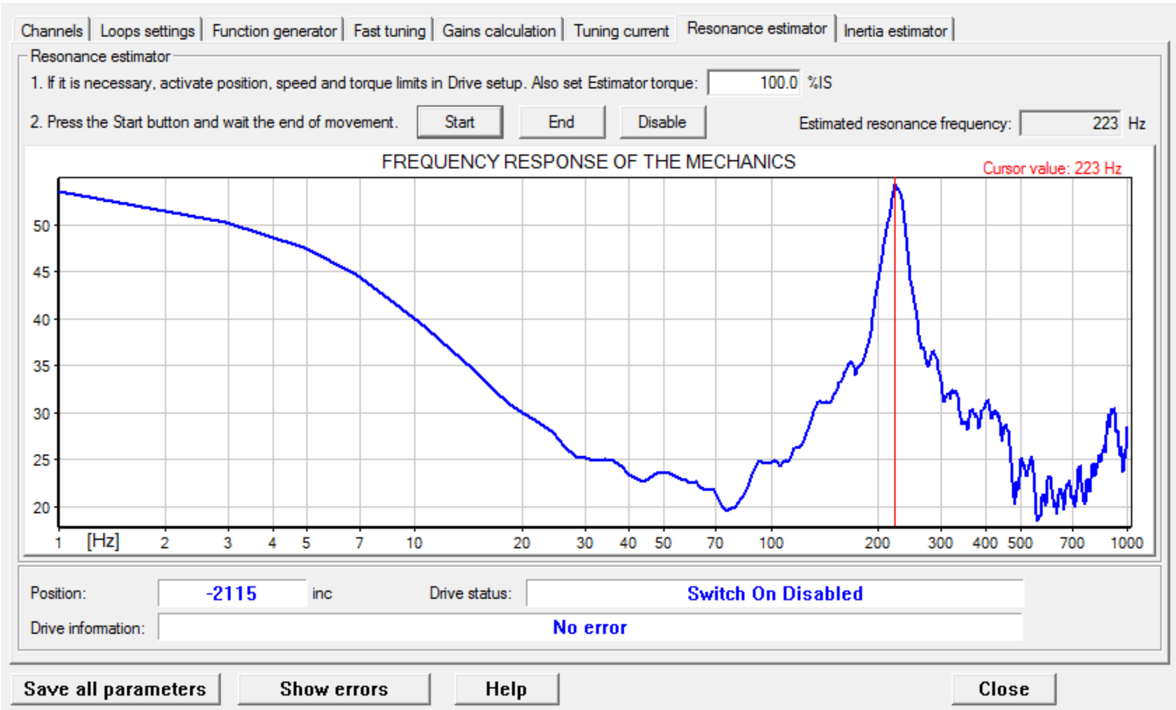
**During the mechanical resounances estimation, the motor is free to move. Therefore, in case of vertical load or relevant disturbing forces on the load, the estimation cannot be executed.**

For a correct estimation of EstimatedResonanceFrequency follow the instructions reported in the *Resonance estimator* Tab:

Access with MotionDrive:

**Main menu** > **Drive** > **Tuning ...** > **Tab Resonance estimator**

**Toolbar** > > **Tab Resonance estimator**




ADVICE

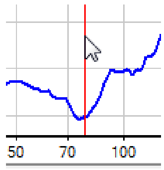
**Try with different values of EstimatorTorque.**

**According to the machine structure and typology, try by positioning the load that's connected to the motor, in different positions of the working cycle.**

IMPORTANT

**If when the EstimatorTorque value or the axis position change, the resonance frequency peak moves in the chart for a significant quantity, don't use the Resonance filter or Double resonance filter options, in the Gains calculation tab, because a band-stop filter is not sufficient to cut off the resonance.**

NOTE




**To read more precisely the value of the frequency peaks showed on the chart, drag and drop the red cursor to the desired position.**

## 20.9. Inertia estimator

⚠ WARNING

**Before to activate the Inertia estimator, be sure that the motor can be safely stopped. To avoid unwanted motions or collisions, take all the necessary precautions and configure precisely the drive limits (Chapter 19, *Motion limits*).**

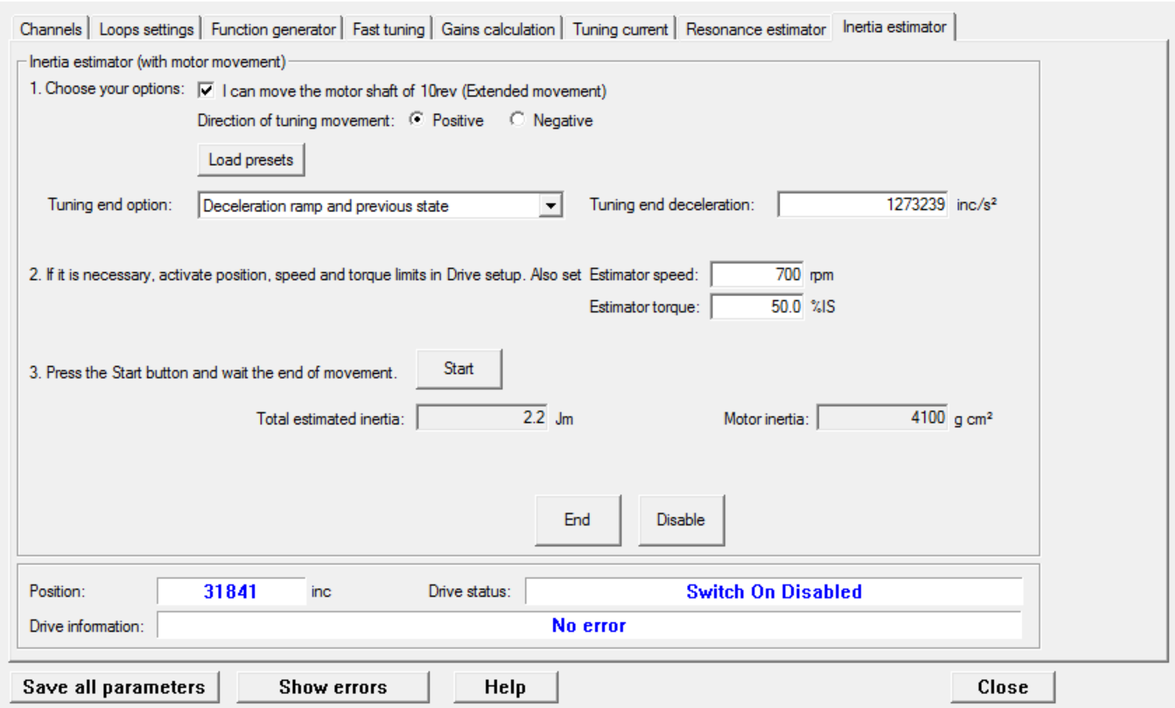
The *Inertia estimator* executes a controlled movement of the motor in order to estimate the total inertia moment, calculating according to the motor shaft. The estimated inertia moment (Total EstimatedInertia) is used to automatically calculate the parameters of the speed and position loops. Follow the instructions reported in the *Inertia estimator* Tab.

IMPORTANT

**In case of vertical load or relevant disturbing forces on the load, the estimator produces wrong results. Its use is not recommended.**

Access with MotionDrive:

> **Main menu** > **Drive** > **Tuning ...** > **Tab Inertia estimator**
>
> **Toolbar** >  > **Tab Inertia estimator**




### 1. Options selection

- *I can move the motor shaft of 10rev* must be activated if the system mechanics allows the motor to safely move for 10 revolutions. This option requires more accurate the extimation of EstimatedInertia. If the motor can move for no more than a quarter of revolution, it's anyway possible to proceed with the estimation without selecting this option. If, instead, it's absolutely not possible to move the motor, the estimation can't be executed.

- *Direction of tuning movement* is used to select the direction of the movement of the motor during the estimation.

- *Load presets* automatically sets the oscilloscope for a correct estimation evaluation.

- TuningEndOption and TuningEndDeceleration define the operation that are executed when the *End* button (command 100 of the System Manager) is pressed or at the end of the tuning movement.

### 2. Movement limits selection

If necessary set the movement limits according to what's reported in Chapter 19, *Motion limits* and insert the InertiaEstimatorVelocity and EstimatorTorque values that will be used during the estimation. In most cases it is not necessary to modify the default values of these two parameters.

### 3. Estimation movement

By pressing the *Start* button the estimation procedure starts, and automatically stops when the progress bar disappears. Then, check that there are no anomalies reported in the *Drive* *information* field. If the *Load presets* button has been pressed, the oscilloscope automatically shows the trend of some variables that are important to evaluate the estimation quality.

#### 4. Advised procedure to estimate the inertia moment

a. If possible, select the *Extended movement* option. Select the estimation direction and push *Load presets*.
b. Set the motor in order to run the requested motion. Take all the necessary precautions and configure precisely the drive limits (Chapter 19, *Motion limits*).

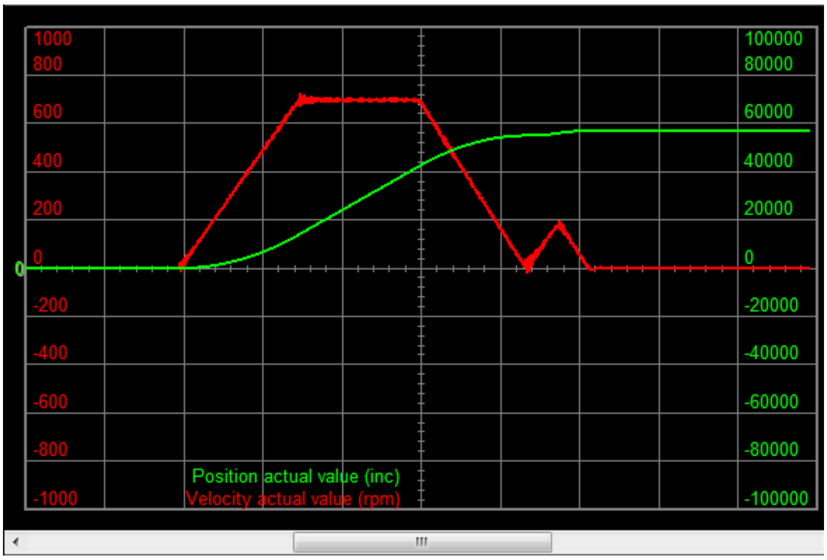



***Figure 20.7. Example of speed and position progress obtained through the selected option Extended movement (EncoderIncrements is 8000 inc/rev).***

c. Press *Start*.
d. When the movement is finished, control the informations that are reported in the *Drive information* field.
e. Unselect the *Extended movement* option.
f. Position the shaft so that it is possible to run the requested motion and check if the shaft reaches the machine position limits.
g. Press *Start*.
h. When the movement is finished, control the informations that are reported in the *Drive information* field.
i. Check if the result of the estimate Total EstimatedInertia, is approximately coherent with the applied load.
j. Run for some times the tuning command by starting from the point f and check if the estimation does not sensibly change. Variations of 10-20% can be tolerated.

ADVICE

**In case of heavy friction, increase EstimatorTorque up to the nominal value.**

#### 5. Inertia moment estimation verification

To verify the estimation, use the oscilloscope. If the *Load presets* button has been pushed, the oscilloscope is automatically set for this aim.

During the first phase of the estimation, while the torque step is applied, the velocity should be a linear ramp (with the *Extended movement* option unselected).

ADVICE

**In general, if there are some vibrations during ther first phase of the estimation, try to repeat decreasing the value of EstimatorTorque.**
**In order to limit the velocity and the position reached during the estimation, the InertiaEstimatorVelocity value can be decreased.**
**In case of transmission with chains, backlash or plays, reduce the value of EstimatorTorque in order to maintain as linear as possible the velocity ramp, during the first step of the estimation.**

- During the first phase of the estimation, the more the velocity ramp is linear the better the estimation is (see the following figure).

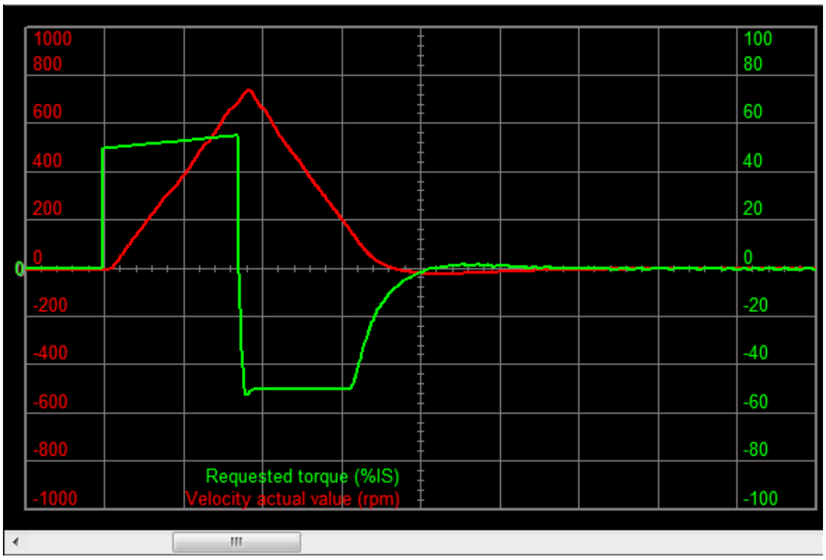



***Figure 20.8. Example of a correct estimate of the inertia moment.***

- If there are vibrations during the first phase of the estimation, but the velocity profile is quite a linear ramp, specially in the final part of the ramp, the estimation can be considered reliable (see the following figure). Consider however, during the tuning, a possible presence of mechanical elasticities of the load (see Section 25.8, *Tuning problems*).

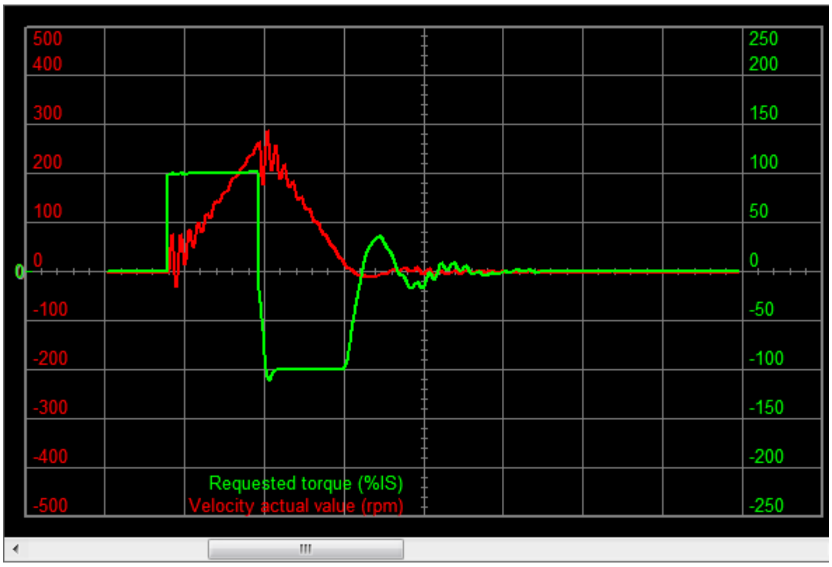



*Figure 20.9. Example of a correct estimation of the inertia moment with vibrations.*

- If there are some heavy vibrations during ther first phase of the estimation, it's probable that the mechanical transmission is too elastic (see the following figure). In this case the estimation is not reliable and it is necessary to make the correct operations during the tuning (see Section 25.8, *Tuning problems*).

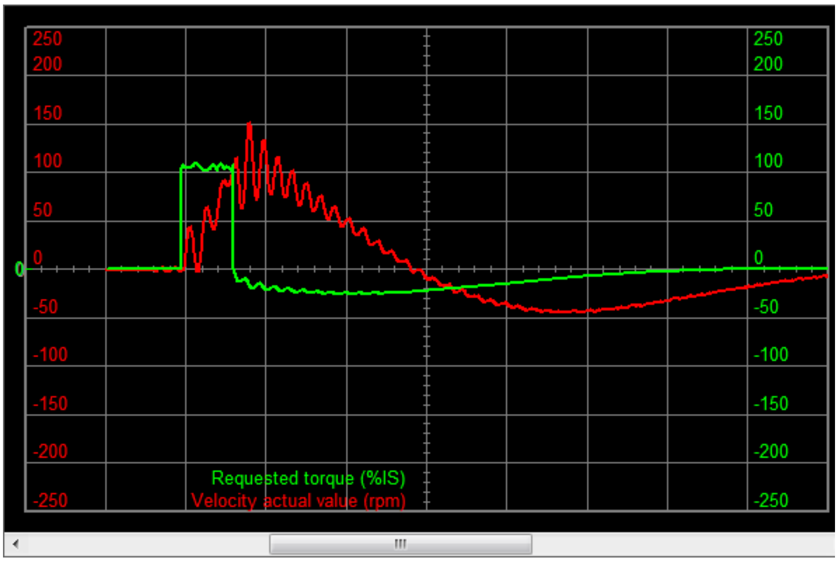



*Figure 20.10. Example of a not valid estimation, with not rigid mechanical transmission.*

- During the first phase of the estimation, when there is backlash in the transmission, the velocity will rise rapidly and then decrease, and may even reverse direction when the transmission engages. If this velocity "bounce" lasts for the entire first phase, then the estimation is not reliable (see the following figure). In any case it's necessary to consider the presence of this behaviour during the tuning (see Section 25.8, *Tuning problems*).

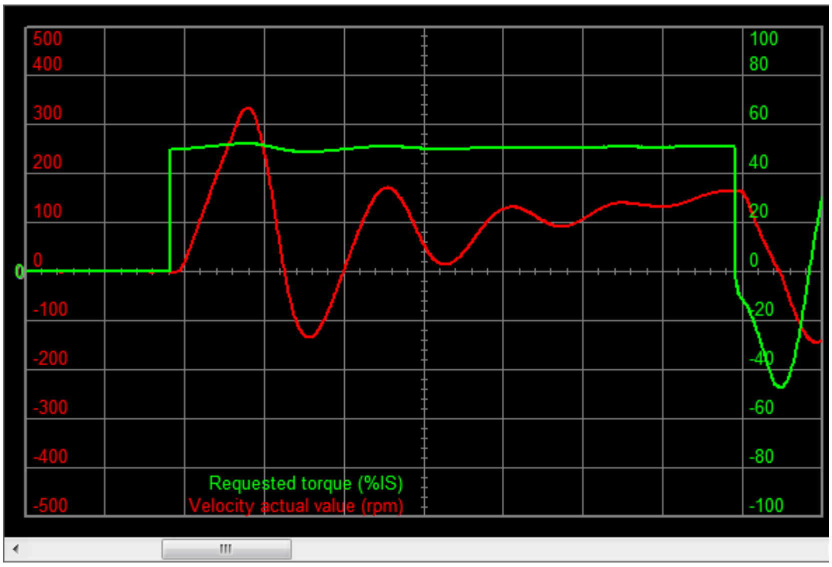



*Figure 20.11. Example of a not valid estimation due to backlashes in the transmission.*

**6. Verification of the tuning configuration, calculated by the inertia moment estimator**

The second phase of the estimation is a quick stop of the motor, that uses the loop parameters that have been recalculated according to the measured value of TotalEstimatedInertia. evaluating this phase, we can get the following conclusions:

- If the stop has a behaviour similar to the one that's reported in Figure 20.8, so without vibrations, without noise and with a little and immediatly damped overshoot, then the velocity loop tuning is finished without problems.
- Otherwise, if the stop presents a behavious similar to the one that's reported inFigure 20.9, so with little and immediatly damped vibrations, the system is however stable and it's better to improve its stability (see the points 3. System stabilisation and 4. Filters of the *Detailed Tuning Guide*).
- At last, if the stop presents a behavious similar to the one that's reported inFigure 20.10, so with wide and persistent vibrations, the system is unstable and it has to be stabilized (see the points 3. System stabilisation and 4. Filters of the *Detailed Tuning Guide*).

## Chapter 21

# Saving, restoring or cloning the drive configuration.

## 21.1. Drive configuration

By *configuring the drive* it is meant to configure all the parameters of the vocabulary. The drives of the series *i*BMD provide some commands to manage in an organic way (not only according to a single parameter) the current configuration. The configuration management of the drive is summarized in the following image.

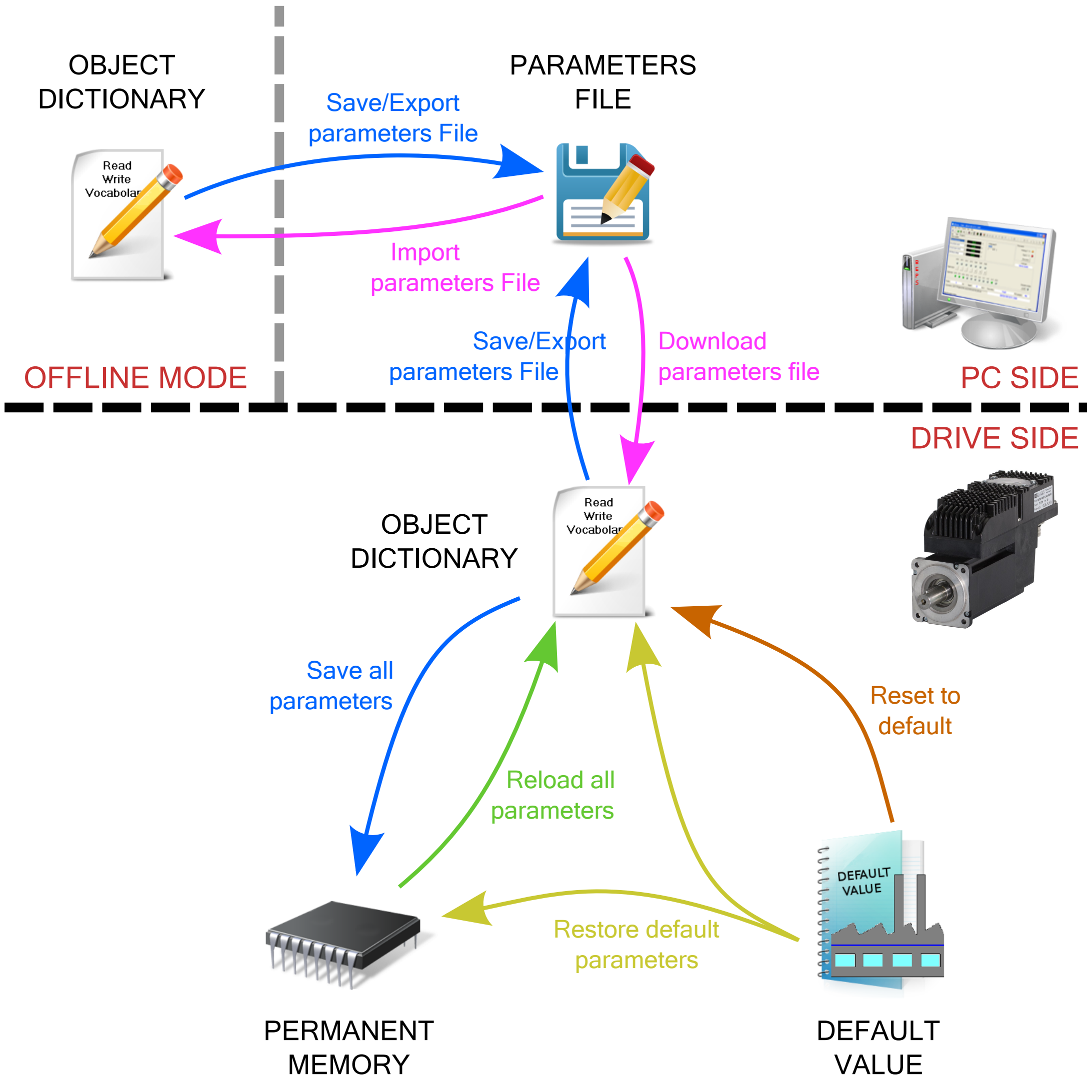



*Figure 21.1. Commands for the configuration management of the drive.*

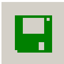
| Command | System Manager | Description |
|---|---|---|
| Save/Export parameters file | MotionDrive reserved | Saving the current configuration on parameter files |
| Import parameters file | MotionDrive reserved | Updating the current configuration with all the data in a parameter file (mode Offline) |
| Download parameters file | MotionDrive reserved | Updating the current configuration with all the data contained in a parameter file (mode Online) |
| Save all parameters | 2001 | Saving the current configuration in the drive permanent memory |
| Restore default parameters | 2200 | Updating the current configuration and the permanent memory with the default values |
| Reset to defaults | 2201 | Updating the current configuration with the default values |
| Reload all parameters | 2300 | Updating the current configuration with all the data in the permanent memory |
| Reload value of loops parameters and tuning configuration | 2301 | Updating the loops parameters and the TuningConfigurations with the values in the permanent memory |

| Command | System Manager | Description |
|---|---|---|
| Hard reset | 5000 | It's equivalent, for all parameters, to switching off and restarting the drive |
| Soft reset | 5001 | It's equivalent, for all parameters, except for the Non-reset-table parameters, to switching off and restarting the drive |

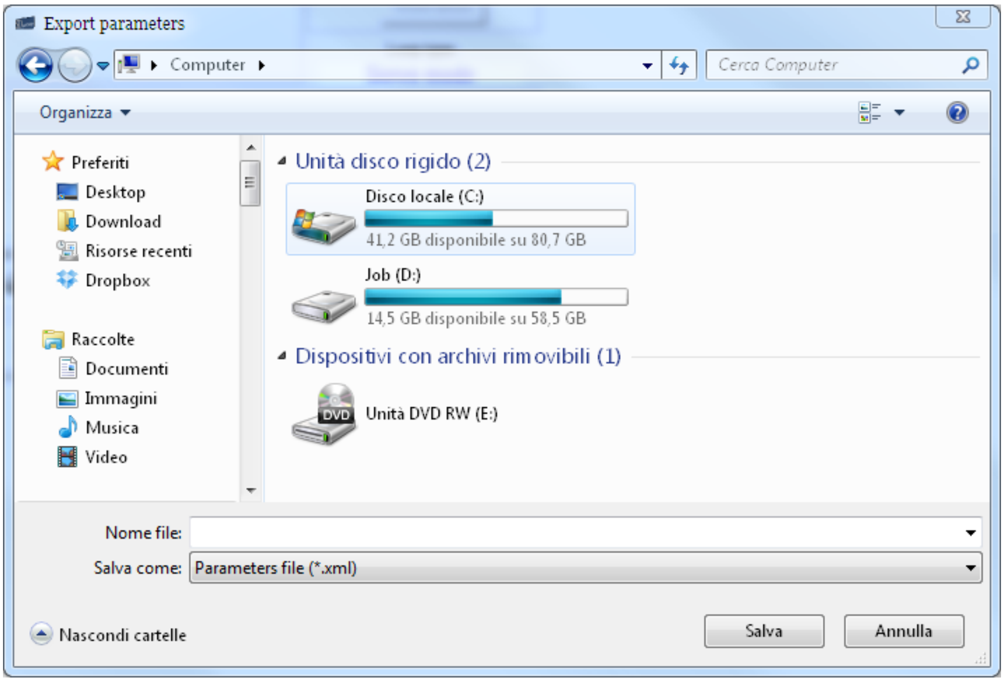
## 21.2. Saving/Exporting parameters file

To export the drive parameters on a file, in order to save them in an archive or download them in other drives, please follow this procedure. Access with MotionDrive:

**Main menu** > **Drive** > **Save/Export parameters...**

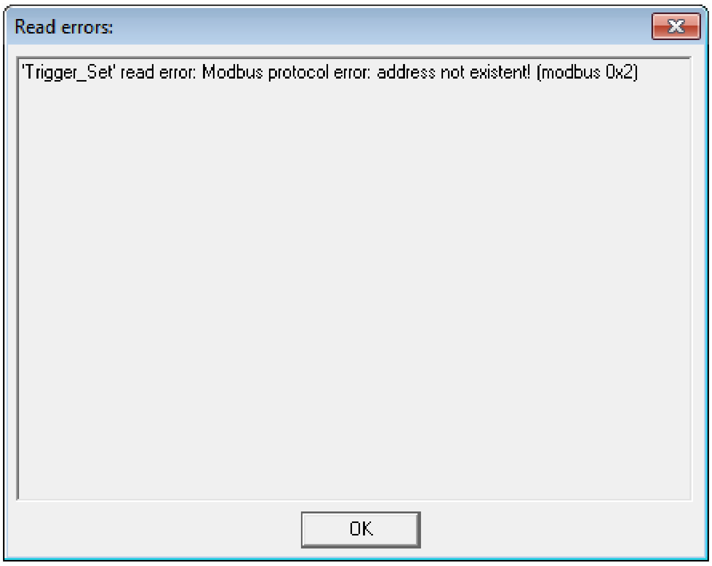
**Toolbar** > 

Choose the directory and the name of the destination file in the window Export parameters and press Save.




### Some problems that may occur while exporting the parameters

1. If there is an error in reading one or more parameters, the user is informed through a notice, as the one shown in the following window, that contains the list of the errors.



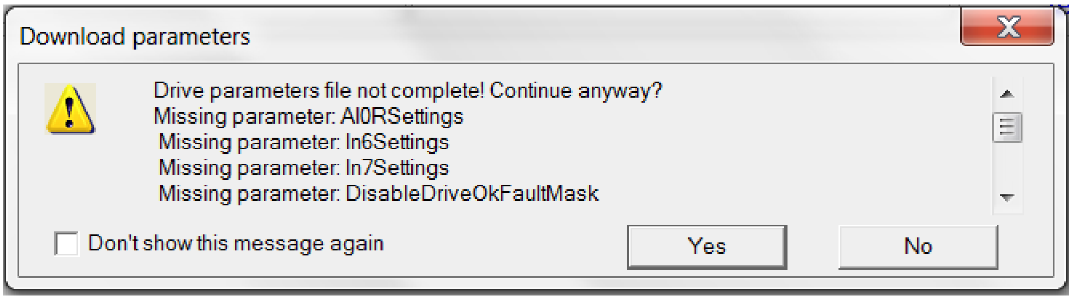

In this case it is better:

i. to update MotionDrive and the configuration files (Section 26.1, *MotionDrive updating*)

ii. to update the firmware with the latest available version (Section 26.3, *Firmware updating*)

iii. to interpret the error details (Section 8.4, *Errors in reading / writing of the parameters*)

2. If case of downloading in a drive a parameter file generated with a firmware that's older than the actually installed one, the user is informed through a notice, as the one shown in the following window, that contains the list of the missing parameters.




In this case it is sufficient to:

i. confirm and proceed with the downloading of the parameter file, even if it is incomplete

ii. save a new parameter file, that will be in this way generated compatible with the firmware that's actually installed and includes the parameters that have just been downloaded.

## 21.3. Downloading parameters file

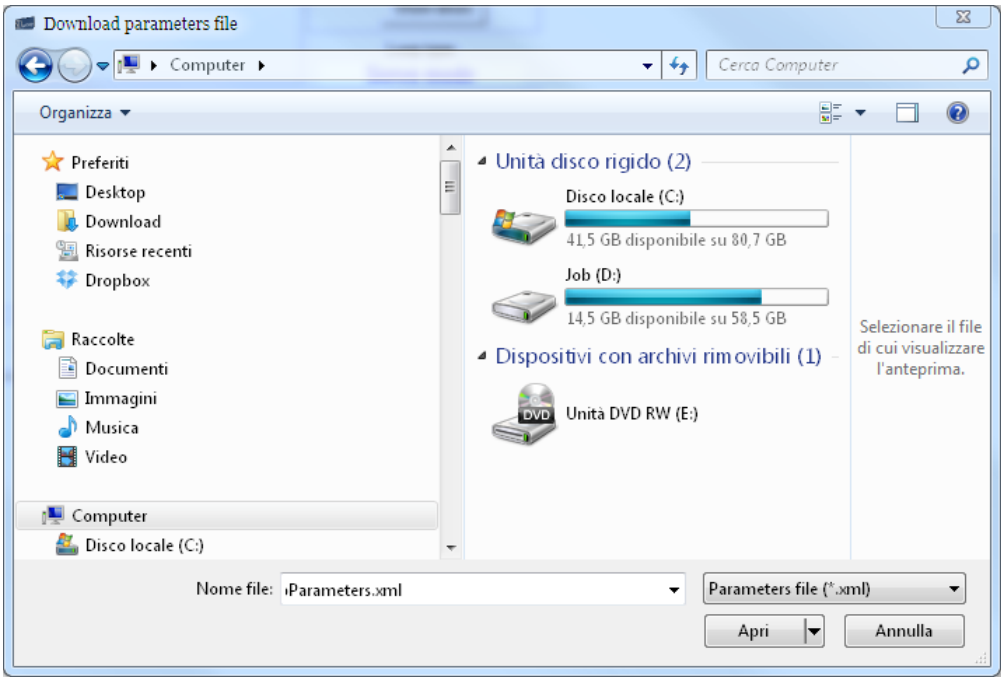
To update the drive parameters with a previously created parameters file, please use the following procedure. Access with MotionDrive:

**Main menu** > **Drive** > **Download parameters file...**

**Toolbar** > 

Choose in the window Download parameters file the parameters file to download in the drive and press 

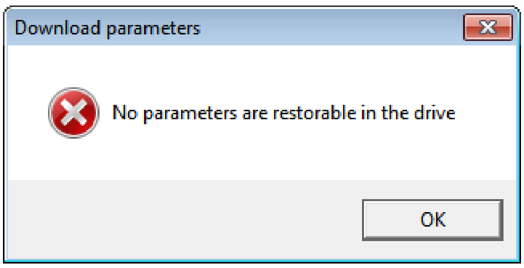
.




| NOTE |
|---|
| **The parameters downloaded in the drive are not automatically saved in the internal permanent memory.** |

## Some problems that may occur while downloading the parameters

1. If the following window appears, it means that this function is not available because of the firmware version that's installed in the drive. In this case it is necessary to update the firmware (Section 26.3, *Firmware updating*).



# 21.4. Importing parameters file

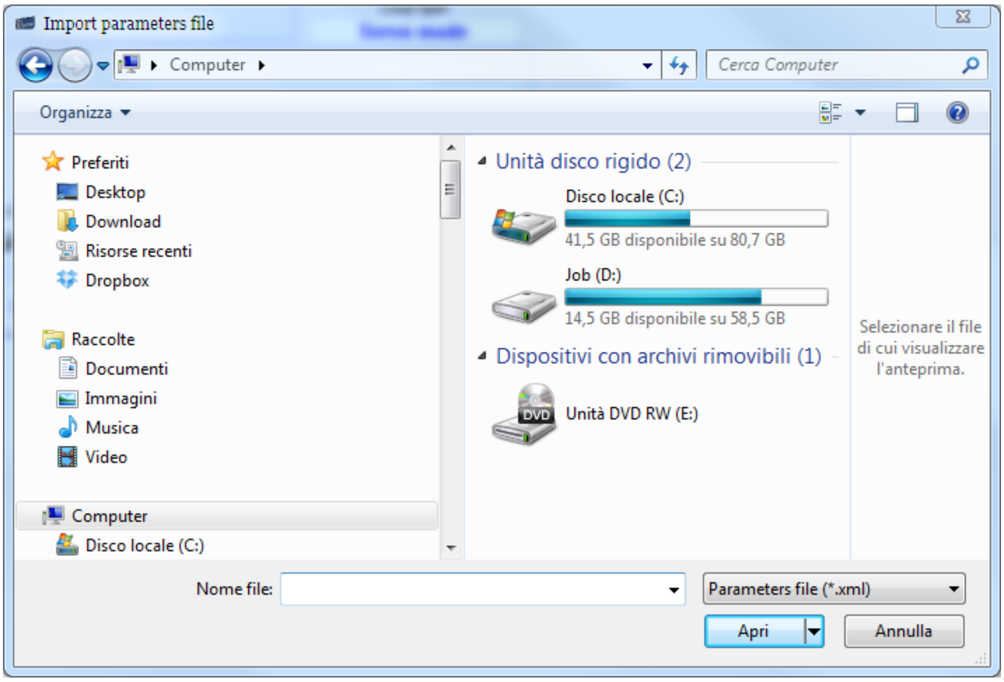
To change the parameters file during the Offline mode, please follow this procedure. Access with MotionDrive:

**Main menu** > **Drive** > **Import parameters file...**

**Toolbar** > 

Choose in the window Import parameters file, the directory and the file name to import and press 

.




NOTE

**Any changes in the imported parameters file are not automatically saved in the file. To save changes in the parameters file in Offline mode, use the Save/Export parameters file command.**

# 21.5. Compare parameters file

The MotionDrive environment provides an instrument to compare 2 parameters files. This method can be even used to verify the differences between the currently configured parameters and the default configuration. Access with MotionDrive:

**Main Menu** > **Drive** > **Compare parameters ...**

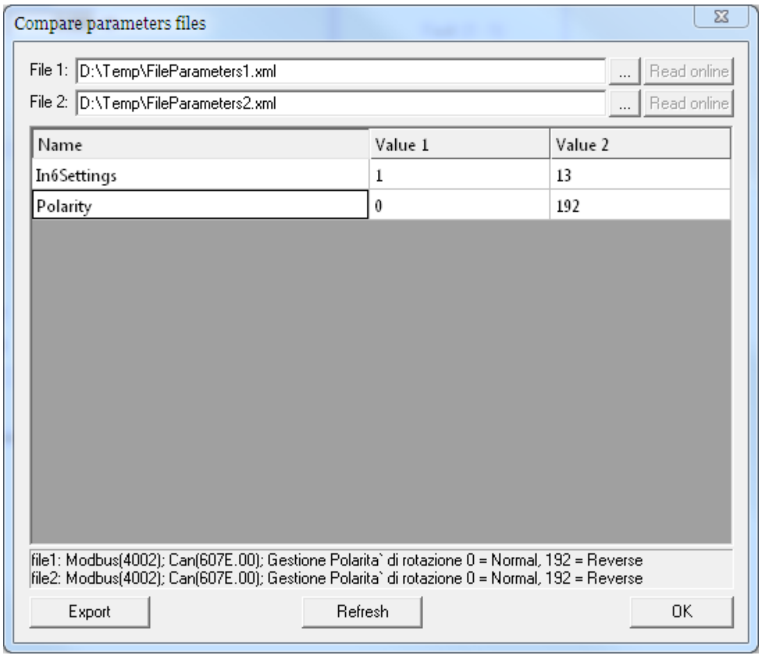
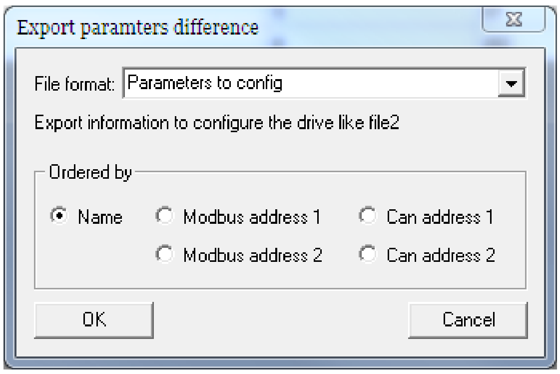



***Figure 21.2. Compare of 2 parameters***

Here follow the steps to execute the compare function:

1. Insert in the *File1* field the reference parameters file (.xml).
2. Insert in the *File2* field the parameters file (.xml) that you want to compare.
3. If a difference appears in the window, click on it to read its details in the text area on the bottom.
4. If you modify a file, click on the *Refresh* button to update the comparing.
5. To export the differences, click on the *Export* button and select the exportation mode:




   a. *Full Text Exportation:* to export in a ".txt" file the list of data of every parameter that results different or not present (in one of the two files) after the comparing.
   b. *Parameters to config:* in order to export in a ".txt" file the list of the data, referring to the file that's inserted in the *File2* field, that are necessary to write the parameters in the drive through a NON Bonfiglioli master: Modbus addresses CAN, Parameters name and Priority[1].

[1]The priority value is used to determine the writing order of the parameters. A parameter with a lower priority value must be written before, while a parameter with a higher priority value must be written after. The parameters with the same priority value can be written in any order.

IMPORTANT

**The files that have been obtained with the export operation ARE NOT PARAMETERS FILES (therefore they cannot be directly downloaded in the drive through MotionDrive), but only simple text files, that contain the list of the differences between the 2 parameters files. The purpose of those files is, as well as to list the differences, to provide the data that are necessary to write in the drive, through a NON Bonfiglioli master, the parameters that result different after the comparing.**

NOTE

**If you want to compare a parameter file with the actual drive parameterization, it's sufficient to load the first one and to use the "Read Online" function for the second one. This function creates a temporary file that will be used to compare operation and that will be automatically deleted on the Compare window closing action.**

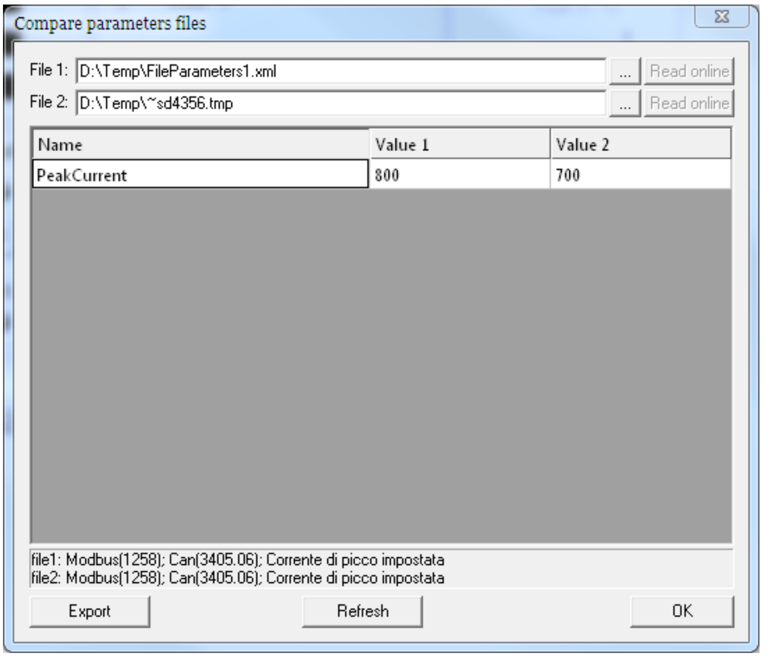



*Figure 21.3. Compare of a parameter file with the current drive parameters.*

# 21.6. Parameters cloning

For *cloning* it is meant the copy of only the configuration parameters from a drive to another one. It will be not executed any firmware update.

⚠ NOTICE

**The cloning procedure is guaranteed only if the two drives have the same firmware version. If the drives are not aligned, you have to update the drives with the desired firmware version.**

To clone the parameters of a *i*BMD series drive there are two procedures, depending on the instrument that you want to use:

**Parameters cloning with MotionDrive**

1. Connect to the drive that has to be cloned and export the parameter file (see Section 21.2, *Saving/Exporting parameters file*).
2. Connect to the drive that has to be configured and import the saved parameters file (see Section 21.3, *Downloading parameters file*).

**Cloning of the parameters with Master**

In this case, in order to clone the parameterization of a system in another one, the Compare procedure has to be used (see Section 21.5, *Compare parameters file*). In order to do this it's necessary:

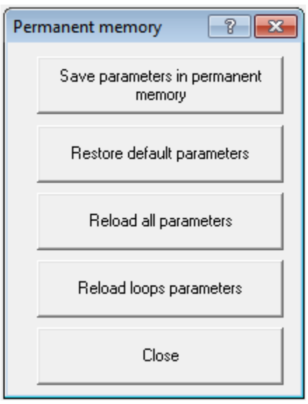
1. Save the actual parameters file from the drive to be cloned (see Section 21.2, *Saving/Exporting parameters file*).
2. Restore the default values in the drive to be cloned (see Section 21.7, *Permanent memory* or else with the command 2200 of the System Manager).
3. Save the parameter file with the default values and insert it in the *File 1* field (or use the "Read Online" function).
4. Load the parameter file, related to the point 1, in the *File 2* field.
5. Export in a text file the results of the comparing by using the option *Parameters to config*.
6. Restore the default parameters in the drive to be configured (see Section 21.7, *Permanent memory* or else with the command 2200 of the System Manager).
7. Use the data contained in the exported file to write in the Master the instructions to write the parameters that has to be modified.
8. Proceed with the parameters writing in the drive through Master.

## 21.7. Permanent memory

Management of the current configuration with the permanent memory. Access with MotionDrive:

**Main menu** > **Drive** > **Permanent memory...**

**Toolbar** > 



To save the current configuration in the drive permanent memory, press Save parameters in permanent memory (available also in Drive setup) or use the command 2001 of System Manager.

To update the current configuration and the permanent memory with the default values, press Restore default parameters or use the command 2200 of System Manager.

To update the current configuration with the data in the permanent memory, press Reload all parameters or use the command 2300 of System Manager.

To update only the loops configuration with the data in the permanent memory, press Reload loops parameters or use the command 2301 of System Manager.

IMPORTANT

**It is not necessary to reboot the drive for the new parameters resulted from Reload and Restore commands of the permanent memory to take effect.**

## 21.8. Resetting

The drives of the *i*BMD series provide different reset levels. Access with MotionDrive from **Main menu** > **Drive**.

| Command | System Manager | Reset-Cause | Description |
|---|---|---|---|
| Hard reset | 5000 | 2 | It's equivalent, for all parameters, to switching off and restarting the drive |
| Soft reset | 5001 | 6 | Except for the Non-resettable parameters and the Non-resettable on the reset parameters, it is the same as a turn off and on again of the drive |
| NMT reset | - | 5 | Except for the Non-resettable parameters and the Non-resettable on the reset parameters, it is the same as a turn off and on again of the drive. It can be run only in the CAN drive version and only with the NMT command (see Node control services). |
| Reset to defaults | 2201 | - | Updating the current configuration with the default values. The Reset-Cause does not change. |

### Non-resettable parameters

The *not resettable parameters* do not change when in the drive it is used a Soft reset, NMT reset or Reset to defaults command. They are:

- PositionActualValue.
- HomingStatus.

⚠ WARNING

**If the feedback sensor is incremental and if during the firmware reset the motor moves more than a half of revolution, the drive does NOT keep in memory these parameters.**

> ⚠ WARNING
>
> **If the feedback sensor is absolute and if during the firmware reset the encoder position is between its range limits, then these parameters are always kept in memory.**

> ⚠ WARNING
>
> **in case the FeedbackSensorAbsMode or FeedbackSensorCode parameters value is modified, these parameters are reset.**

### Non-resettable on the reset parameters

The *Non-resettable on the reset parameters* are not modified if the drive receives a Soft reset or a NMT reset command. They are:

- EncoderIncrements.
- Polarity.

> ⚠ WARNING
>
> **In case the sensor is incremental (or the FeedbackSensorAbsMode parameter make it work in this way), the modification of one of these parameters causes the reset of the HomingStatus parameter and of the executed homing procedure.**

### Enabled parameters after reset

The *enabled parameters after reset*, once written and saved in the drive permanent memory, take effect only after the Hard reset, Soft reset, NMT reset command or after the drive switching off and restarting. They are:

- CANopenPortSwitchSetup;
- CANopenPortSetupNodeID;
- CANopenPortSetupBaudRate;
- AuxiliaryPortSetupBaudRate;
- node number and baud rate set by dip switch (see Section 7.4, *Dip switches*) in the drive version CAN;
- Configured station alias in the drive version ETC.

## Chapter 22
# Creating a motion

Using the drives of the *i*BMD series, the motor motion can be commanded through:

- Master with  CANopen or EtherCAT port that supports the CoE protocol ("CANopen over EtherCAT)
- PLC integrated in the drive
- digital inputs and outputs + drive analog input

The drive makes it possible to carry out motions by controlling the torque, the speed and the motor position according to the operating mode set in the parameter ModesOfOperation. In the following chart you can find the features of the available operating modes. To learn how to command and control the motor motion, or how to enable, disable and stop the motor, or how to change the ModesOfOperation, please follow the instructions in the next paragraphs of this chapter.

| Type of motion | ModesOfOperationDisplay | Paragraph | Standard CiA-402 | Real-time | Digital I/O | Analog input | Enable automatic |
|---|---|---|---|---|---|---|---|
| **Position** | *Profile Position Mode* | Section 22.9, *Profile Position Mode* | *YES* | - | - | - | - |
| | *Interpolated Position Mode* | Section 22.10, *Interpolated Position Mode* | *YES* | *YES* | - | - | - |
| | *Cyclic Synchronous Position Mode* | Section 22.11, *Cyclic Synchronous Position Mode* | *YES* | *YES* | - | - | - |

| Type of motion | ModesOfOperationDisplay | Paragraph | Standard CiA-402 | Real-time | Digital I/O | Analog input | Enable automatic |
|---|---|---|---|---|---|---|---|
| Velocity | *Profile Velocity Mode (CiA402)* | Section 22.14, *Profile Velocity Mode (CiA402)* | *YES* | - | - | - | - |
| | *Profile Velocity Mode (CUSTOM)* | Section 22.15, *Profile Velocity Mode (CUSTOM)* | - | - | - | - | - |
| | *Profile Velocity AI Mode* | Section 22.16, *Profile Velocity AI Mode* | - | - | - | *YES* | *YES* |
| | *Cyclic Synchronous Velocity Mode* | Section 22.12, *Cyclic Synchronous Velocity Mode* | *YES* | *YES* | - | - | - |
| Torque | *Torque Mode* | Section 22.17, *Torque Mode* | *YES* | - | - | - | - |
| | *Torque AI Mode* | Section 22.18, *Torque AI Mode* | - | - | - | *YES* | *YES* |
| | *Cyclic Synchronous Torque Mode* | Section 22.13, *Cyclic Synchronous Torque Mode* | *YES* | *YES* | - | - | - |
| Other | *Homing Mode* | Section 22.19, *Homing Mode* | *YES* | - | - | - | - |

***Table 22.1. Features of the operating modes and modes of generation of the references.***

IMPORTANT

**To command the drive by using any Master, it's necessary to know and use the CiA402 State Machine, which characteristics are reported in Section 8.5, *CiA402 state machine*.**

## 22.1. Enabling by using the master

To enable the drive it's necessary to set the CiA402 State Machine to one of these 2 states:

- Operation enable
- *Switched On* (only if SwitchedOnOptionCode value is 1)

To enable the motor movement, set the CiA402 State Machine to the Operation enable state (see Section 8.5, *CiA402 state machine*). In the Operation enable state the changing of the operative mode is allowed both using the on-the-fly mode change (please see Section 22.5, *On-the-fly operative mode change*) and through the parameter ModesOfOperation.

Access with MotionDrive:

**Main menu** > **Drive** > **Drive enable**

**Toolbar** > 

IMPORTANT

**To enable the drive, the EnableInputStatus parameter must be equal to 1.**

## 22.2. Disabling by using the master

To disable the drive it is necessary to execute one of the following two operations:

- if the SwitchedOnOptionCode parameter value is 0, set the CiA402 State Machine to the *Switched On* state (see Section 8.5, *CiA402 state machine*)

- set the CiA402 State Machine to the *Switch On Disabled* state (see Section 8.5, *CiA402 state machine*). Through this operation it is possible to disable also the motor motion.

Access with MotionDrive:

> **Main menu** > **Drive** > **Drive disable**
>
> **Toolbar** >

To only disable the motor movement without disabling the drive is only possible if the SwitchedOnOptionCode parameter value is 1. In this case it's sufficient to set the CiA402 State Machine to the *Switched On* state (see Section 8.5, *CiA402 state machine*).

> ⚠ ATTENTION
>
> **If the drive is in Operation enable and the motor is in motion, the disabling operations causes the motor stop with maximum deceleration by setting RequestedSpeed to zero.**
>
> **Before starting a disabling operation, it is recommended to stop the motor motion by using the procedures described in Section 22.3, *Carrying out a stop by using the master*.**

## 22.3. Carrying out a stop by using the master

To stop the motor when the drive is in the Operation enable state and with any ModesOfOperation enabled, it is possible to run a stop command. The two stop commands implemented in the drive follow the specifications of CiA-402 and they are:

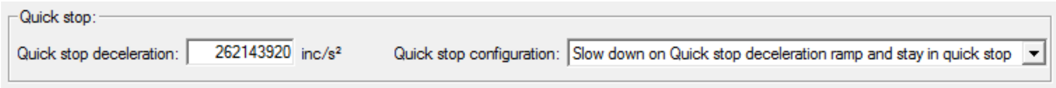
- *Halt*: stop command which carries out a braking ramp with a ProfileDeceleration deceleration. The stop ramp starts from the RequestedPosition and RequestedSpeed, without nullify PositionFollowingError and SpeedFollowingError;
- *Quick stop*: stop command which carries out a braking ramp parametrized according to the QuickStopConfiguration parameter. The braking ramp starts from the VelocityActualValue value, setting to zero SpeedFollowingError at the beginning of the ramp.

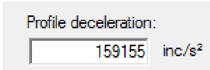
> ADVICE
>
> **Use the command Halt for a normal stop and the command Quick stop for an emergency braking.**

The stop execution can be commanded through the Controlword and the state of the CiA402 State Machine can be checked with the Statusword (see Section 8.5, *CiA402 state machine*).

Access with MotionDrive to parametrize the stops:

> **Main menu** > **Drive** > **Drive setup ...** > **Control mode**
>
> **Toolbar** > > **Control mode**



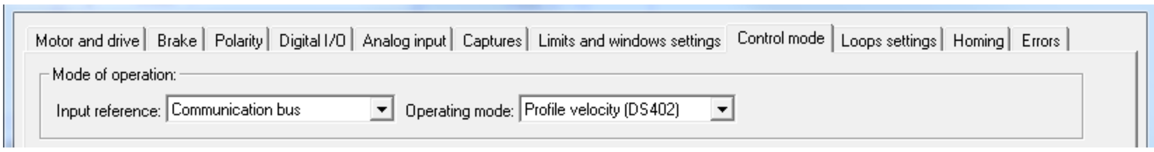




## 22.4. Change the operative mode with CiA-402 standard parameters

This type of operative mode change follows the CiA-402 specifications. To execute the operative mode change it's necessary to write the ModesOfOperation parameter; by reading the ModesOfOperationDisplay it's possible to check the active operative mode.

⚠ NOTICE

**In the Operation enable state it's possible to change ModesOfOperation only if the bit 4 of the Controlword is set to 0 (see Section 8.5, *CiA402 state machine*). The operative mode change, with this method, must be done with the motor stopped, and it's user's responsibility to be sure of this condition.**




In the frame *Mode of operation* choose the source of the reference in the pull-down menu *Input reference* and the operating mode in the pull-down menu *Operating mode*.

NOTE

**The change of ModesOfOperation is not allowed in the states Not Ready to Switch On, Quick Stop Active and Fault Reaction Active.**

## 22.5. On-the-fly operative mode change

NOTE

**The change ModesOfOperation in Operation enable is currently available only writing the parameters of additional bus Modbus.**

This kind of mode change is also called makes possible to move from any operative mode to another one without stopping the motor and keeping the drive in Operation enable. The operating modes allowing the change are:

- *Profile Position Mode*;
- *Homing Mode*;
- *Interpolated Position Mode*;
- *Profile Velocity Mode (CUSTOM)*.

The management of the on-the-fly mode change can be parametrized and commanded by using the following parameters.

| Parameter | Description |
|---|---|
| ApplyModeOperationCommand | Desired operating mode |
| ApplyModeOperationStatus | Status of the operating mode change |
| ApplyModeOperationParameters and following ones | Group of 7 parameters to set the mode change. The meaning of each of these parameters changes when ApplyModeOperationCommand changes, as shown in Table 22.2. |

| N. par. | *Profile Position Mode* | *Homing Mode* | *Interpolated Position Mode* | *Profile Velocity Mode (CUSTOM)* |
|---|---|---|---|---|
| 1 | TargetPosition | HomingMethod | IpPosFirstParameter | TargetVelocity |
| 2 | ProfileVelocity | SpeedForSwitch | IpPosSecondParameter | - |
| 3 | EndVelocity | SpeedForZero | - | EndVelocity |
| 4 | ProfileAccceleration | HomingAcceleration | IpPosSubModeSelect | ProfileAccceleration |
| 5 | ProfileDeceleration | IndexPulseDeadZone | - | ProfileDeceleration |
| 6 | EndIncrements | HomeOffset | - | - |
| 7 | StartVelocity | - | - | StartVelocity |

***Table 22.2. Meaning of the ApplyModeOperationParameters parameters.***

NOTE

**The writing of this group of parameters follows the same writing restrictions as the single parameters in their original addresses.**

To start a change of the on-the-fly operating mode the drive must be in Operation enable. Run this operations sequence:

1. Configure the change of the operating mode by properly setting ApplyModeOperationParameters.
2. Write the code of the new operating mode in ApplyModeOperationCommand.
3. Verify the result of the change by reading the parameter ApplyModeOperationStatus.

For some values of ApplyModeOperationCommand, there can be some peculiarities. Here follow them:

- ***Profile Position Mode***
  With the functionality on-the-fly mode change, the *Profile Position Mode* runs positions **absolute** in mode **Single set-point**. For details on the operating mode please see Section 22.9, *Profile Position Mode*.

## 22.6. How to control a motion in position

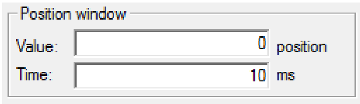
In the drives of the series *i*BMD some functionalities (common to all position modes) have been implemented and through these it is possible to check if the motion is run in conformity with the parameterization made by the user.

**Position**
To check the motor position, read the parameter PositionActualValue.

**Position reached target**
If in ModesOfOperationDisplay a position mode is set, to check if the motor reached its final position it is sufficient to check if the bit *Target reached* della Statusword is equal to 1. Such bit is set when the difference between PositionActualValue and the requested position is below PositionWindow (in absolute value) for a time at least equal to PositionWindowTime. The bit is reset when the difference gets over the window.

Access with MotionDrive:

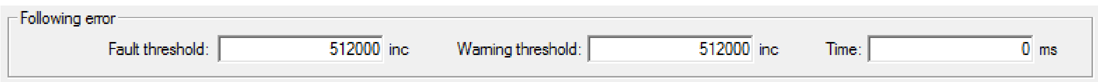
> **Main menu** > **Drive** > **Drive setup ...** > **Limits and windows setting**
>
> **Toolbar** > > **Limits and windows setting**




**Error of position tracking**
If in ModesOfOperationDisplay a position mode is set, it is possible to check the PositionFollowingError during the motor motion. Properly configuring the FollowingErrorWindow and FollowingErrorWindowWarn parameters it's possible to activate the Position following error (Fault and Warning respectively), if the PositionFollowingError exceeds the window for a time greater or equal to FollowingErrorTimeOut. Furthermore, in some operating modes, when PositionFollowingError exceeds the Fault threshold for a time at least equal to the time out, also the *Following error* bit of Statusword is set. The bit is reset when PositionFollowingError is lower, in absolute value, than the Fault window. For any further information on the error notice please see Chapter 24, *Fault and Warning*.

Access with MotionDrive:

> **Main menu** > **Drive** > **Drive setup ...** > **Limits and windows setting**
>
> **Toolbar** > > **Limits and windows setting**




## 22.7. Control of a speed motion

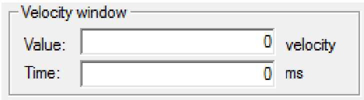
In the *i*BMD series drives have been implemented some functionalities (common to all the velocity modes) that permit to control if the movement is executed according to the user parametrization.

**Speed**
To check the motor speed you can use the only-reading-parameter VelocityActualValue.

**Speed target reached**
If in the ModesOfOperationDisplay a speed mode is set, it is sufficient to check if the bit *Target reached* of the Statusword is equal to 1 to check if the drive reached the final speed. This bit is set when the difference between the motor speed and the speed target is lower (in absolute value) to VelocityWindow for a time period at least equal to VelocityWindowTime. The bit is reset when the difference gets over the window.

Access with MotionDrive:

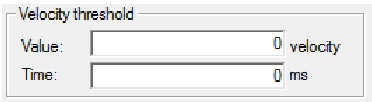
> **Main menu** > **Drive** > **Drive setup ...** > **Limits and windows setting**
>
> **Toolbar** > > **Limits and windows setting**




**Stopped motor**
If in the ModesOfOperationDisplay a speed mode is set, to check if the motor is stopped it is sufficient to check if the bit *Speed* of the Statusword is equal to 1. This bit is set when the motor speed is lower (in absolute speed) to VelocityThreshold for a period of time at least equal to VelocityThresholdTime. The bit is reset as soon as the difference is higher than the threshold.

Access with MotionDrive:

> **Main menu** > **Drive** > **Drive setup ...** > **Limits and windows setting**
>
> **Toolbar** > > **Limits and windows setting**




# 22.8. Control of a torque motion

In the drives of the series *i*BMD some functionalities (common to all torque modes) have been implemented and through these it is possible to check if the motion is run in conformity with the parameterization made by the user.

> ⚠ ATTENTION
>
> **The speed limit is not enabled with the torque modes. With an incorrectly high torque reference, the motor can reach a wrongly high speed.**

**Torque**
To check the torque created by the motor, read the ActualTorque parameter or the ActualFilteredTorque parameter.

**Torque target reached**
If in the ModesOfOperationDisplay only one torque mode is set, to check if the motor reached the requested torque it is sufficient to check that the bit *Target reached* of the Statusword is equal to 1. This bit is set when the difference between RequestedTorqueCurrent and ActualTorqueCurrent is lower (in absolute value) than the 5% of MotorStallCurrent for a time period of at least 1ms. The bit is reset when the difference gets over the window.

## 22.9. *Profile Position Mode*

The *Profile Position Mode* is used to run a motion in position, absolute or relative, where the positioning profile is created by the drive. This operating mode follows the specifications of the CiA-402.

To use this mode you need first of all to set the ModesOfOperation with the value 1 (*Profile Position Mode*), the MotionProfileType and the options that set the behaviour of the profiler with PositioningOptionCode. Finally you can proceed with the writing of the parameters defining how the position profile must be run:

- TargetPosition;
- EndIncrements;
- ProfileVelocity;
- StartVelocity;
- EndVelocity;
- ProfileAcceceleration;
- ProfileDeceleration.

⚠ NOTICE

**If the value of ProfileVelocity is lower than StartVelocity or EndVelocity, its value is internally set at the highest value of the two.**

Figure 22.1 You can find an example of profile and the meaning of the parameters defining it.

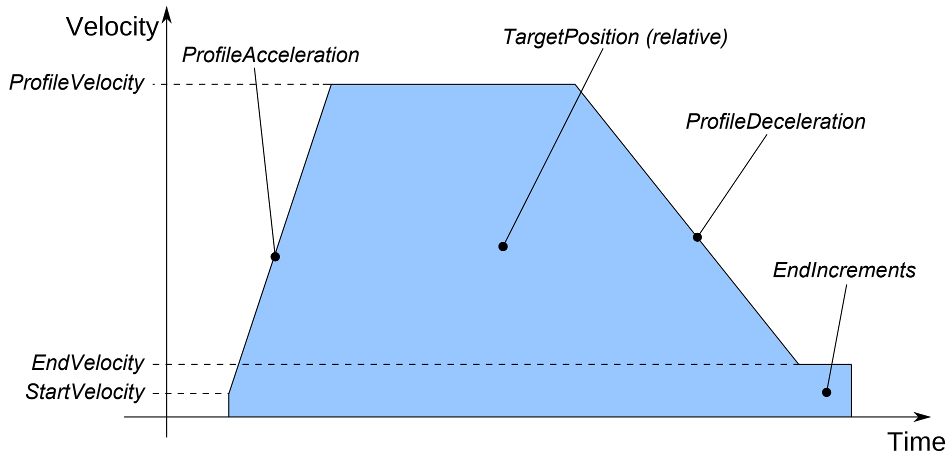



*Figure 22.1. Example of position profile with linear ramps.*

After having parametrized the drive and set it in the Operation enable state, you can run the commands to start the positionings and to check the state. To command a positioning you have to write the Controlword and read the Statusword following the procedures described in the CiA-402. Particularly, through this operating mode it is possible to run a positioning by commanding the following bits in the Controlword:

- bit *New set-point*: bit that, on the rising edge, it enables the application of the new position set-point, provided that it is allowed by the bit *Set-point acknowledge* of the Statusword;
- bit *Change set immediately*: through this bit it is possible to choose the mode of positioning between *Single set-point* (if the bit is set to 0) and *Set of set-point* (if the bit is set to 1). In the Set of set-point mode (see Figure 22.3) you can find only a buffer of data, the one for the data used during the positioning. In the Single set-point mode (see Figure 22.2) the positioning acts as described in the bit *Change immediately option* in the parameter PositioningOptionCode. The bit must be set with the transition of the bit *New set-point*;
- bit *Absolute / Relative*: through this bit it is possible to choose the mode to read the parameter TargetPosition: for absolute positionings if the bit is set to 0, for relative positionings if the bit is set to 1. The bit must be set with the transition of the bit *New set-point*.

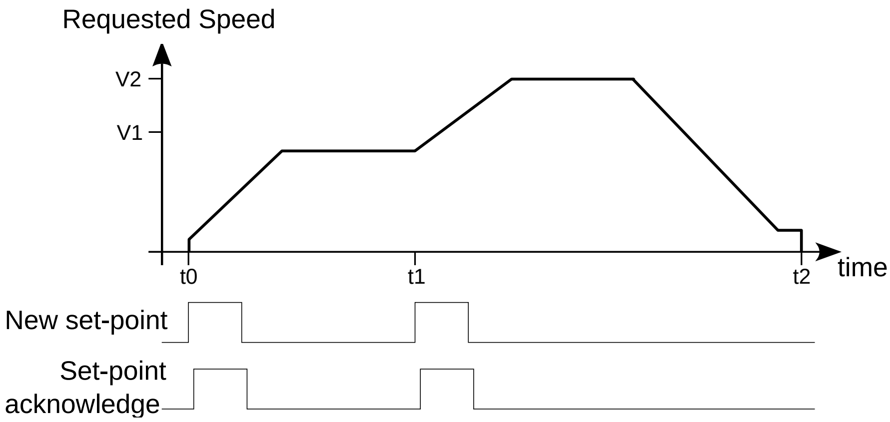



***Figure 22.2. Timing chart Profile Position Mode in Single set point mode.***

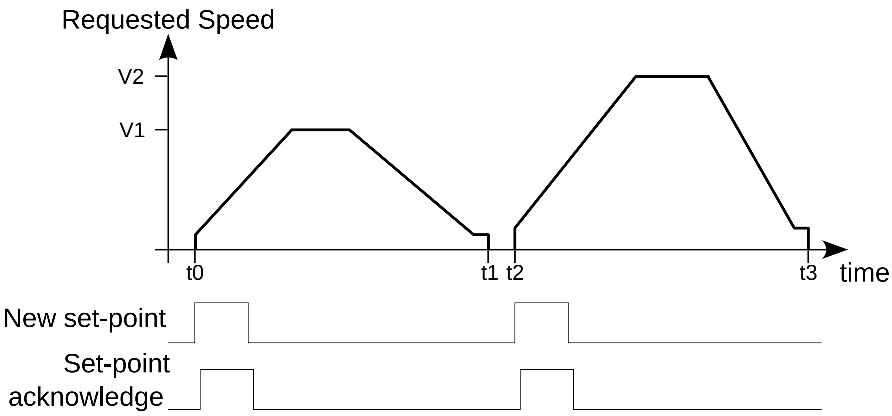



***Figure 22.3. Timing chart Profile Position Mode in Set of set points mode.***

NOTE

**After having reached the requested position at the end of the profile, the bit is set to Target reached of the Statusword (see Position reached target).**

In the Statusword there are three bits showing the status of the positioning:

- bit *Target reached* showing the status of Position reached target;
- bit *Set-point acknowledge* showing if a new set point of positioning can be accepted (bit equal to 0) or not (bit equal to 1);
- bit *Following error* showing the status of Error of position tracking.

NOTE

**If a new positioning is started in the Single set-point mode, the ongoing one is aborted and the new one is started without motor stopping.**

## 22.10. *Interpolated Position Mode*

IMPORTANT

**To command the drive by this operating mode it is necessary to have a Master supporting at least a Real-time protocol on CANopen or EtherCAT bus.**

The *Interpolated Position Mode* is an operative mode that allows to control the motor in Real-time by using a CANopen or EtherCAT master. This operative mode respects the CiA-402 specifications.

To work, this mode requires the cyclic sending from the master within a defined time (which will be later called $T_{SYNC}$, synchronization time) of the following parameters (the synchronization techniques are described in the chapter of the communication interfaces):

- IpPosFirstParameter: position reached when the $T_{SYNC}$ expires. This datum is necessary in all kinds of interpolation implemented in the drive.
- IpPosSecondParameter: speed reached when the $T_{SYNC}$ expires. This datum is not used in the linear interpolation.

The writing of the IpPosFirstParameter and IpPosSecondParameter parameters does not use the SDOs, but the PDOs combined with some techniques used for the synchronization with the other nodes linked to the bus. In Figure 22.4 an example of linear interpolation with the Hard sync synchronization technique is reported. This technique is used in the EtherCAT field bus through the Sync Signal synchronization signal (SYNC).

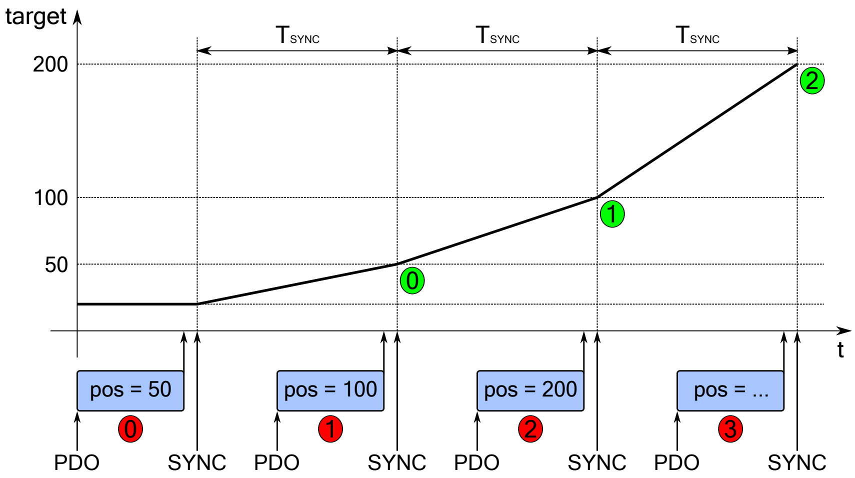



*Figure 22.4. Linear interpolation with synchronization made through Sync Signal synchronization signal (SYNC)*

⚠ NOTICE

**The parameters are so contained in the PDO RX (see the communication interface related chapters) and determine the construction of the movement profile.**

**To avoid this drawback, Bonfiglioli has implemented in the drives of the *i*BMD series a monitoring and management functionality for the corrupted or missing PDO RX (see Section 10.3.2, *Missing or corrupted PDO RX management*).**

The kind of interpolation can be set through the IpPosSubModeSelect parameter and the following methods are available:

- **Linear interpolation**
  The drive runs the interpolation of the position only, by linking with a straight line the set-point of the previous position, reached at the beginning of the new period of $T_{SYNC}$, with the position set point sent by the master in the IpPosFirstParameter parameter. The FeedForwardSpeed is calculated by the drive and is constant during the whole $T_{SYNC}$ period. In Figure 22.5 you can find an example of linear interpolation.

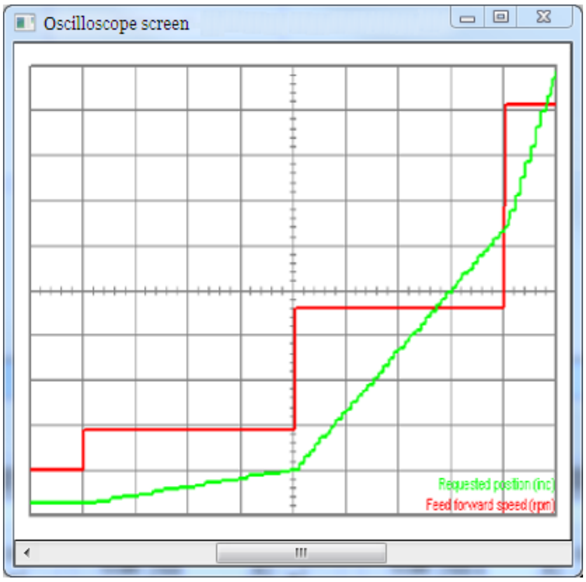



*Figure 22.5. Example of linear interpolation*

- **Linear interpolation with FeedForwardSpeed**
  The drive runs the interpolation only of the position by linking with a straight line the set-point of a previous position, with the position set point sent by the master in the IpPosFirstParameter parameter. The necessary FeedForwardSpeed to command the control loops is obtained from the set-point of the speed sent by the master in the IpPosSecondParameter parameter and is constant during the whole $T_{SYNC}$ period. This kind of interpolation allows a better motion fluidity compared to the simple linear interpolation.
- **Cubic interpolation**
  The drive runs the interpolation of both the position and the velocity by linking with segments of curve line, cubic for the position and quadratic for the velocity, the initial values (IpPosFirstParameter and VelocityOffset values received by the master with the previous $T_{SYNC}$ period) with the end values (IpPosFirstParameter and VelocityOffset values received by the master through IpPosFirstParameter and IpPosSecondParameter). The movement fluidity of this kind of interpolation respect to the other ones is better, as you can observe by comparing Figure 22.6 with Figure 22.5. In fact considering that these two pictures have been created by using the same parameters (except of course the interpolation type selector IpPosSubModeSelect) and with a $T_{SYNC}$ of medium duration, it is evident how the Figure 22.6 curves have a trend with less abrupt movements.

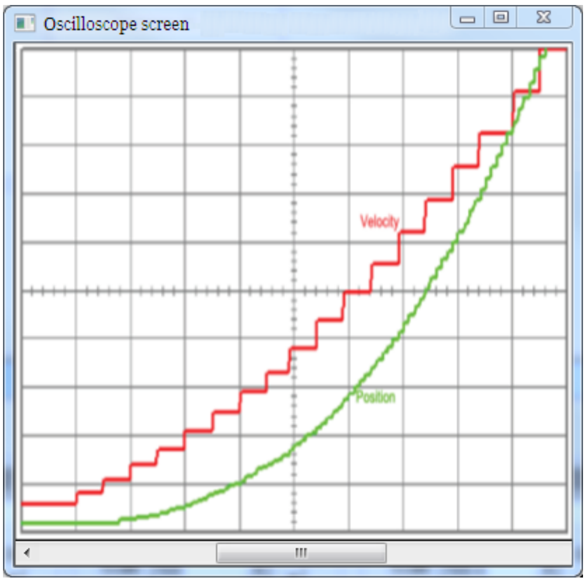



*Figure 22.6. Example of cubic interpolation*

⚠ NOTICE

**In the linear interpolation with or without feed-forward (values 0 and -10 in the IpPosSubModeSelect parameter) the set $T_{SYNC}$ period must be greater than MotionLoopPeriod.**

**In the cubic interpolation (value -1 in the IpPosSubModeSelect parameter) the set $T_{SYNC}$ period must be 4 times greater than MotionLoopPeriod parameter.**

ADVICE

**The cubic interpolation use is advantageous only if the $T_{SYNC}$ time has a middle-long duration (about over 4 ms) while, for interpolations that have short $T_{SYNC}$ times (about up to 4ms), these advantages are not, so it's better to use the linear interpolation.**

To command the drive with the interpolated mode it is necessary:

1. to set the ModesOfOperation with the value 7 (*Interpolated Position Mode*);
2. to configure the communication parameters of the bus field (PDO configuration and mapping, setting of the synchronization system, ...):
3. to set the IpPosSubModeSelect;
4. to enable in the master the management that allows, at regular intervals, the sending of the set-points via PDO in the IpPosDataRecord and the synchronism management;
5. to take the drive in the Operation enable state;
6. enable the position interpolator by setting the *Enable ip mode* bit (see Table 8.14 of the Controlword and check that the *Ip mode active* bit is set (see Table 8.12 of the Statusword;
7. at this point it is possible to command the drive.

⚠ ATTENTION

**If the Enable ip mode bit of the Controlword is reset, the motion is stopped and the motor stops with maximum deceleration by resetting RequestedSpeed to zero.**

## 22.11. *Cyclic Synchronous Position Mode*

IMPORTANT

**To command the drive by this operating mode it is necessary to have a Master supporting at least a Real-time protocol on CANopen or EtherCAT bus.**

The *Cyclic Synchronous Position Mode* is an operative mode that allows to control the motor in Real-time by using a CANopen or EtherCAT master. This operative mode respects the CiA-402 specifications.

To work, this mode requires to the master the cyclic sending, within a defined time (which will be later called $T_{SYNC}$, synchronization time) of the TargetPosition parameter (the synchronization techniques are described in the chapter of the communication interfaces):

In case of cubic interpolated mode use, even the VelocityOffset parameter will be necessary.

There are other parameters that are not required by the drive to generate the movement (they are not necessary), but can be useful to improve it. These parameters are:

- PositionOffset: position that will be added to the TargetPosition.
- VelocityOffset:
  - in case of cubic interpolation, this parameter is necessary because it is the velocity that the drive needs to make the interpolation calculations.
  - in case of non cubic interpolation: if the CyclicSynchronousSubMode indicates that the KVff internal calculation is disabled, it will be used as KVff
  - In all the other cases it is not used
- TorqueOffset: it is used as KAff if the CyclicSynchronousSubMode parameter indicates that the KAff internal calculation is disabled, otherwise it's not used.

The writing of the TargetPosition parameter does not use the SDO, but the PDO combined with some techniques that allow the synchronization with the other nodes that are connected to the bus. In Figure 22.4 it is reported an example of linear interpolation with the Hard sync synchronization technique, used in the EtherCAT field bus with the Sync Signal synchronization signal (SYNC).

> ⚠ NOTICE
>
> **The parameters are so contained in the PDO RX (see the communication interface related chapters) and determine the construction of the movement profile. When a CANopen Master is used, if a PDO RX is lost it's automatically re-sent, while in case of an EtherCAT Master this functionality is not provided.**
>
> **To avoid this drawback, Bonfiglioli has implemented in the drives of the *i*BMD series a monitoring and management functionality for the corrupted or missing PDO RX (see Section 10.3.2, *Missing or corrupted PDO RX management*).**

The kind of interpolation can be set through the CyclicSynchronousSubMode parameter and the following methods are available:

- **No interpolation**
  The drive executes the movement without interpolating the position target, that will be directly applied on the $T_{SYNC}$ signal. The FeedForwardSpeed and the FeedForwardAcceleration can be set by the master or internally calculated by the drive (see Table 27.16 for the available combinations).
- **Linear interpolation**
  The drive runs the interpolation only of the position by linking with a straight line the set-point of a previous position, with the position set point sent by the master in the TargetPosition parameter. The FeedForwardSpeed and the FeedForwardAcceleration can be set by the master or internally calculated by the drive (see Table 27.16). In Figure 22.5 an example of linear interpolation can be found.
- **Cubic interpolation**
  The drive executes the interpolation both of the position and of the velocity, linking with a cuved lines, cubic for the position and quadratics for the velocity, the initial values (values of TargetPosition and VelocityOffset received from the master with the previous $T_{SYNC}$ period) with the final ones (values of TargetPosition and VelocityOffset received by the master). This interpolation type, as you can see comparing Figure 22.6 with Figure 22.5, allows a movement improved fluidity respect to all the other interpolation modes. In fact, bearing in mind that the two figures have been created by using the same parameters (except for the CyclicSynchronousSubMode interpolation type selector, of course) and by using a $T_{SYNC}$ with medium duration, it's evident how the curves in Figure 22.6 have a trend with less abrupt deviations. The FeedForwardAcceleration can be set by the master or internally calculated by the drive (see Table 27.16).

> ⚠ NOTICE
>
> **In case of no interpolation or linear interpolation (different values from -147 and -148 in the CyclicSynchronousSubMode parameter) with or without feed-forward, the set $T_{SYNC}$ period must be greater than MotionLoopPeriod.**
>
> **In the cubic interpolation (values -147 and -148 in the CyclicSynchronousSubMode parameter) the set $T_{SYNC}$ period must be greater than 4 times the MotionLoopPeriod parameter.**

> ADVICE

**The cubic interpolation use is advantageous only if the $T_{SYNC}$ time has a middle-long duration (about over 4 ms) while, for interpolations that have short $T_{SYNC}$ times (about up to 4ms), these advantages are not, so it's better to use the linear interpolation.**

To command the drive with this mode it is necessary:

1. to configure the communication parameters of the bus field (PDO configuration and mapping, setting of the synchronization system, ...):
2. to activate in the master the management that allows, at regular intervals, to send the set-points through PDO and the synchronism management; the number and the type of the data (set-point) that have to be sent, depends on the CyclicSynchronousSubMode that has been set;
3. to set the CyclicSynchronousSubMode;
4. to set the ModesOfOperation with the value 8 (*Cyclic Synchronous Position Mode*);
5. to take the drive in the Operation enable state;
6. at this point it is possible to command the drive.

ATTENTION

**By selecting a value of CyclicSynchronousSubMode that assigns to the master the KVff and KAff calculation, it will be obtained a smoother motor motion. Pay attention that is a master duty to ensure that these data are sent, because the drive does not verify their receiving. If this last configuration is selected and the master doesn't anyway send the necessary feed forward values, the motion profile may be not the desired one.**

## 22.12. *Cyclic Synchronous Velocity Mode*

IMPORTANT

**To command the drive by this operating mode it is necessary to have a Master supporting at least a Real-time protocol on CANopen or EtherCAT bus.**

The *Cyclic Synchronous Velocity Mode* is an operative mode that allows the motor control in Real-time by using a CANopen or EtherCAT master. This operative mode respects the CiA-402 specifications.

To work, this mode requires to the master the cyclic sending, within a defined time (which will be later called $T_{SYNC}$, synchronization time) of the TargetVelocity parameter (the synchronization techniques are described in the chapter of the communication interfaces):

There are other parameters that are not required by the drive to generate the movement (they are not necessary), but can be useful to improve it. These parameters are:

- VelocityOffset: velocity that will be added to TargetVelocity.
- TorqueOffset: it is used as KAff if the CyclicSynchronousSubMode parameter indicates that the KAff internal calculation is disabled, otherwise it's not used.

The writing of the TargetVelocity parameter does not use the SDO, but the PDO combined with some techniques that allow the synchronization with the other nodes that are connected to the bus. In Figure 22.4 it is reported an example of linear interpolation with the Hard sync synchronization technique, used in the EtherCAT field bus with the Sync Signal synchronization signal (SYNC).

⚠ NOTICE

**The parameters are so contained in the PDO RX (see the communication interface related chapters) and determine the construction of the movement profile. When a CANopen Master is used, if a PDO RX is lost it's automatically re-sent, while in case of an EtherCAT Master this functionality is not provided.**

**To avoid this drawback, Bonfiglioli has implemented in the drives of the *i*BMD series a monitoring and management functionality for the corrupted or missing PDO RX (see Section 10.3.2, *Missing or corrupted PDO RX management*).**

The kind of interpolation can be set through the CyclicSynchronousSubMode parameter and the following methods are available:

- **No interpolation**
  The drive executes the movement without interpolate the velocity target, that will be directly applied on the $T_{SYNC}$ signal arrival. The FeedForwardAcceleration can be set by the master or internally calculated by the drive (see Table 27.16 for the available combinations).
- **Linear interpolation**
  The drive executes the interpolation of the velocity only, by linking with a straight line the previous velocity set-point with the set-point that has been sent to the master in the TargetVelocity parameter. The FeedForwardAcceleration can be set by the master or internally calculated by the drive (see Table 27.16). In Figure 22.5 you can find an example of linear interpolation.

⚠ NOTICE

**The $T_{SYNC}$ period that has been set must be greater than MotionLoopPeriod.**

To command the drive with this mode it is necessary:

1. to configure the communication parameters of the bus field (PDO configuration and mapping, setting of the synchronization system, ...):
2. to activate in the master the management that allows, at regular intervals, to send the set-points through PDO and the synchronism management; the number and the type of the data (set-point) that have to be sent, depends on the CyclicSynchronousSubMode that has been set;
3. to set the CyclicSynchronousSubMode;
4. to set the ModesOfOperation with the value 9 (*Cyclic Synchronous Velocity Mode*);
5. to take the drive in the Operation enable state;
6. at this point it is possible to command the drive.

⚠ ATTENTION

**By selecting a value of CyclicSynchronousSubMode that assigns to the master the KAff calculation, it will be obtained a smoother motor motion. Pay attention that is a master duty to ensure that these data are sent, because the drive does not verify their receiving. If this last configuration is selected and the master doesn't anyway send the necessary feed forward values, the motion profile may be not the desired one.**

# 22.13. *Cyclic Synchronous Torque Mode*

IMPORTANT

**To command the drive by this operating mode it is necessary to have a Master supporting at least a Real-time protocol on CANopen or EtherCAT bus.**

The *Cyclic Synchronous Torque Mode* is an operative mode that allows the motor control in Real-time by using a CANopen or EtherCAT master. This operative mode respects the CiA-402 specifications.

To work, this mode requires to the master the cyclic sending, within a defined time (which will be later called $T_{SYNC}$, synchronization time) of the TargetTorque parameter (the synchronization techniques are described in the chapter of the communication interfaces):

There are other parameters that are not required by the drive to generate the movement (they are not necessary), but can be useful to improve it. These parameters are:

- TorqueOffset: torque that will be added to the TargetTorque.

The writing of the TargetTorque parameter does not use the SDO, but the PDO combined with some techniques that allow the synchronization with the other nodes that are connected to the bus. In Figure 22.4 it is reported an example of linear interpolation with the Hard sync synchronization technique, used in the EtherCAT field bus with the Sync Signal synchronization signal (SYNC).

⚠ NOTICE

**The parameters are so contained in the PDO RX (see the communication interface related chapters) and determine the construction of the movement profile. When a CANopen Master is used, if a PDO RX is lost it's automatically re-sent, while in case of an EtherCAT Master this functionality is not provided.**

**To avoid this drawback, Bonfiglioli has implemented in the drives of the *i*BMD series a monitoring and management functionality for the corrupted or missing PDO RX (see Section 10.3.2, *Missing or corrupted PDO RX management*).**

The kind of interpolation can be set through the CyclicSynchronousSubMode parameter and the following methods are available:

- **No interpolation**
  The drive executes the movement without interpolating the torque target, that will be directly applied on the $T_{SYNC}$ signal.
- **Linear interpolation**
  The drive runs the interpolation only of the torque by linking with a straight line the set-point of a previous position, reached at the beginning of the new period of $T_{SYNC}$, with the torque set point sent to the master in the TargetTorque parameter. In Figure 22.5 you can find an example of linear interpolation.

⚠ NOTICE

**The $T_{SYNC}$ period that has been set must be greater than MotionLoopPeriod.**

To command the drive with the interpolated mode it is necessary:

1. to configure the communication parameters of the bus field (PDO configuration and mapping, setting of the synchronization system, ...):

2. to activate in the master the management that allows, at regular intervals, to send the set-points through PDO and the synchronism management; the number and the type of the data (set-point) that have to be sent, depends on the CyclicSynchronousSubMode that has been set;
3. to set the CyclicSynchronousSubMode;
4. to set the ModesOfOperation with the value 10 (*Cyclic Synchronous Torque Mode*);
5. to take the drive in the Operation enable state;
6. at this point it is possible to command the drive.

## 22.14. *Profile Velocity Mode (CiA402)*

The *Profile Velocity Mode (CiA402)* is used to carry out a speed motion, in which the speed profile is created by the drive. This operating mode follows the specifications of the CiA-402.

To use this mode you need to set the ModesOfOperation with the value 3 (*Profile Velocity Mode (CiA402)*). Then you can proceed to write the parameters defining how to run the speed profile:

- MotionProfileType;
- StartVelocity;
- EndVelocity;
- ProfileAccceleration;
- ProfileDeceleration;
- TargetVelocity.

In Figure 22.7 you can find an example showing how it is possible to change the TargetVelocity and the other profile parameters in any moment. In the first phase the motor is accelerated until it reaches the V1 speed; later it is accelerated again to reach the V2 speed but it does not reach it because it gets a new request for reaching the V3 speed. At last the motor slows down until it stops since the V4 speed is equal to 0. You can notice that the acceleration and deceleration ramps are broken, near the value of zero, respectively from StartVelocity and from EndVelocity.

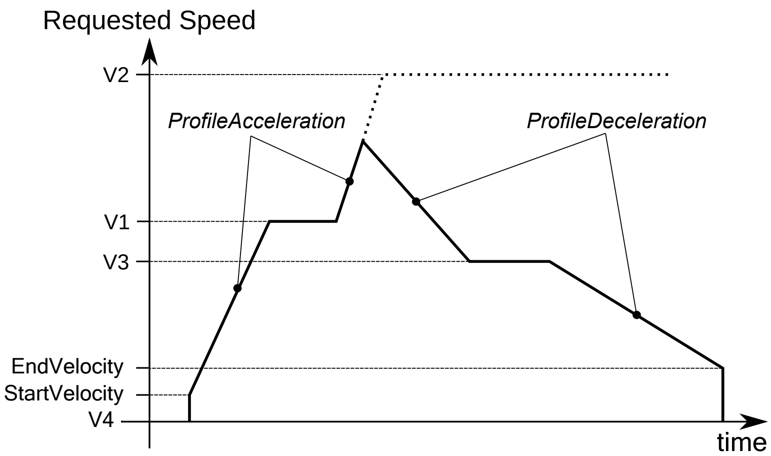



***Figure 22.7. Speed profile with linear ramps.***

After the drive is parametrized and set to the Operation enable mode, the motor will start moving as soon as a TargetVelocity is written with an absolute value higher than EndVelocity and StartVelocity.

In the Statusword there are two bits showing the motion status:

- bit *Target reached*: bit showing the status of the Speed target reached;
- bit *Speed*: bit showing if the drive has the Stopped motor.

> ⚠ NOTICE
>
> **If with the motor in motion in mode Profile Velocity Mode (CiA402) the TargetVelocity is written in an absolute value which is lower than EndVelocity or StartVelocity, the motor slows down with a deceleration equal to ProfileDeceleration until it reaches the EndVelocity and then it stops.**

> ⚠ ATTENTION
>
> **If with the motor in motion in Profile Velocity Mode (CiA402) mode, the EndVelocity or the StartVelocity are written in an absolute value which is higher than TargetVelocity, the motor stops with maximum deceleration resetting RequestedSpeed to zero.**

## 22.15. *Profile Velocity Mode (CUSTOM)*

The *Profile Velocity Mode (CUSTOM)* is used to run a motion in speed whose position is controlled, in which the speed profile is created from the drive. This operating mode works like the *Profile Velocity Mode (CiA402)* with the only difference that the position control is enabled. In Figure 22.8 you can notice a difference in the behaviour of the motor speed between the two operating modes, when a brake torque is applied at the instant $t_1$.

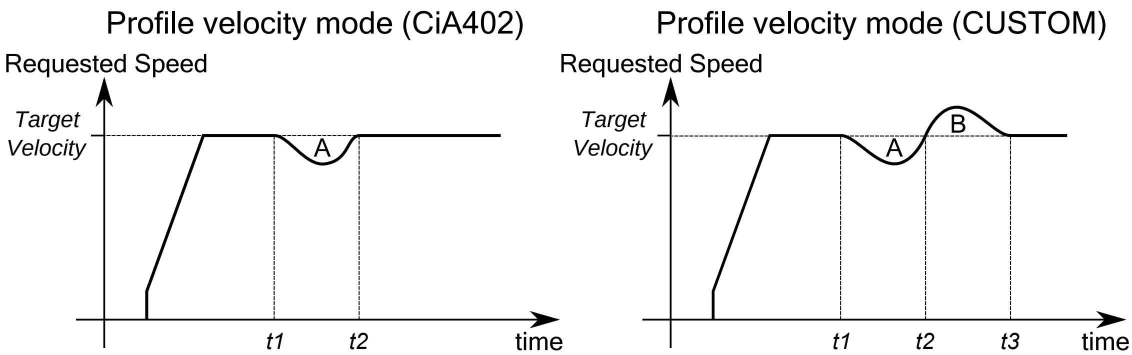



***Figure 22.8. Comparison between a motion in Profile Velocity Mode (CiA402) and in Profile Velocity Mode (CUSTOM).***

In the previous picture you can notice that starting from the instant t2, the two operating modes behave in a different way:

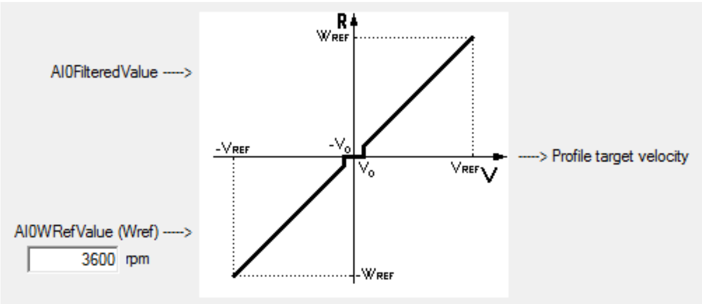
- ***Profile Velocity Mode (CiA402)***
  The drive compensates for the brake torque and returns to the speed TargetVelocity.
- ***Profile Velocity Mode (CUSTOM)***
  The drive returns to the speed TargetVelocity after regaining the lost position. This means that a speed overshoot is created in the time interval t2-t3, so that the A area is equal to the B area (A area = lost position = regained position = B area). At the instant t3, when the lost position is fully regained, the drive returns to the speed TargetVelocity.

> NOTE
>
> **With the Profile Velocity Mode (CUSTOM) the control of the Error of position tracking is enabled.**

To use this operating mode it is sufficient to write -113 in ModesOfOperation and, so, to follow the instructions described in Section 22.14, *Profile Velocity Mode (CiA402)*.

## 22.16. *Profile Velocity AI Mode*

The *Profile Velocity AI Mode* is used to run a motion in speed, in which the speed profile is created from the drive as it happens for the *Profile Velocity Mode (CiA402)*, but in this case the requested speed is not set through the TargetVelocity parameter but it is obtained from AI0FilteredVoltage. The conversion of the values from their voltage to the requested speed occurs according to what is reported in Section 17.3, *Conversion*.




IMPORTANT

**For the Profile Velocity AI Mode you can apply all the considerations related to the Profile Velocity Mode (CiA402) and its parameters, as explained in Section 22.14, *Profile Velocity Mode (CiA402)*, except from what was stated for the TargetVelocity parameter.**

To enable the *Profile Velocity AI Mode* you must write the -111 value in the ModesOfOperation parameter. The *Profile Velocity AI Mode* has the automatic Enable functionality.

ADVICE

**The filter that's applied on the analog input may limit the dynamic of the velocity reference and of the profiler ramps. To have a velocity reference that varies quickly you have to remove the filter and to increase the profiler ramps (ProfileAccceleration and ProfileDeceleration).**

## 22.17. *Torque Mode*

The *Torque Mode* is used to check the motor with a torque reference. To use this mode it's necessary to set the ModesOfOperation with the value 4 (*Torque Mode*) and then set the drive to the Operation enable state as described in the Section 8.5, *CiA402 state machine*. Later you can run the motion by writing the torque reference TargetTorque.

The parameters that define how the torque referred movement has to be executed are:

- TargetTorque;
- TorqueSlope;
- TorqueProfileType;

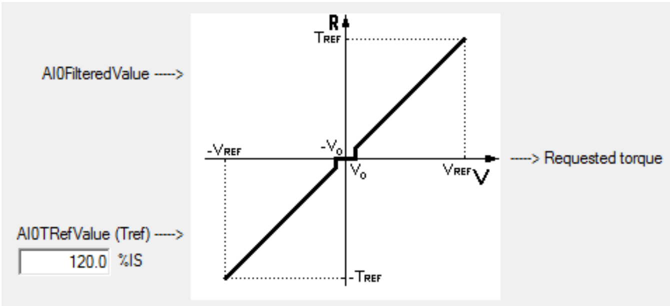
After the drive is parametrized and set to the Operation enable mode, a torque with module and direction consistent with the value in the TargetTorque parameter is applied on the motor.

In the Statusword there is a bit that shows the motion status:

- bit *Target reached*: bit showing the status of the Torque target reached;

## 22.18. *Torque AI Mode*

The *Torque AI Mode* is used to run a motion in torque created from the drive as it happens for the *Torque Mode*, but in this case the requested torque is not set through the Target-Torque parameter but it is obtained from AI0FilteredVoltage. The conversion of the values from the voltage to the requested torque occurs according to what is reported in Section 17.3, *Conversion*.




IMPORTANT

**For the Torque AI Mode you can apply all the considerations related to the Torque Mode and its parameters, as explained in Section 22.17, *Torque Mode*, except from what was stated for the TargetTorque parameter.**

To enable the *Torque AI Mode* you must write the -101 value in the ModesOfOperation parameter. The *Torque AI Mode* has the automatic Enable functionality.

ADVICE

**The filter that's applied on the analog input may limit the dynamic of the torque reference. To have a torque reference that varies quickly you have to remove the filter.**

## 22.19. *Homing Mode*

The *Homing Mode* is used to bring the motor on a known position, using some external references as the Positive limit switch (FC +), the Negative limit switch (FC -), the mechanical stop, the Home switch and the Index of the feedback sensor. This operating mode can be also used to run the preset of PositionActualValue without running any motion. The *Homing Mode* meets the specifications of the CiA-402.

⚠ NOTICE

**Regardless of which feedback sensor type is on the *i*BMD, the homing procedures are always the same, it only change the position reference subsistence conditions. For further information see Non-resettable parameters.**

ADVICE

**To configure the digital inputs like Positive limit switch (FC +), Negative limit switch (FC -) or Home, see Chapter 16, *Digital inputs and outputs*.**

WARNING

**If the Polarity is of Reverse type, the roles of Positive limit switch (FC +) and Negative limit switch (FC -) are reversed: Positive limit switch (FC +) behaves like Negative limit switch (FC -) and Negative limit switch (FC -) behaves like Positive limit switch (FC +). This is true both in the text of this manual and in the MotionDrive.**

NOTICE

**If you change the Polarity it will be necessary to re-execute the homing procedure.**

WARNING

**If a HomingMethod (see also Table 22.3) with mechanical stop is selected (e.g. mode -1), remember to set the torque limit (see Section 19.3, *Torque limit*).**

The following positions related to the *Homing Mode* are defined.

- **End position**: physical position of the motor at the end of the homing procedure, when the motor is stopped after the deceleration ramp
- **Home position**: physical position of the motor where the final phase of the homing procedure is noticed
- **Zero position**: physical position of the motor where PositionActualValue is equal to 0 inc.
- HomeOffset: difference between Zero position and Home position.

In the following picture can be found an example of homing motion with searching of the Home switch and the Index of the feedback sensor. When the motion starts, Home switch is not engaged and the motor is moved in the positive direction at the *SpeedForSwitch* speed. With the engagement of the Home switch, the motion is reversed and taken to the *SpeedForZero* speed. After the disengagement of the switch, the motor is stopped on the first Index found on the feedback sensor.

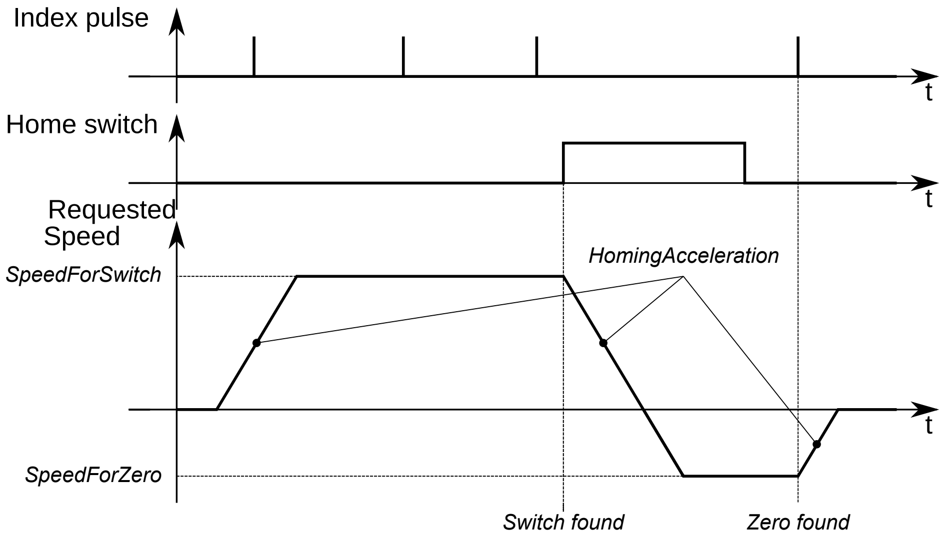



*Figure 22.9. Example of timing chart of a homing profile (HomingMethod = 7).*

At the end of the homing procedure, a preset of the position is carried out. PositionActualValue gets the value according to the following formula:

PositionActualValue = End position - Home position - HomeOffset

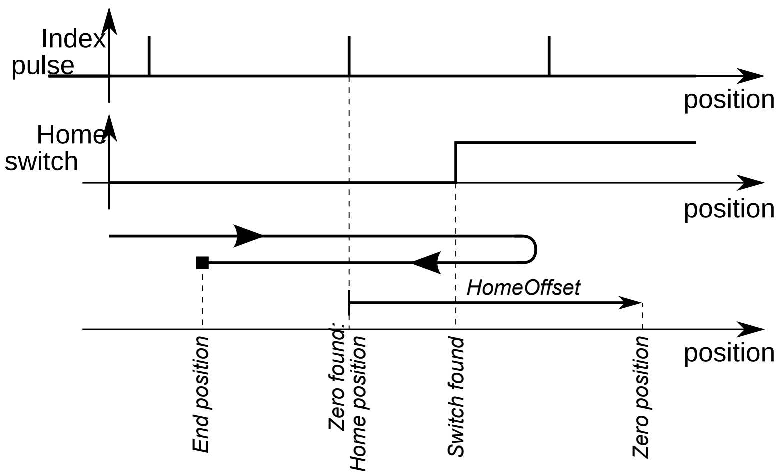



*Figure 22.10. Example of position trend of a homing profile (HomingMethod = 7).*

To use this operating mode, you must set the ModesOfOperation with the value 6 (*Homing Mode*). Then you proceed writing the parameters that define how the profile and the homing procedure must be run:

- HomingMethod, see the following table;
- HomeOffset;
- SpeedForSwitch;
- SpeedForZero;
- StartVelocity;
- EndVelocity;
- HomingAcceleration;
- IndexPulseDeadZone.

<table>
<tr><th>Val.</th><th>Procedure description</th></tr>
<tr><td>1</td><td>The motor is moved in the negative direction with SpeedForSwitch velocity. With the engagement of the Negative limit switch (FC -), the motion is reversed and taken to the speed SpeedForZero. After the disengagement of the limit switch, the motor is stopped on the first detected Index pulse.</td></tr>
<tr><td>2</td><td>The motor is moved in the positive direction with SpeedForSwitch velocity. With the engagement of the Positive limit switch (FC +), the motion is reversed and taken to the speed SpeedForZero. After the disengagement of the limit switch, the motor is stopped on the first detected Index pulse.</td></tr>
<tr><td>7</td><td>You can find the following sub-cases:<br>a. At the start of the motion Home switch is not engaged, the motor is moved in the positive direction with SpeedForSwitch velocity. With the engagement of the Home switch, the motion is reversed and taken to the speed SpeedForZero. After the disengagement of the switch, the motor is stopped on the first detected Index pulse.<br>b. At the start of the motion Home switch is engaged, the motor is moved in the negative direction with SpeedForZero velocity. After the disengagement of the Home switch, the motor is stopped on the first detected Index pulse.<br>c. At the start of the motion Home switch is not engaged, the motor is moved in the positive direction with SpeedForSwitch velocity. With the engagement of the Positive limit switch (FC +), the motor motion is reversed. With the engagement of the Home switch, the motion is taken to the speed SpeedForZero. After the disengagement of the switch, the motor is stopped on the first detected Index pulse.</td></tr>
<tr><td>8</td><td>You can find the following sub-cases:<br>a. At the start of the motion Home switch is not engaged, the motor is moved in the positive direction with SpeedForSwitch velocity. After the engagement of the Home switch, the motion is taken to the speed SpeedForZero. The motor is stopped on the first detected Index pulse.<br>b. At the start of the motion Home switch is engaged, the motor is moved in the negative direction with SpeedForZero velocity. After the disengagement of the Home switch, the motion is reversed. After a new engagement of the switch, the motor is stopped on the first detected Index pulse.<br>c. At the start of the motion Home switch is not engaged, the motor is moved in the positive direction with SpeedForSwitch velocity. With the engagement of the Positive limit switch (FC +), the motor motion is reversed. With the engagement of the Home switch, the motion is taken to the speed SpeedForZero. With the disengagement of the switch, the motion is reversed again. After a new engagement of the Home, the motor is stopped on the first detected Index pulse.</td></tr>
<tr><td>9</td><td>You can find the following sub-cases:<br>a. At the start of the motion Home switch is not engaged, the motor is moved in the positive direction with SpeedForSwitch velocity. With the engagement of the Home switch, the motion is taken to the speed SpeedForZero, with the disengagement of the switch the motion is reversed. After a new engagement of the Home, the motor is stopped on the first detected Index pulse.<br>b. At the start of the motion Home switch is engaged, the motor is moved in the positive direction with SpeedForZero velocity. With the disengagement of the Home switch, the motion is reversed. After a new engagement of the switch, the motor is stopped on the first detected Index pulse.<br>c. At the start of the motion Home switch is not engaged, the motor is moved in the positive direction with SpeedForSwitch velocity. With the engagement of the Positive limit switch (FC +), the motor motion is reversed. With the engagement of the Home switch, the motion is taken to the speed SpeedForZero. The motor is stopped on the first detected Index pulse.</td></tr>
<tr><td>10</td><td>You can find the following sub-cases:<br>a. At the start of the motion Home switch is not engaged, the motor is moved in the positive direction with SpeedForSwitch velocity. With the engagement of the Home switch, the motion is taken to the speed SpeedForZero. After the disengagement of the switch, the motor is stopped on the first detected Index pulse.<br>b. At the start of the motion Home switch is engaged, the motor is moved in the positive direction with SpeedForZero velocity. After the disengagement of the Home switch, the motor is stopped on the first detected Index pulse.<br>c. At the start of the motion Home switch is not engaged, the motor is moved in the positive direction with SpeedForSwitch velocity. With the engagement of the Positive limit switch (FC +), the motor motion is reversed. With the engagement of the Home switch, the motion is reversed and taken to the speed SpeedForZero. After the disengagement of the switch, the motor is stopped on the first detected Index pulse.</td></tr>
</table>

| Val. | Procedure description |
|---|---|
| 11 | Symmetrical to the 7. Differences: reversed speed signs; in the sub-case c) reversed motion on the Negative limit switch (FC -). |
| 12 | Symmetrical to the 8. Differences: reversed speed signs; in the sub-case c) reversed motion on the Negative limit switch (FC -). |
| 13 | Symmetrical to the 9. Differences: reversed speed signs; in the sub-case c) reversed motion on the Negative limit switch (FC -). |
| 14 | Symmetrical to the 10. Differences: reversed speed signs; in the sub-case c) reversed motion on the Negative limit switch (FC -). |
| 17 | Same as 1. Differences: without search for Index pulse, motion stopped on the correct edge of the limit switch. |
| 18 | Same as 2. Differences: without search for Index pulse, motion stopped on the correct edge of the limit switch. |
| 23 | Same as 7. Differences: without search for Index pulse, motion stopped on the correct edge of the Home switch. |
| 26 | Same as 10. Differences: without search for Index pulse, motion stopped on the correct edge of the Home switch. |
| 27 | Symmetrical to the 7. Differences: reversed speed signs; in the sub-case c) reversed motion on the Negative limit switch (FC -); without search for Index pulse, motion stopped on the correct edge of the Home switch. |
| 30 | Symmetrical to the 10. Differences: reversed speed signs; in the sub-case c) reversed motion on the Negative limit switch (FC -); without search for Index pulse, motion stopped on the correct edge of the Home switch. |
| 35 | The motor does not move and the drive takes the current position as Home position. |
| -1 | The motor is moved in the negative direction with *SpeedForSwitch* velocity. When the mechanical stop is reached, the movement is inverted and taken to the *SpeedForZero* velocity. The motor is stopped on the first detected Index pulse. |
| -2 | The motor is moved in the positive direction with *SpeedForSwitch* velocity. When the mechanical stop is reached, the movement is inverted and taken to the *SpeedForZero* velocity. The motor is stopped on the first detected Index pulse. |
| -7 | Same as 7. Differences: with the engagement of a limit switch the procedure is stopped and you get an error message (bit *Homing error* = 1). |
| -8 | Same as 8. Differences: with the engagement of a limit switch the procedure is stopped and you get an error message (bit *Homing error* = 1). |
| -9 | Same as 9. Differences: with the engagement of a limit switch the procedure is stopped and you get an error message (bit *Homing error* = 1). |
| -10 | Same as 10. Differences: with the engagement of a limit switch the procedure is stopped and you get an error message (bit *Homing error* = 1). |
| -11 | Symmetrical to the 7. Differences: reversed speed signs; with the engagement of a limit switch the procedure is stopped and you get an error message (bit *Homing error* = 1). |
| -12 | Symmetrical to the 8. Differences: reversed speed signs; with the engagement of a limit switch the procedure is stopped and you get an error message (bit *Homing error* = 1). |
| -13 | Symmetrical to the 9. Differences: reversed speed signs; with the engagement of a limit switch the procedure is stopped and you get an error message (bit *Homing error* = 1). |
| -14 | Symmetrical to the 10. Differences: reversed speed signs; with the engagement of a limit switch the procedure is stopped and you get an error message (bit *Homing error* = 1). |
| -17 | Same as -1. Differences: after the mechanical stop is reached and the direction is inverted, the motor is distanced from the mechanical stop with a minimum pulse number set on HomingPosDisengagement (minimum disengagement position). |
| -18 | Same as -2. Differences: after the mechanical stop is reached and the direction is inverted, the motor is distanced from the mechanical stop with a minimum pulse number set on HomingPosDisengagement (minimum disengagement position). |
| -23 | Same as 7. Differences: without search for Index pulse, motion stopped on the correct edge of the Home switch; with the engagement of a limit switch the procedure is stopped and you get an error message (bit *Homing error* = 1). |

| Val. | Procedure description |
|---|---|
| -26 | Same as 10. Differences: without search for Index pulse, motion stopped on the correct edge of the Home switch; with the engagement of a limit switch the procedure is stopped and you get an error message (bit *Homing error* = 1). |
| -27 | Symmetrical to the 7. Differences: reversed speed signs; without search for Index pulse, motion stopped on the correct edge of the Home switch; with the engagement of a limit switch the procedure is stopped and you get an error message (bit *Homing error* = 1). |
| -30 | Symmetrical to the 10. Differences: reversed speed signs; without search for Index pulse, motion stopped on the correct edge of the Home switch; with the engagement of a limit switch the procedure is stopped and you get an error message (bit *Homing error* = 1). |
| -35 | Same as 35. Differences: the drive takes RequestedPosition as Home position. |

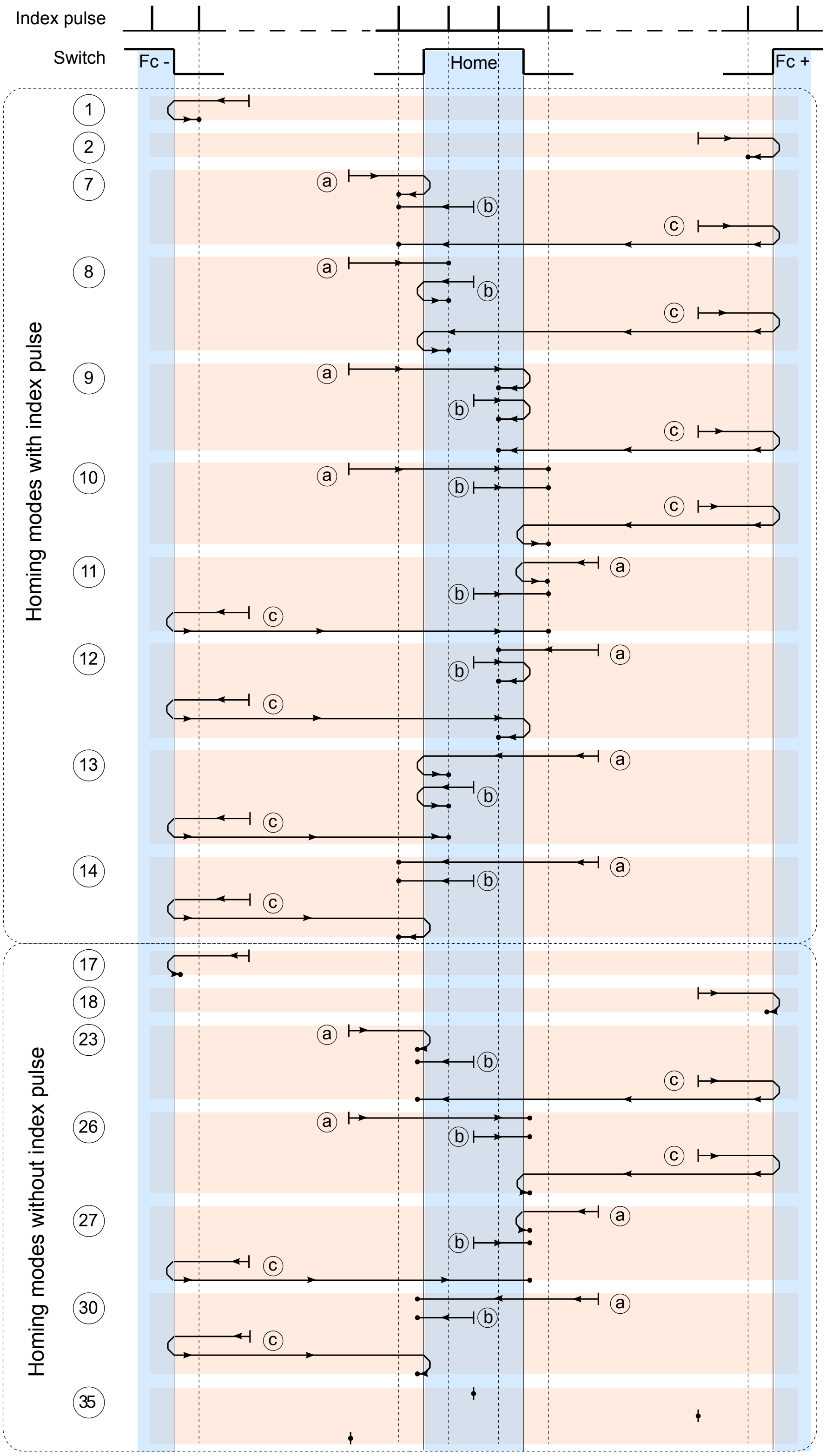
***Table 22.3. Available Homing procedures (HomingMethod).***

⚠ WARNING

**If the Polarity is of Reverse type, the roles of Positive limit switch (FC +) and Negative limit switch (FC -) are reversed: Positive limit switch (FC +) behaves like Negative limit switch (FC -) and Negative limit switch (FC -) behaves like Positive limit switch (FC +). This is true both in the text of this manual and in the MotionDrive.**

⚠ WARNING

**If a HomingMethod (see also Table 22.3) with mechanical stop is selected (e.g. mode -1), remember to set the torque limit (see Section 19.3, *Torque limit*).**




Legenda:

⊢ : axis position at the beginning of the homing movement

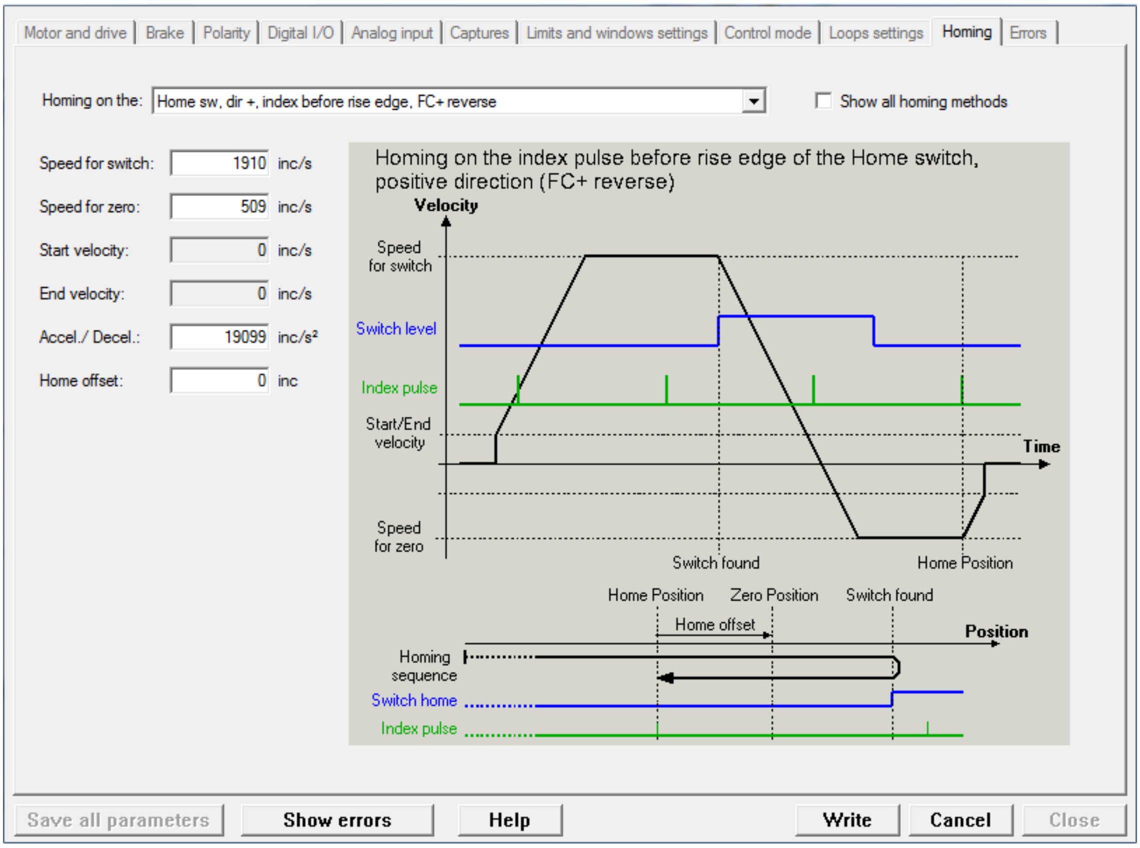
—• : axis position at the end of the homing movement

***Figure 22.11. Available Homing CiA-402 procedures.***

Access with MotionDrive:

**Main menu** > **Drive** > **Drive setup ...** > **Homing**

**Toolbar** >  > **Homing**




NOTE

**On the digital input with functionalities of Home a filtering at 10ms is run: the input status is considered as valid if it remains unchanged for at least 10 ms.**

IMPORTANT

**When a homing method which uses the Index pulse is selected, it's not permitted to use the capture peripheral A to execute other captures and in the capture peripheral B it is not possible to configure the Index pulse as trigger source.**

After having parametrized the drive and taken it to the Operation enable state, you can start the homing procedure by setting the bit *Homing operation start* of the Controlword. On the Statusword you can see the status of the bits procedure.

- bit *Target reached*: it shows if the procedure is ended;
- bit *Homing attained*: it shows if the homing procedure was correctly concluded;
- bit *Homing error*: it shows that there has been an error during the procedure running.

For further details on these bits, please refer to Table 8.12.

## 22.20. *Tuning Mode*

This operating mode is used only for the drive configuration and calibration. It is temporary enabled by the drive when some commands of the SysMngCommand are requested.

# Chapter 23
# Oscilloscope and monitoring

## 23.1. Parameters monitoring

To monitor the parameters it is possible to follow three methods:

1. Object dictionary for non repetitive instant monitoring (Section 27.2, *Reading and writing a parameter*)
2. Show variables (Figure 3.1) for repetitive instant monitoring
3. MotionDrive oscilloscope for longer monitoring in precise time intervals (Section 23.2, *Monitoring by oscilloscope*).

| Parameter | Show variables | Oscilloscope |
|---|---|---|
| PowerTemperature | *YES* | - |
| LogicTemperature | *YES* | - |
| MotorTemperature | *YES* | - |
| FeedbackSensorTemperature | *YES* | - |
| DCBusVoltage(+HV) | *YES* | *YES* |
| ActualMotorCurrent | *YES* | *YES* |
| ActualFieldCurrent | *YES* | - |
| ActualTorqueCurrent | *YES* | *YES* |
| RMSMotorCurrent | *YES* | *YES* |

| Parameter | Show variables | Oscilloscope |
|---|---|---|
| RequestedField | - | *YES* |
| ActualField | - | *YES* |
| RequestedTorque | *YES* | *YES* |
| ActualTorque | - | *YES* |
| ActualFilteredTorque | *YES* | - |
| ActualCurrentU | - | *YES* |
| ActualCurrentV | - | *YES* |
| ActualCurrentW | - | *YES* |
| I2TValue | *YES* | *YES* |
| RequestedSpeed | - | *YES* |
| VelocityActualValue | - | *YES* |
| SpeedFollowingError | - | *YES* |
| FeedForwardSpeed | - | *YES* |
| FeedForwardAcceleration | - | *YES* |
| RequestedPosition | - | *YES* |
| PositionActualValue | - | *YES* |
| PositionFollowingError | *YES* | *YES* |
| AI0Voltage | - | *YES* |
| AI0FilteredVoltage | *YES* | *YES* |
| AuxiliaryEncoderPosition | *YES* | - |
| AuxiliaryEncoderSpeed | *YES* | *YES* |
| Motor electric angle | - | *YES* |
| Feedback electric angle | - | *YES* |
| Feedback incremental counter | - | *YES* |
| Feedback hall status | - | *YES* |
| Feedback cosine | - | *YES* |
| Feedback sine | - | *YES* |

## 23.2. Monitoring by oscilloscope

Through the MotionDrive oscilloscope it is possible to acquire up to four channels at the same time. Any of the four channels, conventionally given to the vertical axis Y, and the temporal channel, conventionally given to the horizontal axis X, can be set through the related box in the tab Channels.

NOTE

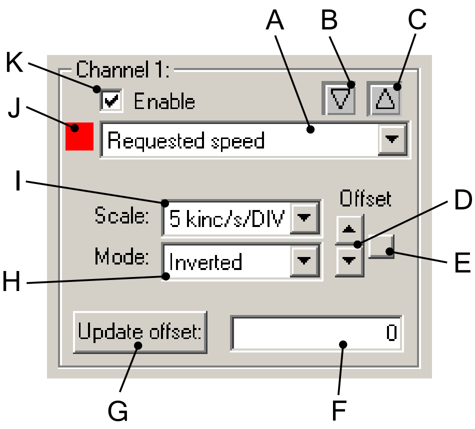
**When the Function Generator and Tuning tabs are enabled, MotionDrive can only run the functions that are at disposal by the Oscilloscope, Oscilloscope screen and Trigger windows. To enable the other functionalities again it is necessary to close the oscilloscope and reopen it by using only the following paths.**

Access:

**Main menu** > **Drive** > **Oscilloscope ...** > **Channels**

**Toolbar** >  > **Channels**

To start a capture see Section 23.3, *Setting the oscilloscope Trigger*, while hereafter you can find the settings related to the oscilloscope channels.




In the previous box you can find:

A. selecting the parameter to capture
B. indicator warning when the track gets over the lowest Screen limit; to let the track get back within bounds, change the scale or the offset
C. indicator warning when the channel gets over the highest Screen limit; to let the track get back within bounds, change the scale or the offset
D. buttons used to modify the offset
E. buttons used to centre the track in the Screen
F. field used to read and write the offset
G. button used to update the offset after having written it in the field F
H. selecting the display mode of the track:
   - *Disable o Hide* they hide the track
   - *Inverted* it inverts the track sign
   - *Normal* normal track display
   - *Zero* resets the track points
I. selecting the channel scale (see also the options in Scale and autoscale options)
J. track colour (to modify it, see Tracks preferences)
K. option used to enable the channel capture

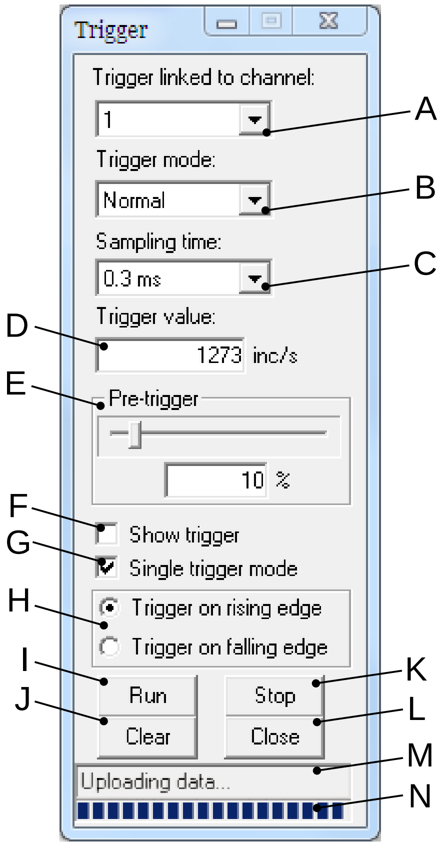
## 23.3. Setting the oscilloscope Trigger

Access:

**Main menu** > **Drive** > **Oscilloscope ...** > **Channels** > View trigger...

**Toolbar** >  > **Channels** > View trigger...

To set the event of *trigger*, which starts the data capture, refer to the Trigger window.




In the Trigger window there are:

A. selecting the channel the trigger is associated to
B. mode selection:
   - *Auto*: data capture without waiting for the trigger
   - *Normal*: data capture waiting for the trigger.

ADVICE

**If during the Normal mode no capture starts, enable the display of the trigger level (F), capture through the Auto mode, check on the screen the channel progress the trigger is associated to (A), set correctly the value (D) and the trigger edge (H) and try to capture again (I).**

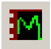
C. selecting the sampling time; if the sampling time increases, the capture interval proportionally increases
D. setting the trigger value
E. setting the points percentage you want to capture before the trigger event, as to the total number of the single track points. For example, if the Pre-trigger is equal to 50%, the trigger event will take place at half track
F. enabling the display at the trigger level
G. option used to run a single capture
H. selecting the trigger edge
I. button used to enable the data capture
J. button used to delete the captured data
K. button used to stop the capture in progress; the data will not be displayed in the Screen
L. button used to close the trigger window

M. reading the oscilloscope status; in case of error the Screen data will not be validated
N. reading the percentage of the progress of the oscilloscope status.

## 23.4. Saving or uploading an oscilloscope capture

Open the tab Channel. Access:

**Main menu** > **Drive** > **Oscilloscope ...** > **Channels**

**Toolbar** >  > **Channels**

To upload a capture saved in a file in the oscilloscope, push the button 

and then the button 

, then select the file to upload.

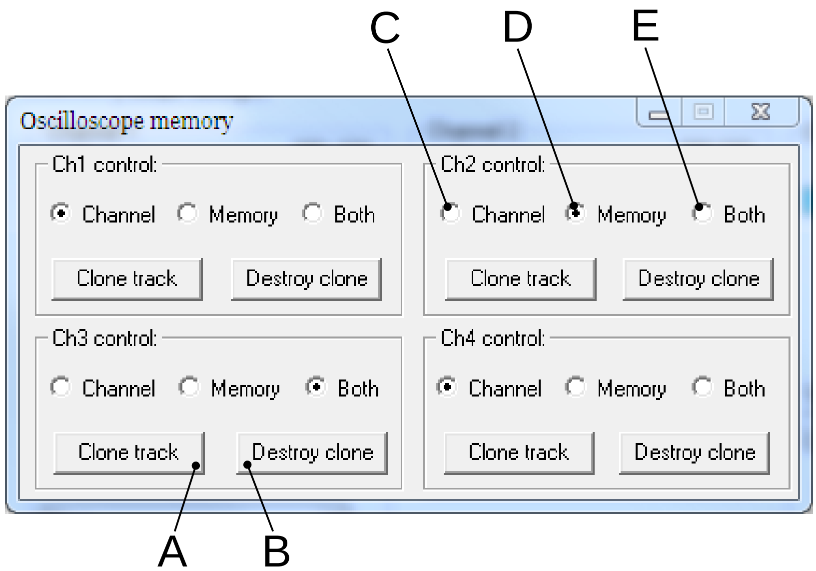
IMPORTANT

**The saved data are independent from the parameters and from the resolutions that are set in the drive. On consequence the oscilloscope will show the data in the charts according to the parameters that has been inserted (for example, a position profile will depend on the revolution resolution that has been set).**

To save the capture in a file, push the button 

. When you save the file you can also add a short description of the capture to save.
To save the captured tracks in the Screen in order to compare them with any new captures, open the window Oscilloscope Memory by pushing the button 

.




In the window Oscilloscope memory you can find:

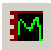
A. button used to save the track in the Screen

B. button used to delete the track saved in the Screen
C. option used to see the chart of only the captured track and modify its features
D. option used to see the chart of only the saved track and modify its features
E. option used to see the chart both of the captured track and the saved track and modify their features

## 23.5. Processing the captured tracks by the oscilloscope

Open the tab Channel. Access:

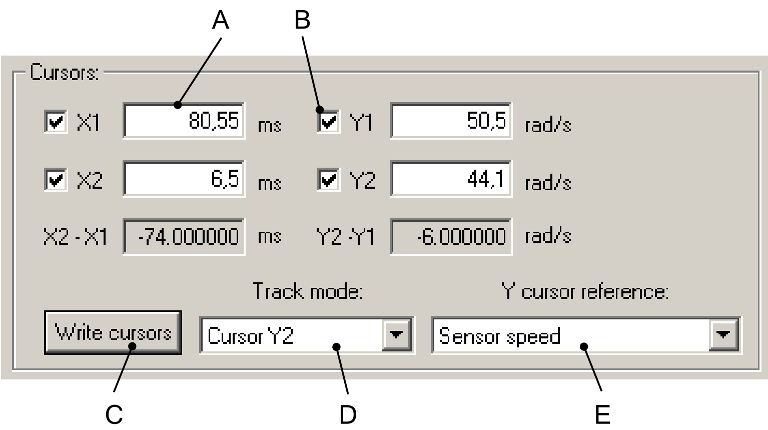
**Main menu** > **Drive** > **Oscilloscope ...** > **Channels**

**Toolbar** >  > **Channels**

**Autoscale**
To set the Screen tracks automatically, according to the settings selected in Scale and autoscale options, push the button 

.

**Measures**
To measure the Screen tracks use the box Cursors.




In the box Cursors you can find:

A. field used to read and write the cursors values
B. option used to enable the cursors
C. button used to confirm the digital values in the fields A
D. selecting the functionalities associated to the mouse:
   - *None*: no operation
   - *Track*: displaying the coordinates pointed with the mouse, by pushing the mouse left button in the Screen
   - *Zoom*: zooming a Screen portion

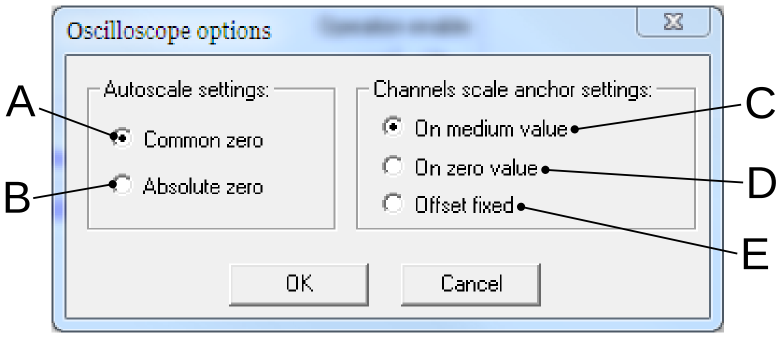
- *Pan XY*: tracks movement along the axes X and Y by moving the mouse pointer on the screen and pushing the left button
- *Pan X*: tracks movement along the axis X by moving the mouse pointer on the screen and pushing the left button
- *Pan Y*: tracks movement along the axis Y by moving the mouse pointer on the screen and pushing the left button
- *Cursor X1*: enabling and positioning the cursor X1 with the mouse on the Screen
- *Cursor X2*: enabling and positioning the cursor X2 with the mouse on the Screen
- *Trigger*: positioning the trigger with the mouse on the Screen, only if the option Show trigger is enabled (see the point F in Section 23.3, *Setting the oscilloscope Trigger*)
- *Cursor Y1*: enabling and positioning the cursor Y1 with the mouse on the Screen
- *Cursor Y2*: enabling and positioning the cursor Y2 with the mouse on the Screen

E. selecting the channel the cursors Y1 and Y2 refer to

**Scale and autoscale options**

To modify the features of the scale and autoscale change open the window Oscilloscope options by pushing the button 

.




In the window Oscilloscope options you can find:

A. option of command Autoscale forcing to zero the channels offset
B. option of command Autoscale modifying the offset in order to maximize the scale
C. option of scale change setting the average tracks value on the Screen
D. option of scale change setting the tracks zero on the Screen
E. option of scale change maintaining the tracks offset

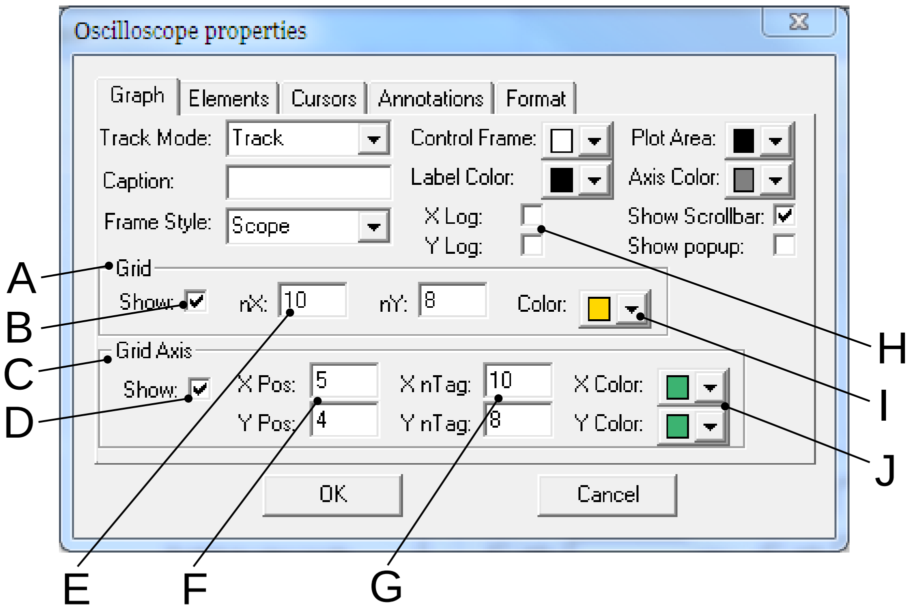
**Screen dimensions**

To modify the dimensions of the window Oscilloscope Screen, drag the window borders with the mouse. If the window is closed push the button 

.

**Screen preferences**

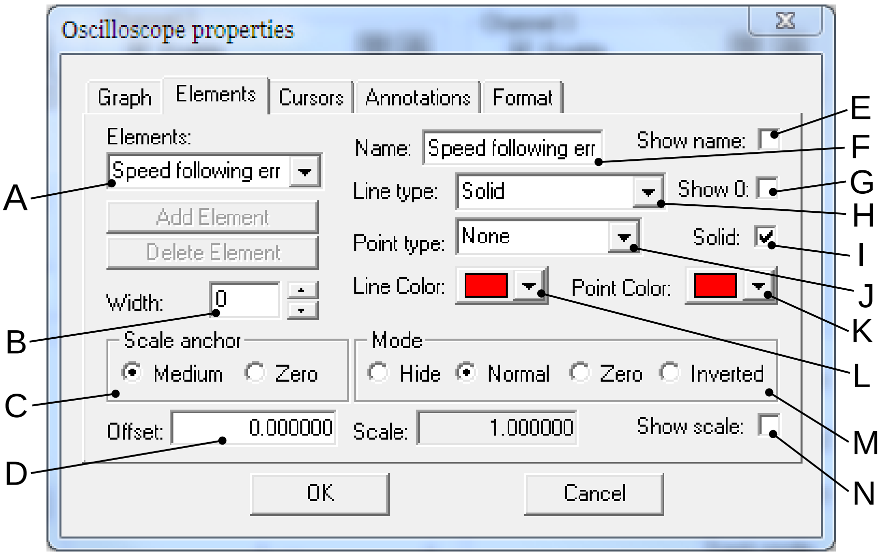
To modify the Screen preferences open the window Properties - ScopeX Control, by double clicking with the mouse left button on the window Oscilloscope Screen.




In the tab *Graph*, of the window Properties - ScopeX Control, you can find:

A. box regarding the main grid
B. option used to display the grid
C. box regarding the axes
D. option used to display the axes
E. total number of divisions of the main grid
F. position of the axes origin expressed according to the divisions number of the main grid
G. number of subdivisions reported on the axes, for each division of the main grid
H. options used to display the logarithmic scales
I. selecting the colour of the main grid
J. selecting the axes colours

**Tracks preferences**
To modify the tracks preferences open the window Properties - ScopeX Control, by clicking twice with the mouse left button on the window Oscilloscope Screen.




In the tab *Elements*, of the window Properties - ScopeX Control, you can find:

A. selection of the channel to modify
B. track dimension
C. track scale change (see the point I in Section 23.2, *Monitoring by oscilloscope*)
D. modification of the track offset (see the point F in Section 23.2, *Monitoring by oscilloscope*)
E. option used to display the name of the captured parameter on the Screen
F. displayed name
G. option used to display the track zero
H. selecting the track line type
I. option used to fill the track points with the same colour selected at the point K
J. selecting the track points type
K. selecting the track points colour
L. selecting the track colour
M. selecting the channel mode (see the point H in Section 23.2, *Monitoring by oscilloscope*)
N. option used to display the channel scale

## Chapter 24

# Fault and Warning

The drives of the series *i*BMD, when finding some anomalies during working or some errors in the parameters setting, notify the error. Errors are divided into two categories depending on their seriousness:

- **Warning**, error which notifies a non-serious condition of the drive
- **Fault**, error preventing and stopping the motor motion; the drive is often in a serious error condition.

When the drive is enabled, the Faults are divided in two types:

- **Fatal fault**, Faults immediately preventing from controlling the motor motion
- **Non fatal fault**: faults letting you temporarily control the motor motion.

Errors can be:

- **Dynamic**, if the error condition is still present in the drive (otherwise the warning automatically disappear)
- **Retentive**, if the error is stored by the drive until it is reset.

In the following table you can find the features of the MainError, the bit which any error is associated to and the features of the masks defining the behaviour of the drive in case of Fault. The abbreviations WD, WR, FD, FR, FA, FE and FS have the following meaning:

- WD (WarnDynamic): main Dynamic Warnings

- WR (WarnRetentive): main Retentive Warnings
- FD (FaultDynamic): main Dynamic Faults
- FR (FaultRetentive): main Retentive Faults
- FA: Faults that can be set as self-restoring (the Fault Reset command is automatically run)
- FE: Faults that can be deactivated
- FS: Faults that can generate the Safety profile.
- FF: errors of Fatal Fault type.

| Bit | Error | MainError | | | | FaultMask | | | FF |
|---|---|---|---|---|---|---|---|---|---|
| | | WD | WR | FD | FR | FA | FE | FS | |
| 0 | DC bus over voltage | *YES* | *YES* | *YES* | *YES* | - | - | - | *YES* |
| 1 | Thermal management | *YES* | *YES* | *YES* | *YES* | - | - | - | - |
| 2 | Reserved | - | - | - | - | - | - | - | - |
| 3 | DC bus under voltage | *YES* | *YES* | *YES* | *YES* | *YES* | *YES* | - | *YES* |
| 4 | Power or motor short circuit | - | - | *YES* | *YES* | - | - | - | *YES* |
| 5 | Parameters soft error | *YES* | - | - | - | - | - | - | - |
| 6 | Parameters serious error | *YES* | - | *YES* | *YES* | - | - | - | - |
| 7 | Real time mode error | - | - | *YES* | *YES* | - | *YES* | *YES* | - |
| 8 | CAN communication error | *YES* | *YES* | *YES* | *YES* | - | *YES* | *YES* | - |
| | EtherCAT communication error | *YES* | *YES* | *YES* | *YES* | - | *YES* | *YES* | - |
| 9 | Reserved | - | - | - | - | - | - | - | - |
| 10 | Power or motor over current | *YES* | *YES* | *YES* | *YES* | - | - | - | *YES* |
| 11 | Reserved | - | - | - | - | - | - | - | - |
| 12 | Position following error | *YES* | *YES* | *YES* | *YES* | - | *YES* | - | - |
| 13 | Last command requested failed | *YES* | *YES* | *YES* | *YES* | - | - | - | - |
| 14 | /STO Management Error | - | - | *YES* | *YES* | - | - | - | *YES* |
| 15 | User Fault | - | - | *YES* | *YES* | - | *YES* | - | - |
| 16 | I2T limit reached | *YES* | *YES* | *YES* | *YES* | - | *YES* | - | - |
| 17 | I2T Warning threshold reached | *YES* | *YES* | - | - | - | - | - | - |
| 18 - 19 | Reserved | - | - | - | - | - | - | - | - |
| 20 | Limit reached | *YES* | - | - | - | - | - | - | - |
| 21 | Possible no tuning of regulator | *YES* | *YES* | - | - | - | - | - | - |
| 22 | Drive is in disable state, since the enable input is or has been in not active state | *YES* | *YES* | - | - | - | - | - | - |
| 23 | Feedback sensor error | - | - | *YES* | *YES* | - | - | - | *YES* |
| 24 | Digital IO configuration error | - | *YES* | *YES* | *YES* | - | - | - | - |
| 25 | Logic voltage error | *YES* | *YES* | *YES* | *YES* | - | *YES* | - | - |
| 26 | Motion parameter limited | *YES* | - | - | - | - | - | - | - |
| 27 | Digital output overtemperature or overload | *YES* | *YES* | - | - | - | - | - | - |
| 28 | Over Speed | - | - | *YES* | *YES* | - | - | - | *YES* |
| 29 - 30 | Reserved | - | - | - | - | - | - | - | - |
| 31 | Internal Error | - | - | *YES* | *YES* | - | - | - | - |

***Table 24.1. Features of the MainError.***

NOTE

**To choose the self-restoring Faults, use the FaultMaskAutoErase parameter.**
**To choose the Faults to enable/disable, use the FaultMaskEnable parameter.**
**To choose the Faults generating the Safety profile, use the FaultMaskSafetyPrfExecute parameter.**

## 24.1. Management of the errors with MotionDrive

In the tab Main of MotionDrive, in the frame *Drive error* you can find two indicators showing the status of the Warnings and of the Faults. The colours have the following meaning:

[0 - 0] indicator switched off, no error
[0 - 1] orange light, Warning presence
[1 - 1] red light, Fault presence.

Next to the indicators you can find the number of the found errors. The number on the left shows the number of the errors of dynamic type, the number on the right shows the number of errors of retentive type, both for Faults and for Warnings. To see the details of the errors, open the window Show errors:

**Main menu** > **Drive** > **Show errors ...**

**Tab Main** > Show errors

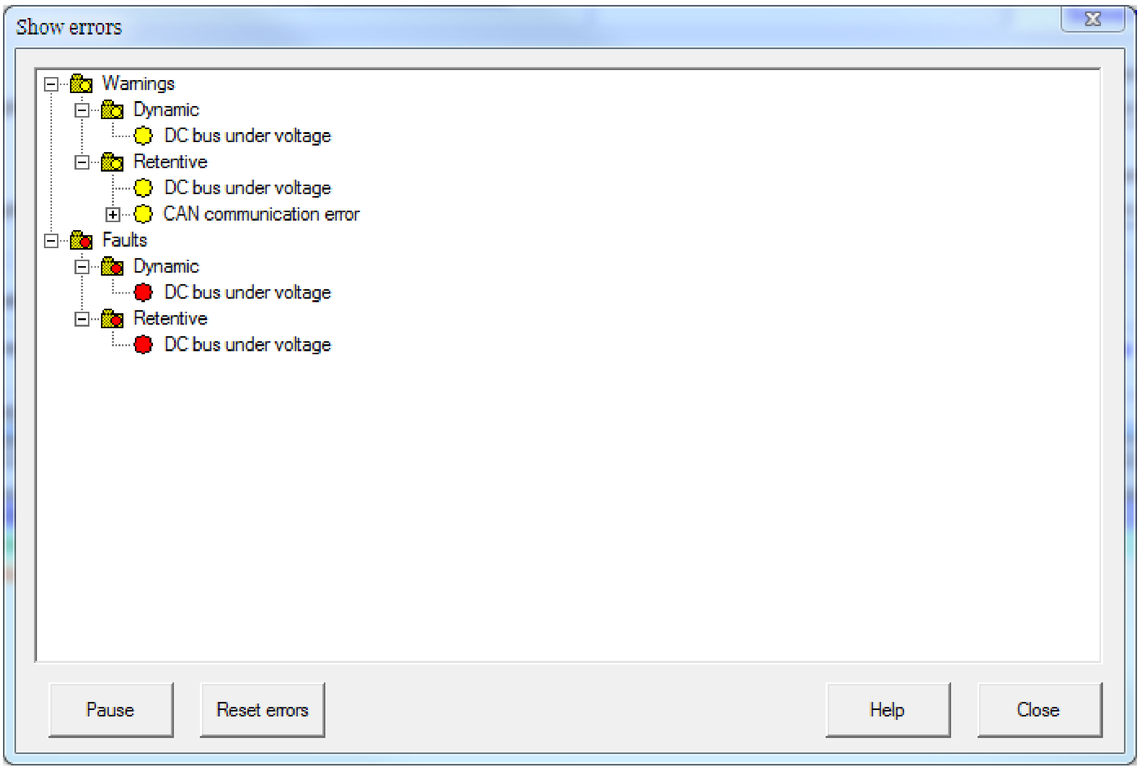



*Figure 24.1. Window Show errors.*

In the window Show errors, every error is placed next to an indicator, the colour of which has the following meaning:

green indicator, no error

yellow light, Warning
orange light, Warning becoming Fault if you try to enable the drive
red light, Fault.

The errors are written in Section 24.6, *Errors description*.

NOTE

**To reset the errors push the button** Reset errors **. To stop the updating of the errors shown in the page, push the button** Pause **.**

# 24.2. Monitoring the errors on the status leds

The *i*BMD drives show the status of the errors through the L1 and L2 leds (Section 7.3, *Leds*) that can take the following colours:

- green led, no error found, showing the enabling status of the drive (see Table 8.10)
- orange led, only Warnings and no Faults are present.
- red led, Faults presence.

In case of more errors, the leds only show the error that in the following chart has the lowest visualization order (that corresponds to a higher priority).

IMPORTANT

**In the boot and firmware start up phases of the *i*BMD the below description of the leds is no more valid. Please refer to Chapter 26, *Software updating***

| Error | L1 | L2 | Order |
|---|---|---|---|
| CAN communication error | 1 FL | ON | 11 |
| EtherCAT communication error | | | |
| Real time mode error | 1 FL | BLK | 12 |
| Position following error | 1 FL | 1 FL | 13 |
| Limit reached | 1 FL | 2 FL | 14 |
| I2T limit reached | 1 FL | 3 FL | 15 |
| I2T Warning threshold reached | 1 FL | 3 FL | 16 |
| Parameters soft error | 2 FL | BLK | 17 |
| Possible no tuning of regulator | 2 FL | 1 FL | 18 |
| Motion parameter limited | 2 FL | 2 FL | 19 |
| User Fault | 2 FL | ON | 21 |
| Over Speed | 3 FL | 1 FL | 23 |
| Internal Error | 3 FL | ON | 24 |
| /STO Management Error | BLK | ON | 6 |
| Feedback sensor error | BLK | BLK | 7 |
| Last command requested failed | BLK | 1 FL | 8 |

| Error | L1 | L2 | Order |
|---|---|---|---|
| Parameters serious error | BLK | 2 FL | 9 |
| Digital IO configuration error | BLK | 3 FL | 10 |
| Digital output overtemperature or overload | BLK | 3 FL | 20 |
| Drive is in disable state, since the enable input is or has been in not active state | BLK | ON | 22 |
| DC bus over voltage | ON | BLK | 0 |
| Power or motor short circuit | ON | ON | 1 |
| Power or motor over current | ON | ON | 2 |
| Thermal management | ON | 1 FL | 3 |
| DC bus under voltage | ON | 2 FL | 4 |
| Logic voltage error | ON | 3 FL | 5 |

*Table 24.2. Status of the L1 and L2 leds for drives iBMD in case of error.*

NOTE

**The leds activation depends on some parameters. The value of the FaultRetentive parameter, if different from 0, determines the activation of the red colour leds. If there are no Faults, but Warnings only, then the orange leds depend by the most significant bit between the ones that are different from 0 in the WarnDynamic and WarnRetentive parameters (this bit is determined by executing the OR logic operation between the two parameters).**

## 24.3. Reaction to the Warnings

When an error of Warning type occurs, the drive runs the following operations:

1. the bits of the parameters WarnDynamic, WarnRetentive and of any other parameter showing the details are set
2. In the CAN versions if a communication error is present and if a Warning of the CAN communication error, CANopenEmcyService and/or CANopenEmcyProcess parameters is active, then bit0 and bit4 of ErrorRegister are set.
3. with the CAN versions if a communication error is detected then the messages of EMCY are sent to the CANopen network (see Section 9.2, *Emergency object (EMCY)*)
4. if no Faults are active (FaultRetentive is equal to 0), the leds show the Warning according to the order in Table 24.2
5. the state of the CiA402 State Machine is not modified (see Section 8.5, *CiA402 state machine*).

NOTE

**When the error condition that generated the Warning is no longer noticed, the corresponding bits in the dynamic Warning parameters are reset.**

**In Table 24.1 you can find the errors of Warning type.**

**If all the causes that have activated the bit4 of ErrorRegister are removed, then the bit resets and, if it was the only present alarm, then even the bit0 resets.**

## 24.4. Reaction to the Faults

When an error of Fault type occurs, the drive executes the following operations:

1. the bits of the parameters FaultDynamic and FaultRetentive and of any other parameter showing the details are set
2. The bit (or the bits) that's related to the error type and the bit 0 of the ErrorRegister are set.
3. with the versions CAN the messages of EMCY are sent to the network CANopen
4. viene visualizzato dai led il Fault secondo l'ordine riportato in Table 24.2
5. the CiA402 State Machine goes to the *Fault Reaction Active* state (see Section 8.5, *CiA402 state machine*)
6. One of the following operations is executed on the motor:
   - If the fault is a Fatal Fault, then the motor is immediately disabled (see Faults with FF property in the Table 24.1);
   - If the fault is NOT a Fatal Fault and has not been deactivated, even if it is self restoring, (see Faults with FA and FE properties in the Table 24.1) then the drive executes a ramp stop and the motor disabling according to the FaultReactionOptionCode parameter setting;
   - If the fault is NOT a Fatal Fault and requires the Safety profile (see Faults with FS properties in Table 24.1), that can be set through the SafetyPrfCommand and FaultMaskSafetyPrfExecute parameters, then the drive executes the positioning and the disabling of the motor.

⚠ NOTICE

**If contemporary more faults happen and between these there are some with different severity (and so each one provides a different reaction) it's important to know that the 3 different reactions (above described on point 6) have a different priority:**

**- The more serious the Fault is, the more high is the priority.**

**In other words the FatalFault reaction has the highest priority, then follows the Fault reaction that requires the ramp stop, while the reaction that requires the positioning in the security position has the lowest priority.**

⚠ NOTICE

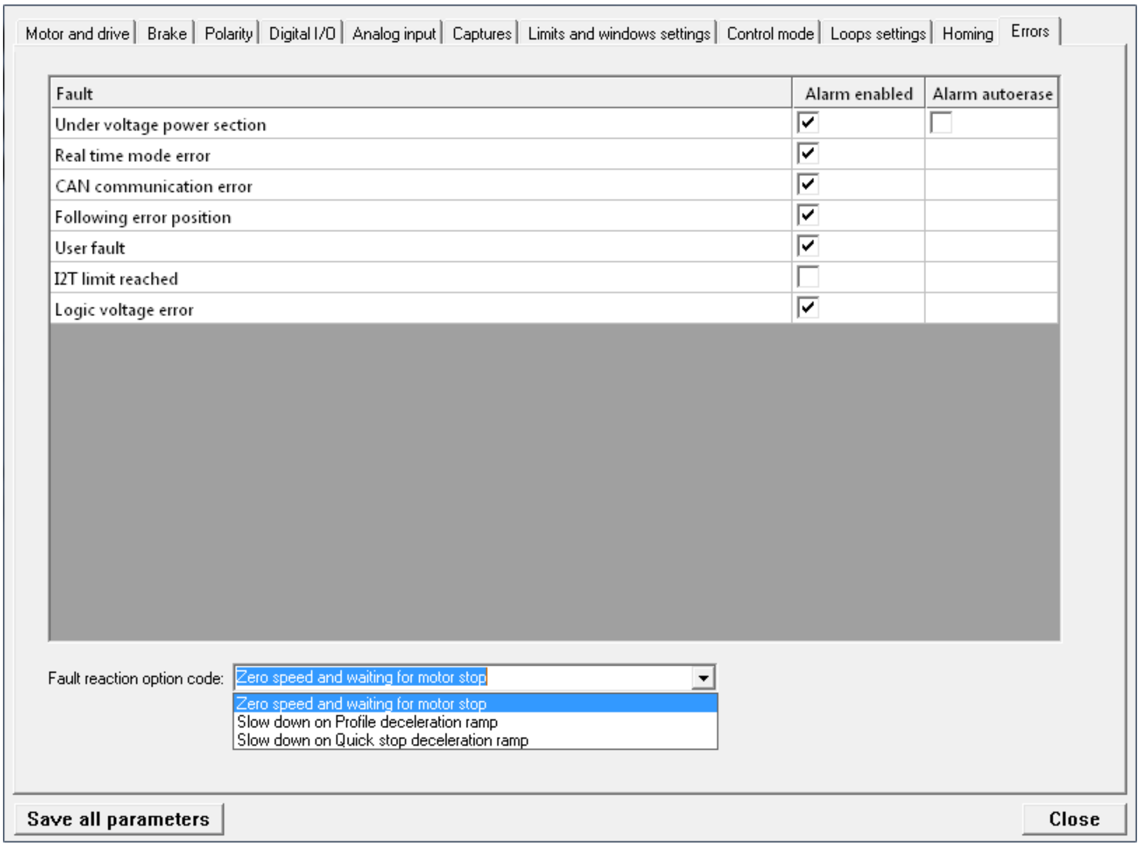
**If all the causes that have activated a particular bit of ErrorRegister are removed, then that bit resets; if all the alarm causes are removed, then even the generic bit0 is reset.**

The settings about Faults deactivation, self-restoring and selection of the motor deceleration ramp before the disabling, can be easily managed through the "Errors" table in the Drive Setup.

Access through MotionDrive:

**Main menu** > **Drive** > **Drive Setup ...** > **Errors**

**Tab Main** >  > **Errors**




- In the windows on the right it's possible to select which Faults are active (Alarm enable) and self-restoring (Alarm autoerase).
- In the drop-down menu on the bottom it's possible to select the deceleration ramp type that precedes the axis disabling, in case of the detection of one of the active Faults. Every item of this menu corresponds to a value of the FaultReactionOptionCode parameter, in particular:
  - Zero speed and waiting for motor stop = -1;
  - Slow down on Profile deceleration ramp = 1;
  - Slow down on Quick stop deceleration ramp = 2;

**⚠ ATTENTION**

**If a FE Fault is deactivated through the FaultMaskEnable parameter (see Table 24.1), the related Warning is anyway signaled, but the Fault reaction actions will not be executed.**

**NOTE**

**If the fault requires the possibility to solve with the Safety profile (see Faults with FS property on Table 24.1), and this profile has not been activated, the drive will execute the deceleration ramp as set on the FaultReactionOptionCode parameter.**

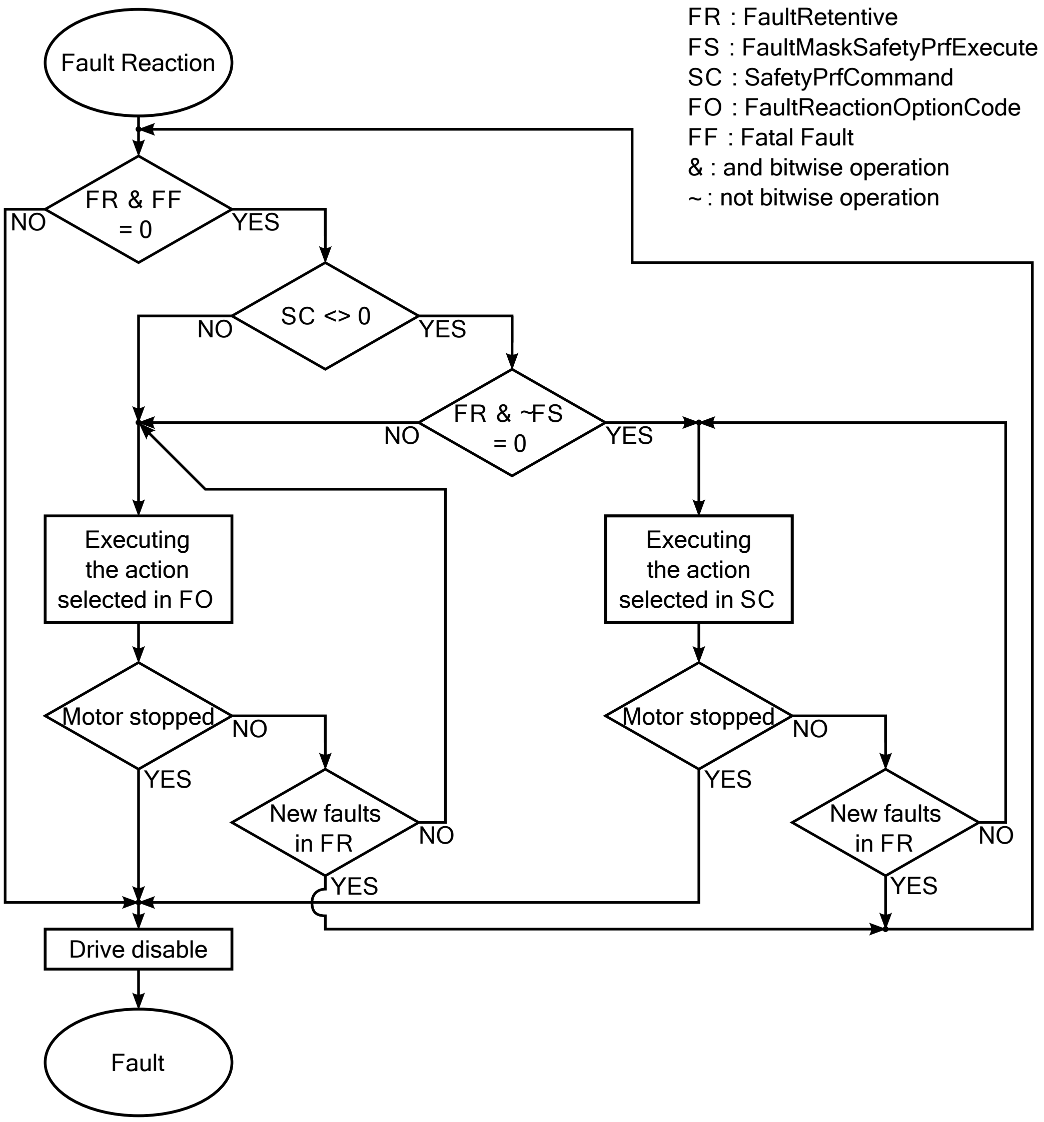



*Figure 24.2. Flow diagram of the reaction to the Faults.*

NOTE

**When the error condition that generated the Faults is no longer noticed, the corresponding bits in the dynamic Fault parameters are reset.**

**In Table 24.1 you can find the errors of Fault type.**

### Safety profile

The safety profile is a motion of the motor carried out in the reaction to the Faults to take the motor to a safe position. As a Fault is detected, check if the drive is the Operation enable state, if the profile is enabled (see SafetyPrfCommand) and if the detected Fault allows its running (see FaultMaskSafetyPrfExecute). If all the conditions are respected the operation shown in SafetyPrfCommand is run.

## 24.5. Resetting the errors

> ⚠ NOTICE
>
> **It is up to the operator to find and to eliminate the causes that have caused the Fault condition before running a command of Fault Reset. The continuous repetition of the command of Fault Reset without removing the causes could cause some permanent damage to the drive.**

To reset only the errors of retentive Warning type, write the WarnRetentive parameter; any written value is accepted.

> IMPORTANT
>
> **Before resetting the errors it is necessary to remove all the causes that generated them.**

To reset all errors, only if the CiA402 State Machine is in the *Fault* state (see Section 8.5, *CiA402 state machine*), write the Fault reset command in the Controlword. The command consists of a transition from 0 to 1 of the bit 7 of the Controlword following these symbols: xxxx xxxx ↑xxx xxxx (the status of the bits shown with 'x' is not important to determine the command). This command resets the errors of retentive type, only if there are some retentive Faults (FaultRetentive different from 0). The CiA402 State Machine switches to the *Switch On Disabled* state (see Section 8.5, *CiA402 state machine*) only if there are no other Faults (FaultDynamic value is 0).

> NOTE
>
> **Power or motor short circuit and Power or motor over current do not allow the running of the command Fault Reset before at least 20 seconds after the Fault event.**

## 24.6. Errors description

**DC bus over voltage**
Overvoltage for the power supply of the power section (DC bus) (Warning = 800 V; Fault = 840 V). Check the size of the power supply and the electrical connections. For any further information, see Section 14.3, *Regeneration*. The DC bus voltage can be monitorized through the DCBusVoltage(+HV) parameter.

**Thermal management**
Error related to the drive thermal management. The details can be found in the following chart and in the parameters ThermalManageError.

| Bit | Name | Type | Description |
|---|---|---|---|
| 0 | Power over temperature | W/F | Overtemperature of the power section (Warning = 105°C; Fault = 110°C). Check the environment temperature and the ventilation of the power section. |
| 1 | Control over temperature | W/F | Overtemperature of the control section (Warning = 85°C; Fault = 95°C). Check the environment tempera- |

| Bit | Name | Type | Description |
|---|---|---|---|
| | | | ture, the ventilation and the power consumption of the control section with an external amperometer. Check that the values of the current, that are reported in the Chapter 5, *Technical features*, are respected |
| 2 | Motor over temperature | W/F | Overtemperature of the motor (Warning = 10°C before of the Fault threshold; Fault = it depends on which motor is used). To know the Fault threshold value : from MotionDrive see MotorParameters table, from parameter see FaultTemperatureThrs. Verify the environment temperature, the ventilation, the dissipation, analyze the working cycle in relation to the motor performance and torque curves |
| 3 | Fan 1 stuck | W | Fan 1 stuck. Clean the fans and try again. |
| 4 | Fan 2 stuck | W | Fan 2 stuck. Clean the fans and try again. |
| 5 | Fan 3 stuck | W | Fan 3 stuck. Clean the fans and try again. |
| 6 | Power temp. sensor hardware failure | F | Failure of the Power Temperature Sensor of the power section. Please contact Bonfiglioli |
| 7 | Logic temp. sensor hardware failure | F | Failure of the Temperature Sensor of the control section. Please contact Bonfiglioli |
| 8 | Motor temp. sensor hardware failure | F | Failure of the Motor Temperature Sensor. Please contact Bonfiglioli |
| 9 | Feedback sensor over temperature | W/F | Feedback sensor over temperature. This error is present only if the position sensor is provided of the temperature sensor (and then the temperature measurement is supported by the hardware) Verify the environment temperature, the ventilation, the dissipation, analyze the working cycle in relation to the motor performance and torque curves. |
| 10 | Motor temperature sensor unknown - selection forced to none sensor | W | Temperature sensor unknown The firmware doesn't recognize the temperature sensor code, the system will maintain as selected the sensor that has been set by the user, but will internally manage it as it had been selected "None sensor" (MotorTemperatureSensorType = 0), so without monitoring the motor temperature. Every time that the system is turned off and on again and finds a sensor that's not provided by the firmware, this Warning will recur unless a motor without a temperature sensor, or with a supported one, is saved in EEPROM (see MotorTemperatureSensorType). |
| 11 - 15 | Reserved | | |

***Table 24.3. Details about ThermalManageError (W = Warning, F = Fault, W/F = both).***

### DC bus under voltage

Undervoltage for the power supply of the power section (DC bus) (Warning = 200 V; Fault = 150 V). Check the output voltage of the secondary of the transformer and the input voltage of the converter (if are present), check the drive supply voltage and the wirings, use the oscilloscope to monitor the power section voltage and to check its trend and, if in some particular motion condition a voltage drop happens, decrease the velocities and the accelerations of the working cycle and/or substitute the power supply with another one more powerful. The DC bus voltage can be monitorized through the DCBusVoltage(+HV) parameter.

### Power or motor short circuit

Power or motor short circuit error. Wait for 20 seconds before running the Fault Reset to allow the dissipation of the accumulated power. Check that the insulation voltage of the motor windings is compatible with the drive supply voltage. If the problem persists, please contact Bonfiglioli.

### Parameters soft error

Soft error in the drive parametrization. You can find the details in the following chart and in the parameter ParamSoftError and AI0CalibrationStatus.

| Bit | Name | Type | Description |
|---|---|---|---|
| 0 | I2T Limited to max drive value | W | UserMaxI2T higher than DriveMaxI2T. Decrease UserPeakCurrent and/or I2TTime. |
| 1 | Peak current too high for motor or drive | W | UserPeakCurrent higher than MotorPeakCurrent and/or MaxPeakCurrent. Decrease UserPeakCurrent. |
| 2 - 4 | Reserved | | |
| 5 | Loops configuration selected is not supported | W | LoopConfiguration not supported by the current firmware. Update the firmware or change configuration. |
| 6 | Software position limits incompatibility | W | PositionLimitPositive lower than PositionLimitNegative. Correct the limits |
| 7 | Capture Trigger Source equal on both Capture peripheral | W | CaptureTriggerInput_A equal to CaptureTriggerInput_B. Choose two different values. |
| 8 - 9 | Reserved | | |
| 10 | Capture A: Filter or trigger on both edges not allowed on selected trigger input | W | If it has been tried to contemporary set the capture on Index and the space filter (CaptureSource0_A) or the capture on both edges. Or it has been tried to set the space filter (CaptureSource0_A) with the CiA402 mode. Or it has been tried to modify the capture trigger with capture enabled. |
| 11 | Capture B: Filter or trigger on both edges not allowed on selected trigger input | W | If it has been tried to contemporary set the capture on Index and the space filter (CaptureSource0_B) or the capture on both edges. Or it has been tried to set the space filter (CaptureSource0_B) with the CiA402 mode. Or it has been tried to modify the capture trigger with capture enabled. |
| 12 | Capture A: Selected trigger not available (previous value has been kept) | W | The last value written in CaptureTriggerInput_A has been refused since it is not supported by the current firmware. Verify that the inserted data is valid, if necessary update the firmware. |
| 13 | Capture B: Selected trigger not available (previous value has been kept) | W | The last value written in CaptureTriggerInput_B has been refused since it is not supported by the current firmware. Verify that the inserted data is valid, if necessary update the firmware. |
| 14 | Reserved | | |
| 15 | Capture setup using disabled parameters' interface (look at parameter 'CaptureInterfaceMode') | W | It has been tried to access, in reading or writing, to the not selected interface (see CaptureInterfaceMode parameter). This bit cannot auto-reset, but must be reset by the user. |

***Table 24.4. Details about ParamSoftError (W = Warning, F = Fault, W/F = both).***

Parameters soft error is active even when AI0CalibrationStatus assumes the following values (W = Warning, F = Fault, W/F = both).

| Value | Name | Type | Description |
|---|---|---|---|
| 0 | Analog input 0 is not calibrated | W | Analog input 0 is not correctly calibrated. Run the calibration according to what is reported in Section 17.2, *Calibration* or update the current configuration and the permanent memory with the default values. |
| 1 | Analog input 0 calibration not complete (only offset) | W | |
| 2 | Analog input 0 calibration not complete (only gain) | W | |

**Parameters serious error**
Serious error in the drive parametrization. The details can be found in the following chart and in the parameters ParamSeriousError. The Warning becomes a Fault if you try to enable the drive.

| Bit | Name | Type | Description |
|---|---|---|---|
| 0 | Stall current not set | W/F | MotorStallCurrent is equal to 0. Turn the drive off and then on again. If the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |
| 1 | Motor peak current not set | W/F | MotorPeakCurrent is equal to 0. Turn the drive off and then on again. If the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |
| 2 | Motor torque constant not set | W/F | TorqueConstant(ForceConstant) is equal to 0. Turn the drive off and then on again. If the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |
| 3 | Motor inductance not set | W/F | MotorInductance is equal to 0. Turn the drive off and then on again. If the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |
| 4 | Motor resistance not set | W/F | MotorResistance is equal to 0. Turn the drive off and then on again. If the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |
| 5 | Motor inertia not set | W/F | MotorInertia(MotorMass) is equal to 0. Turn the drive off and then on again. If the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |
| 6 | Motor pole number not set | W/F | MotorPoles is equal to 0. Turn the drive off and then on again. If the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |
| 7 | Motor rated speed not set | W/F | MotorRatedSpeed is equal to 0. Turn the drive off and then on again. If the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |
| 8 | Sensor not set | W/F | FeedbackSensorCode is equal to 0. Turn the drive off and then on again. If te problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |
| 9 | Max rated current not set | W/F | MaxRatedCurrent is equal to 0. Please contact Bonfiglioli Mechatronic Research S.p.a |
| 10 | Max peak current not set | W/F | MaxPeakCurrent is equal to 0. Please contact Bonfiglioli Mechatronic Research S.p.a |
| 11 | Current not calibrated | W/F | Please contact Bonfiglioli Mechatronic Research S.p.a |
| 12 | Voltage not calibrated | W/F | Please contact Bonfiglioli Mechatronic Research S.p.a |
| 13 | Sensor not supported | W/F | FeedbackSensorCode not supported by the current firmware. Update the firmware or change the sensor. |
| 14 | Sensor not phased | W/F | Feedback sensor phasing problems. The cause is specified by the Feedback sensor error. Reset the errors, if the problem persists please contact Bonfiglioli Mechatronic Research S.p.a. |

| Bit | Name | Type | Description |
|---|---|---|---|
| 15 | Pole pitch not set | W/F | PolePitch is equal to 0. Turn the drive off and then on again. If the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |

***Table 24.5. Details about ParamSeriousError (W = Warning, F = Fault, W/F = both).***

**Real time mode error**

Error of *Interpolated Position Mode*. The details can be found in the following chart and in the parameter RealTimeModeError.

The following descriptive table refers to the parameters for the interpolation and to the synchronization methods. Their use is intended according to the following rules:

With *parameters for the interpolation* is meant:

- if ModesOfOperationDisplay = 7 → IpPosFirstParameter and IpPosSecondParameter (the second one is not necessary if IpPosSubModeSelect value is 0);
- if ModesOfOperationDisplay = 8 → TargetPosition and VelocityOffset (the second one is necessary only if CyclicSynchronousSubMode value is -147 or -148);
- if ModesOfOperationDisplay = 9 → TargetVelocity;
- if ModesOfOperationDisplay = 10 → TargetTorque;

The limits of the *parameters for the interpolation* related to alarm that corresponds to the bit5, according to the value of ModesOfOperationDisplay, mean that:

- in a $T_{SYNC}$ period the resultant position difference must be lower than 16.38 rev. The position difference is calculated according to the value that's written in the IpPosFirstParameter parameter if ModesOfOperationDisplay value is 7 or according to the value of TargetPosition if ModesOfOperationDisplay value is 8;
- the velocity must be lower than ±3216 rad/s. The velocity set in IpPosSecondParameter if ModesOfOperationDisplay value is 7, in VelocityOffset if ModesOfOperationDisplay value is 8, in TargetVelocity if ModesOfOperationDisplay value is 9;

With *synchronization method* is meant:

- with CANopen drive, see Section 9.3, *Synchronization object (SYNC)*
- with EtherCAT drive, see Section 10.4, *Synchronization*

| Bit | Name | Type | Description |
|---|---|---|---|
| 0 | CANopen not in operational state | F | The NMT state machine is not in the OPERATIONAL state (only in the versions CAN). This Fault is caused by:<br>• it has been required to change the state of the NMT state machine with the NMT service;<br>• the drive signals CAN communication error; analyse the error details. |
| | EtherCAT not in operational state | | The EtherCAT state machine is not in the OPERATIONAL state (only in the versions ETC). This Fault is caused by:<br>• it has been required to change the state to the EtherCAT state machine;<br>• the drive signals EtherCAT communication error; analyse the error details. |
| 1 | Pdo missing | F | The parameters for the interpolation were not received via PDO, before the synchronization; correctly manage |

| Bit | Name | Type | Description |
|---|---|---|---|
| | | | the PDOs in the master according to the synchronization method. |
| 2 | Incompatibility of cubic interpolation parameter | F | Cubic interpolation parameters not coherent. Check that the data that are being sent by the master are correct. |
| 3 | Wrong cubic interpolation cycle period | F | $T_{SYNC}$ too short. Increase the period according to the synchronization method and to the ModesOfOperationDisplay (For further details see what is reported in each single operative mode in the Chapter 22, *Creating a motion*). |
| 4 | Wrong interpolation cycle period | F | |
| 5 | Interpolation parameters out of range | F | Interpolation parameters out of allowed ranges. Check that the data that are being sent to the master are correct and respect the limits set in the drive. |
| 6 | Incompatibility interpolation sub mode | F | The CyclicSynchronousSubMode is not compatible with the ModesOfOperationDisplay. Check the settings according to the contents of Table 27.16. |
| 7 - 15 | Reserved | | |

***Table 24.6. Details about RealTimeModeError (W = Warning, F = Fault, W/F = both).***

## CAN communication error

NOTE

**The information in this paragraph are valid only for the drives version CAN.**

Main port communication error CANopen. The details can be found in the following charts and in the parameters CANopenEmcyRegister.

| Bit | Name | Type | Description |
|---|---|---|---|
| 0 | This bit is equal to 1 if at least one emergency has been sent. | | |
| 1 | CAN in error passive state | W | CAN in the Error-passive state state; analyze the causes by reading the group of parameters CANopenCounters. |
| 2 | Life guard error | F | Error of *Life Guard*. See Error control services; check the wiring. |
| 3 | Recovered from bus-off | W | Drive retrieved from the Bus-off state state; analyze the causes by reading the group of parameters CANopenCounters. |
| 4 | Sync controller error | F | Error in the SYNC controller. See Section 9.3, *Synchronization object (SYNC)*. |
| 5 - 7 | Reserved | | |
| 8 | Peripheral initialization error | W | Error during the hardware peripheral initialization phase; check the wirings. See Section 7.2.3, *Field bus (CAN)*. |
| 9 - 24 | Reserved | | |
| 25 | Message overrun: Node guard (RX) | W | CANopen overrun: overwritten NODEGUARD RX; manage a longer life time period with the master (see Error control services) or diminish the network load CANopen. |
| 26 | Message overrun: Node guard (TX) | W | CANopen overrun: overwritten NODEGUARD TX; manage a longer life time period with the master (see Error control services) or diminish the network load CANopen. |

| Bit | Name | Type | Description |
|---|---|---|---|
| 27 | Message overrun: SDO (RX) | W | CANopen overrun: overwritten SDO RX; diminish the network load CANopen. |
| 28 | Message overrun: SDO (TX) | W | CANopen overrun: overwritten SDO TX; diminish the network load CANopen. |
| 29 | Message overrun: EMCY | W | CANopen overrun: overwritten EMCY; diminish the network load CANopen. |
| 30 | Message overrun: SYNC | W | CANopen overrun: overwritten SYNC; manage a SYNC with the master with a longer CommunicCyclePeriod. |
| 31 | Message overrun: NMT | W | CANopen overrun: overwritten NMT. |

***Table 24.7. Details about CANopenEmcyService (W = Warning, F = Fault, W/F = both).***

| Bit | Name | Type | Description |
|---|---|---|---|
| 0 | Message overrun: PDO TX 1 | W | The PDO 1 TX has been overwritten:<br>• if the overrun comes from a PDO sending the results to the capture peripheral, increase the inhibit time of the related peripheral (see Chapter 18, *Capture Peripherals*);<br>• if the overrun comes from a generic PDO, increase the inhibit time of the PDO (see Section 27.23, *PDO managed by the port CANopen (10000-11999)*). |
| 1 | Message overrun: PDO TX 2 | W | The PDO 2 TX has been overwritten; see the description of the PDO TX 1. |
| 2 | Message overrun: PDO TX 3 | W | The PDO 3 TX has been overwritten; see the description of the PDO TX 1. |
| 3 | Message overrun: PDO TX 4 | W | The PDO 4 TX has been overwritten; see the description of the PDO TX 1. |
| 4 - 7 | Reserved | | |
| 8 | Message overrun: PDO RX 1 | W | The PDO 1 RX has been overwritten; slow the timing the PDO is sent from the producer of the PDO itself (the producer is the node sending the PDO to the network CANopen). |
| 9 | Message overrun: PDO RX 2 | W | The PDO 2 RX has been overwritten; see the description of the PDO RX 1. |
| 10 | Message overrun: PDO RX 3 | W | The PDO 3 RX has been overwritten; see the description of the PDO RX 1. |
| 11 | Message overrun: PDO RX 4 | W | The PDO 4 RX has been overwritten; see the description of the PDO RX 1. |
| 12 - 15 | Reserved | | |
| 16 | PDO RX 1 too short | W | PDO 1 RX with data field length too short: the sent data have no effect; verify that the PDOs that have been programmed in the master and in the slave are coherent. |
| 17 | PDO RX 2 too short | W | PDO 2 RX with data field length too short: the sent data have no effect; see the description of the PDO RX 1. |
| 18 | PDO RX 3 too short | W | PDO 3 RX with data field length too short: the sent data have no effect; see the description of the PDO RX 1. |
| 19 | PDO RX 4 too short | W | PDO 4 RX with data field length too short: the sent data have no effect; see the description of the PDO RX 1. |
| 20 - 23 | Reserved | | |
| 24 | PDO RX 1 too long | W | PDO 1 RX with data field length too long: verify that the PDOs that have been programmed in the master and in the slave are coherent. |

| Bit | Name | Type | Description |
|---|---|---|---|
| 25 | PDO RX 2 too long | W | PDO 2 RX with data field length too long: see the description of the PDO RX 1 |
| 26 | PDO RX 3 too long | W | PDO 3 RX with data field length too long: see the description of the PDO RX 1 |
| 27 | PDO RX 4 too long | W | PDO 4 RX with data field length too long: see the description of the PDO RX 1 |
| 28 - 31 | Reserved | | |

***Table 24.8. Details about CANopenEmcyProcess (W = Warning, F = Fault, W/F = both).***

## EtherCAT communication error

NOTE

**The information in this paragraph are valid only for the drive version ETC.**

Main port communication error EtherCAT. The details can be found in the following chart and in the parameters EtherCAT_Diagnostics.

| Bit | Name | Type | Description |
|---|---|---|---|
| 0 | Sync Manager watchdog expired | F | The watchdog of the Sync manager (SM) of the PDO RX expired; the PDO RX has not been received; manage correctly in the master the sending of the PDO RX or read the watchdog times in the registers of the ET1100. |
| 1 | Sync 0 watchdog expired | F | The watchdog of the Sync Signal 0 expired; set and enable correctly the signal Sync Signal 0 and the watchdog times in the registers of the ET1100. |
| 2 | PLL Error | F | PDO and Sync Signal 0 are not synchronized; manage correctly in the master the sending of the PDO before the synchronization; the synchronization methods can be found in Section 10.4, *Synchronization*. |
| 3 | Synchronization Error | F | The PDOs RX don't arrive or however not in correspondence to the set synchronization reference (see Section 10.4, *Synchronization*), within a tolerance from [Sync/2] to [Sync + Sync/2] with a maximum value of [Sync + 1ms]; verify that the PDOs RX are sent by the master in correspondence to the synchronization reference. |
| 4 - 7 | Reserved | | |
| 8 | Hardware failure | W | Serious error in ET1100; please contact Bonfiglioli Mechatronic Research S.p.a |
| 9 | ESI eeprom may not be updated | W | ESI eeprom not updated; update the ESI eeprom according to what's reported in Section 26.6, *ESI EEPROM updating procedure on ETC drive*. |
| 10 | ESI eeprom will be updated at the next power-up cycle | W | ESI eeprom not updated; in case the drive characteristics allow the ESI eeprom automatic update , when a firmware download ends then the drive is ready to execute it on next start-up. |
| 11 | ESI eeprom updating at the power-up failed | W | ESI eeprom not updated; the automatic procedure has failed due to a drive HW problem, execute the ESI eeprom manual update procedure according to what's reported in Section 26.6, *ESI EEPROM updating procedure on ETC drive*. |
| 10 - 31 | Reserved | | |

***Table 24.9. Details about EtcErrorRetentCommMsg (W = Warning, F = Fault, W/F = both).***

**Power or motor over current**

Too high and anomalous current in the power section or in the motor phases. The overcurrent values can be found in OverCurrentAValue, OverCurrentBValue and OverCurrentC-Value. It generally happens when the drive is not on condition to correctly control the current due to an anomaly or a non optimal parametrisation (tuning). If the alarm stands for a short time period and it's a Warning, it means that the overcurrent has lasted for a short time, not dangerous for the drive; if the alarm is a Fault it means that the overcurrent has a value and a duration such that the drive might damage Check the current loop tuning and decrease its dynamic response. Decrease the value of UserPeakCurrent. Check that the insulation voltage of the motor windings is compatible with the drive supply voltage.
This anomaly can happen even when ActualFieldCurrent < -50%IS and very steep decelerations are executed. In these conditions the DC bus supply voltage turns out to be not sufficient to control the current, the counter-electromotive force of the motor increases the currents to valules that are higher than the limit and the over current may be reported. If this condition continues, during the Warning presence, it can become a Fault. Decrease the deceleration ramps or the starting ramp velocity, increase the DC bus supply voltage.
Wait 20 seconds before to execute the Fault Reset, in order to allow the dissipation of the stored energy. If the problem persists, please contact Bonfiglioli.

**Position following error**

The PositionFollowingError exceeded the specified thresholds, according to what is reported in Error of position tracking: check if the motor motion is compatible with the settings. The Fault can be disabled by writing the FaultMaskEnable parameter; the Warning cannot be disabled.

**Last command requested failed**

The last command of the SysMngCommand has concluded with an error. The details and the solutions of the error can be found in the parameter SysMngError.

**/STO Management Error**

Error related to the /STO input management. The details are reported in this table and in the STOError parameters.

| Bit | Name | Type | Description |
|---|---|---|---|
| 0 | /STO = 0V with drive enabled | F | It occurs in the following cases:<br>• The drive is in Drive enable state and the voltage on the /STO digital input fails → Disable the drive before to cut off the voltage supply to the /STO input.<br>• It has been tried to enable the drive without the /STO signal → Provide voltage to the /STO input before to give the enabling command. |
| 1 | /STO input level not in valid range | F | The voltage level that's applied on the /STO input has lasted more than 500ms in the intermediate range of values of the voltage thresholds (see Figure 6.3). Be sure that the transition between the voltage levels, from the electrical point of view, takes no more than 500ms and that the voltage values are within the correct ranges (see /STO input electric features) |

| Bit | Name | Type | Description |
|---|---|---|---|
| 2 - 15 | Reserved | | |

***Table 24.10. Details about STOError (W = Warning, F = Fault, W/F = both).***

### User Fault

Error managed directly by the user. It can be useful when, in case of dangerous situations, the user thinks it is necessary to stop the drive working and report a Fault. The details can be found in the following chart and in the parameters UserError.

| Bit | Name | Type | Description |
|---|---|---|---|
| 0 | User Fault 1 | F | User fault number 1 |
| 1 - 15 | Reserved | | |

***Table 24.11. Details about UserError (W = Warning, F = Fault, W/F = both).***

### I2T limit reached

I2TValue has reached the 100%, that is the drive has reached the highest level of overload. If the Fault is active the drive is in Fault state, otherwise only the Warning is reported and the motor current reaches the value NominalCurrent. See what reported in Section 25.6, *Motion problems*.

### I2T Warning threshold reached

I2TValue reached the specified threshold in I2TWarningThreshold. The motor current is not limited. See what reported in Section 25.6, *Motion problems*.

### Limit reached

Limits reached by the motor motion (see Chapter 19, *Motion limits*). The details can be found in the following chart and in the parameter LimitReachedError.

| Bit | Name | Type | Description |
|---|---|---|---|
| 0 | Positive software position limit reached | W | PositionActualValue higher than PositionLimitPositive. |
| 1 | Negative software position limit reached | W | PositionActualValue lower than PositionLimitNegative. |
| 2 | Positive hardware position limit reached. | W | Positive hardware position limit reached Positive limit switch (FC +). If the position limit has not been activated by the machinery that's moved by the motor, verify that the switch has not been accidentally activated, that it's correctly powered, that the cable has not been cut, that's correctly connected with the connector CN4's digital inputs. |
| 3 | Negative hardware position limit reached | W | Negative hardware position limit reached Negative limit switch (FC -). If the position limit has not been activated by the machinery that's moved by the motor, verify that the switch has not been accidentally activated, that it's correctly powered, that the cable has not been cut, that's correctly connected with the connector CN4's digital inputs. |
| 4 - 7 | Reserved | | |

| Bit | Name | Type | Description |
|---|---|---|---|
| 8 | Max motor speed limit reached | W | VelocityActualValue limited by MaxMotorSpeed. |
| 9 - 11 | Reserved | | |
| 12 | Torque limit reached | W | TargetTorque higher or equal to ActualTorqueLimitP in absolute value. Verify that there are not mechanical obstacles; if the limit is set through analog input (in the TorqueLimitSelector parameter has been set the value 2), verify the input value. |
| 13 | Peak current is zero | W | UserPeakCurrent is equal to 0. Set UserPeakCurrent. |
| 14 | Peak current limit reached | W | ActualMotorCurrent limited by UserPeakCurrent. |
| 15 | Reserved | | |

***Table 24.12. Details about LimitReachedError (W = Warning, F = Fault, W/F = both).***

⚠ WARNING

**If the Polarity is of Reverse type, then the roles of Positive limit switch (FC +) and Negative limit switch (FC -) are reversed: Positive limit switch (FC +) behaves like Negative limit switch (FC -) and Negative limit switch (FC -) behaves like Positive limit switch (FC +). This is true both in the text of this manual and in the MotionDrive.**

### Possible no tuning of regulator

Because of the change of one or more motor parameters, of the sensor or of the power pwm, the regulation loops could not be correctly calibrated. It is advisable to check the loop calibration according to what is reported in Chapter 20, *System tuning*, or use one of the commands in Section 20.2, *Reset the tuning*.

### Drive is in disable state, since the enable input is or has been in not active state

The digital input, to which the Enable functionality is related, is set to 0 logic state and it is required the drive enabling (take it to the Drive enable state), or, while the drive is enabled, the input which is programmed with the Enable functionality is set to 0 (is deactivated). This error report is not active in the *Profile Velocity AI Mode* and *Torque AI Mode* modes.

### Feedback sensor error

An error related to a feedback position sensor malfunctioning has occurred. The details can be found in the following chart and in the parameters FeedbackSensorError.

| Code | Name | Type | Description |
|---|---|---|---|
| 0x159 | Phasing: Sensor code does not match | W | The phasing has been executed with a different sensor. If it's possible, repeat the phasing procedure, if not please contact Bonfiglioli Mechatronic Research S.p.a. |
| 0x15A | Phasing: Polarity of quadrature signals A or B is wrong | F | The encoder A and B quadrature signals polarity is not correct. Check the encoder connections. If the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |
| 0x15B | Phasing: Quadrature signals A or B are disconected | F | The drive does not receive the encoder A and B quadrature signals. Check the encoder connections. If the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |

| Code | Name | Type | Description |
|---|---|---|---|
| 0x30A | Phasing: Hall sensors status is not valid | F | The drive does not receive the Hall sensor signals. Check the encoder connections. If the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |
| 0x30B | Phasing: Validation window not respected | F | The alignment between the encoder and the motor position has failed. Check the encoder connections. If the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |
| 0x30C | Phasing: Hall sensors sequence is wrong | F | The Hall sensor signals don't respect the right sequence. Check the encoder connections. If the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |
| 0x359 | Phasing: Sensor code does not match | W | The phasing has been executed with a different sensor. If it's possible, repeat the phasing procedure, if not please contact Bonfiglioli Mechatronic Research S.p.a. |
| 0x35A | Phasing: Polarity of quadrature signals A or B is wrong | F | The encoder A and B quadrature signals polarity is not correct. Check the encoder connections. If the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |
| 0x35B | Phasing: Quadrature signals A or B are disconected | F | The drive does not receive the encoder A and B quadrature signals. Check the encoder connections. If the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |
| 0x457 | Phasing: Incremental counter initialization error | F | The alignment between the encoder and the motor position has failed. Check that, during this procedure, the motor is stopped and the encoder is correctly wired. If the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |
| 0x459 | Phasing: Sensor code does not match | W | The phasing has been executed with a different sensor. If it's possible, repeat the phasing procedure, if not please contact Bonfiglioli Mechatronic Research S.p.a. |
| 0x45A | Phasing: Polarity of Sine or Cosine is wrong | F | The encoder Sine and Cosine quadrature signals polarity is not correct. Check the encoder connections. If the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |
| 0x470 | Position calculation error: Mismatch between Sine/Cosine and Incremental counter | W/F | Electrical problems on the reconstruction of the position from the Sine/Cosine signals. Reset the errors. If the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |
| 0x471 | Sine or Cosine value error | W/F | Electrical problems on the reconstruction of the position from the Sine/Cosine signals. Check the encoder connections. Reset the errors. If the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |
| 0x472 | Phasing: Number of Sine/Cosine is incompatible with number of motor poles | F | The feedback sensor cannot be used for the feedback of the selected motor. Select another sensor so that the pole pairs of the motor is an integer multiple of the number of sensor sinusoids/revolution. |
| 0x473 | Sine or Cosine level out of range | W | Electrical problems on the reconstruction of the position from the Sine/Cosine signals. Check the encoder connections. Reset the errors and execute the Hard Reset command (command 5000 of the System Manager). If the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |
| 0x474 | Sine or Cosine hardware error | F | Drive internal electrical problems. Reset the errors and execute the Hard Reset command (command 5000 of the System Manager). If the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |

| Code | Name | Type | Description |
|---|---|---|---|
| 0x501<br>0x502<br>0x503<br>0x504<br>0x505<br>0x506<br>0x507<br>0x508<br>0x509<br>0x50A<br>0x50B<br>0x50C<br>0x50D<br>0x50E<br>0x50F<br>0x510<br>0x511<br>0x512<br>0x51F<br>0x520<br>0x521<br>0x522<br>0x523<br>0x51C<br>0x51E | Internal sensor error | F | Sensor internal error. If the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a |
| 0x530 | Communication: Timeout receiving data | F | Communication error with the sensor. Reset the errors. If the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |
| 0x531 | Communication: Timeout sending data | F | Communication error with the sensor. Reset the errors. If the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |
| 0x535 | Communication: Out of memory | F | Communication error with the sensor. Stop the oscilloscope and try again. If the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |
| 0x540 | Communication: Checksum error | F | Communication error with the sensor. Reset the errors. If the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |
| 0x541 | Communication: Parity error | F | Communication error with the sensor. Reset the errors. If the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |
| 0x542 | Communication: Framing error | F | Communication error with the sensor. Reset the errors. If the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |
| 0x543 | Communication: Overrun error | F | Communication error with the sensor. If the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |
| 0x557 | Phasing: Incremental counter initialization error | F | The alignment between the encoder and the motor position has failed. Check that, during this procedure, the motor is stopped and the encoder is correctly wired. If |

| Code | Name | Type | Description |
|---|---|---|---|
| | | | the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |
| 0x558 | Phasing: Data not found (sensor serial number does not match) | W | The phasing has been executed with a different sensor. If it's possible, repeat the phasing procedure, if not please contact Bonfiglioli Mechatronic Research S.p.a. |
| 0x559 | Phasing: Sensor code does not match | W | The phasing has been executed with a different sensor. If it's possible, repeat the phasing procedure, if not please contact Bonfiglioli Mechatronic Research S.p.a. |
| 0x55A | Phasing: Polarity of Sine or Cosine is wrong | F | The encoder Sine and Cosine quadrature signals polarity is not correct. Check the encoder connections. If the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |
| 0x570 | Position calculation error: Mismatch between analog and digital position | W/F | Electrical problems on the reconstruction of the position from analog and digital signals. Reset the errors. If the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |
| 0x571 | Sine or Cosine value error | W/F | Electrical problems on the reconstruction of the position from the Sine/Cosine signals. Check the encoder connections. Reset the errors. If the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |
| 0x573 | Sine or Cosine level out of range | W | Electrical problems on the reconstruction of the position from the Sine/Cosine signals. Check the encoder connections. Reset the errors and execute the Hard Reset command (command 5000 of the System Manager). If the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |
| 0x574 | Sine or Cosine hardware error | F | Drive internal electrical problems. Reset the errors and execute the Hard Reset command (command 5000 of the System Manager). If the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |
| 0x673 | Sine or Cosine level out of range | W | Electrical problems on the reconstruction of the position from the Sine/Cosine signals. Check the resolver connections. Reset the errors and execute the Hard Reset command (command 5000 of the System Manager). If the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |
| 0x674 | Sine or Cosine hardware error | F | Drive internal electrical problems. Reset the errors and execute the Hard Reset command (command 5000 of the System Manager). If the problem persists, please contact Bonfiglioli Mechatronic Research S.p.a. |

***Table 24.13. Details about FeedbackSensorError (W = Warning, F = Fault, W/F = both).***

## Digital IO configuration error

The configuration of the functionalities associated to the digital I/O is not correct. The details can be found in the following chart and in the parameters DigitalIoConfigError. The Warning becomes a Fault if you try to enable the drive.

| Bit | Name | Type | Description |
|---|---|---|---|
| 4-0 | Code of the first hardware resource involved in the error | - | 1 = I/O 0<br>2 = I/O 1<br>3 = I/O 2<br>4 = I/O 3<br>5 = In 4<br>6 = In 5<br>7 = In 6 |
| 9-5 | Code of the second hardware resource involved in the error | - | |

| Bit | Name | Type | Description |
|---|---|---|---|
| | | | 8 = In 7<br>9 = In 8<br>10 = In 9<br>11 = Out 4<br>12 = Out 5<br>13 = Out 6 |
| 16-10 | 1 = Exclusive function assigned to both resources | W/F | Functionality to be assigned to a hardware resource has been assigned to two resources; reprogram the functionalities; |
| | 2 = Step function assigned to this resource without Direction | | Step functionality assigned without having assigned the Dir one; assign the lacking functionality; |
| | 3 = Dir function assigned to this resource without Step | | Dir functionality assigned without having assigned the Step one; assign the lacking functionalities; |
| | 4 = Index function assigned to this resource without FA nor FB | | Quadrature Input Index (Idx) functionality assigned without having assigned the Quadrature Input ChA (Ch A) and Quadrature Input ChB (Ch B) ones; assign the lacking functionalities; |
| | 5 = FB function assigned to this resource without FA | | Quadrature Input ChB (Ch B) functionality assigned without having assigned the Quadrature Input ChA (Ch A) one; assign the lacking functionality; |
| | 6 = FA function assigned to this resource without FB | | Quadrature Input ChA (Ch A) functionality assigned without having assigned the Quadrature Input ChB (Ch B) one; assign the lacking functionality; |
| | 7 = Settings in permanent memory not compatible with firmware (default value restored) | | The I/Os setting in the permanent memory is not compatible with the firmware in the drive, so the default values have been automatically restored; Reconfigure the I/Os with the functionalities that are allowed by the actual firmware or restore the firmware that allowed the use of the functionalities that are no more at disposal. |

***Table 24.14. Details about DigitalIoConfigError (W = Warning, F = Fault, W/F = both).***

### Logic voltage error

Error of the power supply voltage of the control section. For any further information, see Section 14.2, *Supply voltages*. The details can be found in the following chart and in the parameters LogicVoltageError.

| Bit | Name | Type | Description |
|---|---|---|---|
| 0 | Logic voltage too low for brake | W/F | The supply voltage of the control section is too low to correctly ensure the brake. Under the "Error threshold for the brake" (see Chapter 5, *Technical features*) the drive enters in the Warning status. If the voltage remains under this threshold, the Fault activates. Increase the supply voltage or stabilize it. |
| 1 - 15 | Reserved | | |

***Table 24.15. Details about LogicVoltageError (W = Warning, F = Fault, W/F = both).***

### Motion parameter limited

One or more motion parameters are set above their own limits. The details can be found in the following chart and in the parameter MotionParamLimitedError.

| Bit | Name | Type | Description |
|---|---|---|---|
| 0 | Target velocity limited | W | With the *Profile Velocity Mode (CiA402)* (CiA402 and CUSTOM) operative modes, TargetVelocity is greater or equal to MaxMotorSpeed or to MaxProfileVelocity. Decrease TargetVelocity.<br>With ther *Profile Velocity AI Mode* mode, the conversion from AI0FilteredVoltage to the required velocity, produces a velocity value tha's grater or equal to MaxMotorSpeed or to MaxProfileVelocity. If the Warning is unexpected, check if the conversion procedure accords to the contents of Section 17.3, *Conversion*. |
| 1 | Profile velocity limited | W | ProfileVelocity is higher or equal to MaxMotorSpeed or to MaxProfileVelocity. Decrease ProfileVelocity. |
| 2 | Start velocity limited | W | StartVelocity is higher or equal to MaxMotorSpeed or to MaxProfileVelocity. Decrease StartVelocity. |
| 3 | End velocity limited | W | EndVelocity is higher or equal to MaxMotorSpeed or to MaxProfileVelocity. Decrease EndVelocity. |
| 4 | Speed during search for switch limited | W | SpeedForSwitch is higher or equal to MaxMotorSpeed or to MaxProfileVelocity. Decrease SpeedForSwitch. |
| 5 | Speed during search for zero limited | W | SpeedForZero is higher or equal to MaxMotorSpeed or to MaxProfileVelocity. Decrease SpeedForZero. |
| 6 | Velocity of the safety profile limited | W | SafetyPrfVelocity is higher or equal to MaxMotorSpeed or to MaxProfileVelocity. Decrease SafetyPrfVelocity. |
| 7 | Reserved | | |
| 8 | Profile acceleration limited | W | ProfileAcceceleration is higher or equal to MaxAcceleration. Decrease ProfileAcceceleration. |
| 9 | Profile deceleration limited | W | ProfileDeceleration is higher or equal to MaxDeceleration. Decrease ProfileDeceleration. |
| 10 | Homing acceleration limited | W | HomingAcceleration is higher or equal to MaxAcceleration or to MaxDeceleration. Decrease HomingAcceleration. |
| 11 | Quick stop deceleration limited | W | QuickStopDeceleration is higher or equal to MaxDeceleration. Decrease QuickStopDeceleration. |
| 12 | Deceleration of MC_Stop/ MC_Emcy function block limited | W | SafetyPrfAcceleration is higher or equal to MaxAcceleration. Decrease SafetyPrfAcceleration. |
| 13 | Acceleration of the safety profile limited | W | SafetyPrfAcceleration is higher or equal to MaxAcceleration. Decrease SafetyPrfAcceleration. |
| 14 | Deceleration of the safety profile limited | W | SafetyPrfDeceleration is higher or equal to MaxDeceleration. Decrease SafetyPrfDeceleration. |
| 15 | Reserved | | |

***Table 24.16. Details about MotionParamLimitedError (W = Warning, F = Fault, W/F = both).***

### Digital output overtemperature or overload

Overcharge or overtemperature have been detected on the drive internal management circuit of the digital output. All digital outputs are switched off. Check the connected wirings and loads.

### Over Speed

Maximum velocity limit exceeded. The threshold value is MaxMotorSpeed*1.2, therefore it's 20% over the maximum velocity that the motor can reach. If VelocityActualValue continuously remains over this threshold for 10ms, the system enters in Fault state because the movement is no more under control.

### Internal Error

A firmware internal error has occurred. The details can be found in the following chart and in the parameters InternalError. Please contact Bonfiglioli

| Bit | Name | Type | Description |
|---|---|---|---|
| 0 | Internal Software Reset | F | A firmware internal error has occurred. Report to Bonfiglioli the codes in the SwResetCode and SwResetInfo parameters. |

***Table 24.17. Details about InternalError (W = Warning, F = Fault, W/F = both).***

### /STO Error

An error on the /STO has occurred. The details are reported in the STOError parameters.

# Chapter 25 Troubleshooting

**ADVICE**

**In case of problems run the Save/Export parameters file command with MotionDrive, so that you can record on a file the full drive situation. The saved file is useful to run in a second moment or remotely the problem analysis.**

## 25.1. Generic problems

| Problem | Solution |
|---|---|
| The leds are switched off. | • Verify that the drive is correctly supplied, in particular the control section; see what's reported in Section 14.2, *Supply voltages*;<br>• check the wiring. |
| The leds are on but the drive does not communicate. | • See what reported in Section 25.5, *Communication problems*. |
| Interpreting the drive status on the basis of the leds status | • See what reported in Section 7.3, *Leds*. |
| Find the status of the Digital I/O | • Read the parameters DigitalInputs and PhysicalOutputs;<br>• open the tab Main of MotionDrive<br>• see what reported in Chapter 16, *Digital inputs and outputs*. |

| Problem | Solution |
|---|---|
| Value the motion performances | • See the available parameters in Show variables (Figure 3.1) for one first evaluation;<br>• use the MotionDrive oscilloscope to estimate the performances of the motion parameters (Section 23.2, *Monitoring by oscilloscope*). |
| Monitor some drive parameters (temperature, currents, speed, etc...) | • See what reported in Chapter 23, *Oscilloscope and monitoring*;<br>• see what reported in Section 27.12, *Drive status (1800-1999)*. |
| Load dynamic braking | • Do not use the internal brake of the *i*BMD drives;<br>• run the braking according to what is reported in Section 22.3, *Carrying out a stop by using the master*;<br>• consider what is reported in Section 14.3, *Regeneration*. |
| Delay the brake | • The brake is managed automatically by the drive and it cannot be enabled with Drive enable;<br>• with Drive disable the brake can be delayed as described in Section 15.6, *Brake*. Pay attention, in this case the load can move in an unforeseen way. |
| The internal brake does not keep the motor steady | • If the load applies a higher torque on the brake torque, it is necessary to apply more efficient braking systems;<br>• the internal brake is damaged; it could be necessary to repeat the brake run-in. Please contact Bonfiglioli Mechatronic Research S.p.a. |
| Use the capture units. | • See what reported in Chapter 18, *Capture Peripherals*. |

## 25.2. Electric and connection problems

| Problem | Solution |
|---|---|
| Which is the reference voltage (0V) of the digital inputs and outputs | • The voltages of the digital inputs and outputs refer to Ground Control supply of CN5 (PIN B). |

## 25.3. Problems with Fault and Warning

| Problem | Solution |
|---|---|
| The drive is in Fault: how to proceed | • Interpret the present Fault precisely (FaultDynamic, Section 24.1, *Management of the errors with MotionDrive* or Section 24.2, *Monitoring the errors on the status leds*);<br>• analyse the Fault type and its possible causes (Section 24.6, *Errors description*);<br>• eliminate the causes that have caused the Fault and run the reset (Section 24.5, *Resetting the errors*). |
| It is not possible to eliminate the Faults | • Ready carefully what is reported in Section 24.5, *Resetting the errors*. |
| How to remove the Fault causes | • Analyse the occurred Faults by using what is reported in Section 24.6, *Errors description*. |
| How to stop the motor in case of Fault | • In case of Fault, the drive runs what is reported in Section 24.4, *Reaction to the Faults*. In some cases it is not possible to check the motor stop or to run a Safety profile. |
| Difference between a dynamic and a retentive Fault | • Dynamic error: the error condition is still in the drive;<br>• retention error: the error is memorized by the drive, until it is reset. |

| Problem | Solution |
|---|---|
| The drive goes in Fault when it is enabled | • Analyse the Fault that's present after the enabling. |
| Enabling the drive while some Warnings are active | • In general the drive can be enabled also with active Warnings;<br>• pay attention: some Warnings become Faults if you try to enable the drive;<br>• it is advisable to eliminate any present Warning. |
| Difference between Parameters serious error and Parameters soft error | • The serious error is a more serious Warning and it becomes a Fault if you try to enable the drive. The soft error does not compromise the drive integrity, that is the drive protects itself. In any case you need to analyse the error type accurately. |

## 25.4. Problems with parameters and configuration

| Problem | Solution |
|---|---|
| How parametrizing the drive | • Use the parameters vocabulary (Section 27.2, *Reading and writing a parameter*);<br>• use Drive Setup of MotionDrive (Toolbar > );<br>• download a parameters file (Section 21.3, *Downloading parameters file*). |
| Restoring a known configuration | • Run the command Restore default parameters (command 2200 of the System Manager): updating the current configuration and the permanent memory with the default values;<br>• run the command Reset to default (command 2201 of the System Manager): updating the current configuration with the default values;<br>• run the command Reload all parameters (command 2300 of the System Manager ): updating the current configuration with the data in the permanent memory;<br>• run the command Hard reset (command 5000 of the System Manager): it means, for all parameters, to switch on and off the drive;<br>• run the command Soft reset (command 5001 of the System Manager): it means, for all parameters, except from the position ones, to switch on and off the drive. |
| The parameters are not kept | • Run the command Save all parameters (command 2001 of the System Manager): saving the current configuration in the drive permanent memory;<br>• check if the parameters are not written in the internal PLC programme; try to cancel the programme;<br>• check if the parameters are not written by the network master; disconnect the field buses;<br>• check if there are no parametrization or System Manager errors at the firmware start. |
| How to manage the drive parametrization | • See what reported in Chapter 21, *Saving, restoring or cloning the drive configuration.*. |
| Modifying a parameters file | • Open a MotionDrive session and connect OFFLINE choosing the parameters file to modify (Section 12.2, *Offline mode*);<br>• at the end of the modifications, save the modified parameters file. |
| Moving the parameters configuration between one drive and another | • Use the parameters file (Section 21.2, *Saving/Exporting parameters file*) only between drives with the same ProductCode and HardwareProductCode. |
| In the tab Motor and drive you cannot find the motor you want to use | • If the motor doesn't appear in the combo box Motor, the motor database has to be updated (Section 26.2, *Updating the motors database*);<br>• in the *i*BMD drives the motor cannot be changed. |
| In the tab Motor and drive you cannot find the FeedbackSensorCode you want to use | • if you cannot find the sensor in the list of the FeedbackSensorCode, check the available sensors with the installed firmware and if necessary update the firmware (Section 26.3, *Firmware updating*);<br>• if the sensor is not supported by any available firmware for the drive, contact Bonfiglioli Mechatronic Research S.p.a; |

| Problem | Solution |
|---|---|
| | • in the *i*BMD drives, FeedbackSensorCode cannot be changed. |
| The parameters to configure the capture units are not writable | • the configuration parameters are not writable if the capture units are enabled. Check the status of the capture units through the parameter CaptureUnitCommand_A (CaptureUnitCommand_B). |

# 25.5. Communication problems

| Problem | Solution |
|---|---|
| The drive does not communicate via Modbus | • Connect the drive and see the connection settings according to what reported in Section 8.3, *Communicate with master Modbus RS232 (auxiliary communication port)*;<br>• check if the drive is on: the L1 and L2 leds (Figure 7.10) must not be off. |
| The drive does not communicate via CANopen | • Connect the drive and see the connection settings according to what reported in Section 8.1, *Communicate with master CANopen*;<br>• check if the drive is on: the L1 and L2 leds (Figure 7.10) must not be off;<br>• check if the led LA is on and the CAN communication error is not enabled; if necessary remove the error. |
| The drive does not communicate via EtherCAT | • Connect the drive and see the connection settings according to what reported in Section 8.2, *Communicate with master EtherCAT*;<br>• check if the drive is on: the L1 and L2 leds (Figure 7.10) must not be off;<br>• check the presence of the link in the communication ports connected to the network EtherCAT: the related leds L/A 0 and L/A 1 must not be off (Figure 7.10);<br>• check if the led ERR is not on and the EtherCAT communication error is not enabled; if necessary remove the error. |
| Updating request of the MotionDrive Configuration files | • Updating according to what is reported in Section 26.5, *Updating the Configuration File*. |
| Reading/Writing parameters | • See Section 8.4, *Errors in reading / writing of the parameters*. |
| Firmware download procedure goes in error. | • Verify that the communication with the drive is active, verify the messages that are reported in the FirmwareStatus and SysMngError parameters and follow the suggested instructions. |

# 25.6. Motion problems

| Problem | Solution |
|---|---|
| How to enable the drive | • See what reported in Section 22.1, *Enabling by using the master*. |
| How to stop the load in safety | • See what reported in Section 22.3, *Carrying out a stop by using the master*. |
| How to measure the drive load level | • Monitoring the regular drive temperatures with the parameters of the group TemperatureStatus;<br>• monitoring the working of TargetTorque;<br>• monitoring the working of RMSMotorCurrent, after having set RMSMotorCurrentFilter equal to the value of the drive working time cycle. |
| The motor does not run the requested motion and the drive signals I2T limit reached or I2T | • Check the working of I2TValue and check the problem in the following line. |

| Problem | Solution |
|---|---|
| Warning threshold reached | |
| The drive signals I2T limit reached or I2T Warning threshold reached | • Monitoring the value of I2TValue and ActualMotorCurrent to find any anomalies;<br>• check the mechanics working to avoid any anomalous absorptions of ActualTorque;<br>• decrease the requested performances and increase the dwell times during which the motor works at low current in order to download I2TValue;<br>• decrease the acceleration ramps and the requested speeds to decrease the TargetTorque;<br>• increase UserMaxI2T till the value of DriveMaxI2T (Section 14.6, *I2T*). |
| The motor does not run the requested motion or the requested motion has been interrupted | • Check for any possible stop command (see Section 22.3, *Carrying out a stop by using the master*);<br>• check if the drive is in Fault (see Chapter 24, *Fault and Warning*);<br>• check for any possible disabling command (see Section 22.2, *Disabling by using the master*);<br>• check if the Limit reached is not enabled;<br>• check if the drive has not run an on-the-fly mode change; in this case the drive is used to insert the new set points according to the new selected operating mode (see Section 22.5, *On-the-fly operative mode change*);<br>• if you work in *Interpolated Position Mode* check if the bit bit *Enable ip mode* of the Controlword has not been reset;<br>• if you work in *Profile Velocity Mode (CiA402)* or *Profile Velocity Mode (CUSTOM)*, check if the absolute value of TargetVelocity is higher than EndVelocity and StartVelocity. |
| the PositionFollowingError doesn't resets during a positioning, while the velocity is constant | • Verificare che l'asse sia tarato correttamente (vedere Chapter 20, *System tuning*) e aumentare la dinamica dei loop per far rientrare più velocemente PositionFollowingError;<br>• Check that the torque/velocity or position limits are not reached (see Chapter 19, *Motion limits*);<br>• If the *Interpolated Position Mode* is used with IpPosSubModeSelect = -10: check that at every variation of RequestedPosition, the FeedForwardSpeed takes the correspondent value (see Figure 22.5 and Figure 22.6);<br>• check that the value of the KVff parameter is 1000; |

# 25.7. Problems with the oscilloscope

| Problem | Solution |
|---|---|
| How to do a trial capture | • Choose at least one parameter to capture from the channels list (point A in Section 23.2, *Monitoring by oscilloscope*);<br>• check if the channel (point K) is enabled and if the display mode is Normal (point H);<br>• select Trigger mode Auto (point B in Section 23.3, *Setting the oscilloscope Trigger*), Sampling time lower than 1ms (point C) and the option Single trigger mode (point G);<br>• press Run (point I) and wait for some seconds for the upload to be finished;<br>• press Autoscale in the tab Channel. |
| The oscilloscope does not capture any data | • No channel has been selected in the tab Channel; select at least one parameter to capture and check if the channel is enabled (point K in Section 23.2, *Monitoring by oscilloscope*);<br>• the trigger has been stopped; the oscilloscope status is Trigger stopped; press Run;<br>• check if the trigger is correctly set; if necessary try with Trigger mode Auto (point B in Section 23.3, *Setting the oscilloscope Trigger*). |
| The oscilloscope loads the data repeatedly | • Select the option Single trigger mode (point G in Section 23.3, *Setting the oscilloscope Trigger*). |
| There is no trigger event | • Check if the parameter the trigger is associated to can cause the event: try with Trigger mode Auto (point B in Section 23.3, *Setting the oscilloscope Trigger*); |

| Problem | Solution |
|---|---|
| | • check if the trigger is associated to the wanted channel (point A);<br>• check if Trigger value and Trigger edge are the wanted ones; enable the option Show trigger for confirmation (point F). |
| The progress of the oscilloscope status bar is too slow or has stopped (point N in Section 23.3, *Setting the oscilloscope Trigger*) | • The Sampling time (point C) is too high (> 10ms) and a large time window will be captured; try to decrease the Sampling time;<br>• the communication does not allow higher uploading speeds: wait for the procedure end or analyse any possible communication problems. |
| The captured time interval is not sufficient or the available track is concentrated in a limited portion of captured data | • Modifying the Sampling time: increasing it to increase the time window to capture; decrease it to concentrate the samples in a lower time interval;<br>• zooming to enlarge the tracks (point D in Measures). |
| The first part of the profiles is not captured | • Select correctly the pre-trigger to capture one part of the profiles before the trigger event (point E in Section 23.3, *Setting the oscilloscope Trigger*). |
| The tracks are not sufficiently resolute | • Disable the useless channels (deselect the option at the point K in Section 23.2, *Monitoring by oscilloscope*);<br>• decrease the Sampling time. |
| There are no traces in the Screen | • Run a capture;<br>• press Autoscale in the tab Channel;<br>• modify scale and channels offset (Section 23.2, *Monitoring by oscilloscope*);<br>• centre the tracks in the Screen (point E in Section 23.2, *Monitoring by oscilloscope*);<br>• select Normal in the display mode (point H). |
| How to measure the captured profiles | • Run a capture;<br>• insert the scales in the screen (point N in Tracks preferences);<br>• see what reported in Measures. |
| How to compare the next captures | • Save in the Screen the first set of tracks (Section 23.4, *Saving or uploading an oscilloscope capture*);<br>• run the second capture for the comparison. |
| How to running motions by the Function Generator | • Select the desired Function Generator in the Function Generator tab;<br>• press Load presets;<br>• in order to avoid unwanted motions or collisions, take all the necessary precautions and configure precisely the drive limits (Chapter 19, *Motion limits*);<br>• press Start and wait for some seconds for the upload of the oscilloscope data to end;<br>• check any possible error in the field Drive information. |
| The Function Generator does not start | • read and interpret the errors in the field Drive information;<br>• the internal PLC interferes with the Function Generator: stop the PLC;<br>• the network master interferes with the Function Generator: disconnect the field buses not used or interrupt the parameters writings;<br>• the drive is in Fault: press Show errors;<br>• press End and try again. |
| The Function Generator must be promptly stopped | • Select properly the parameters for the End command (Tuning end option and Tuning end deceleration);<br>• Press End. |
| The drive must not be disabled once the Function Generator is finished. | • Select for Tuning end option, Deceleration ramp and enable or Zero speed and enable;<br>• never press Disable to stop the Function Generator. |
| The units of measurement of the references are not suit- | • Modify the MotionDrive units of measurement (Section 13.4, *MotionDrive units*). |

| Problem | Solution |
|---|---|
| able for the foreseen motion | |

# 25.8. Tuning problems

| Problem | Solution |
|---|---|
| Difficulties on the visualization of the results on the oscilloscope charts | • See Section 25.7, *Problems with the oscilloscope*. |
| Resonances in VelocityActualValue and in ActualTorqueCurrent | • Decrease the dynamic performances by lowering the DynamicResponse option<br>• For the drives of different type as *i*BMD, check and improve the motor and feedback sensor wirings by following carefully what is reported in Section 7.2, *Electrical connections*;<br>• for the drives of different type as *i*BMD, use a feedback sensor with higher resolution;<br>• decrease the value of KCp_Q and KCp_D; bearing in mind that CurrentLoopEstimatedBandwidth decreases, such as the performances which can be obtained also from the VelocityLoop;<br>• modify the filtering action of the *Sensor filter* in the *VelocityLoop* (VFilterSensorFrequency);<br>• reduce the *VelocityLoop* band by decreasing KVp. |
| Resonances found at the end of the estimate of the inertia moment (Section 20.9, *Inertia estimator*) or when the motor is stopped or at the presence of pure inertial load (with low friction) coupled with less rigid joints or mechanical backlashes. | • Stabilize the system as reported in 3. System stabilisation and 4. Filters of the *Detailed Tuning Guide*;<br>• see the solutions in the next line. |
| Resonances in VelocityActualValue using mechanical transmissions with considerable backlashes and tolerances (mechanical transmission at limited performances) | • Make the load more solid through the mechanical transmission;<br>• improve the mechanical transmission: reduce backlashes, check the chains and the drive belts tension, check the joints, pulleys and gear wheels closure, use transmission components with no backlash, etc...<br>• tighten the mechanical structure and the machine chassis;<br>• decrease PositionLoopEstimatedBandwidth by decreasing KPp;<br>• reduce VelocityLoopEstimatedBandwidth by proportionally reducing KVp and KVi;<br>• if the resonances have a low speed, decrease PositionStandStill and VelocityStandStill<br>• in case of position control, increase the value of PositionErrorDeadBand. |
| Resonances in VelocityActualValue | • Repeat the tuning and follow the instructions that are reported in Section 20.5, *Detailed tuning guide* |
| Resonances at low speeds or at stopped motor | • Decrease the PositionStandStill and VelocityStandStill bars. |
| EstimatedInertia higher than 8-10Jm | • If PositionLoopEstimatedBandwidth is not at least the half of VelocityLoopEstimatedBandwidth, decrease KPp and PositionStandStill;<br>• use motors with higher inertia moment and mechanical transmissions with no backlash and with high stiffness coupling among the components<br>• increase the reduction ratio of the mechanical transmission in order to reduce the load inertia moment on the part of the motor |

| Problem | Solution |
|---|---|
| | • reduce the accelerations and decelerations in accordance with the available torque in order to avoid oscillations at the ramp end<br>• decrease KVi by 20-50 units<br>• modify the outgoing filters of the speed regulator. |

# Chapter 26
# Software updating

In the *i*BMD drives, during the functioning of the boot software and of the firmware startup, the leds don't work as the standard functioning (started firmware, referring to the Table 7.6). In the following table the various leds statuses are reported with a short description.

Every cell that indicates the leds configuration contains their colour and blink type, separated by a comma.
The codes for the colours are made of one letter:

- *G*: green;
- *O*: orange;
- *R* red;
- *x* unimportant (in case of led off);

The blinking codes are the same that are reported in Section 7.3, *Leds*.

| Description | FirmwareStatus | L1 | L2 | L3 | L4 | L5 | L6 |
|---|---|---|---|---|---|---|---|
| Boot startup | - | x, OFF | x, OFF | O, ON | O, ON | x, OFF | x, OFF |
| Firmware startup | - | x, OFF | x, OFF | G, ON | G, ON | x, OFF | x, OFF |
| Firmware download phases | - | O, 1 FL | O, 1 FL | O, ON | O, ON | O, 1 FL | O, 1 FL |
| | | O, BLK | O, BLK | O, ON | O, ON | O, BLK | O, BLK |
| | | O, ON | O, ON | O, ON | O, ON | O, ON | O, ON |
| | | O, FLK | O, FLK | O, ON | O, ON | O, FLK | O, FLK |

<table>
<tr><th>Description</th><th>FirmwareStatus</th><th>L1</th><th>L2</th><th>L3</th><th>L4</th><th>L5</th><th>L6</th></tr>
<tr><td rowspan="2"></td><td rowspan="2"></td><td>G, BLK</td><td>G, BLK</td><td>O, ON</td><td>O, ON</td><td>G, BLK</td><td>G, BLK</td></tr>
<tr><td>R, ON</td><td>R, ON</td><td>R, ON</td><td>R, ON</td><td>R, ON</td><td>R, ON</td></tr>
<tr><td>Firmware exception error</td><td>20</td><td rowspan="2">R, ON</td><td rowspan="2">x, OFF</td><td rowspan="2">R, ON</td><td rowspan="2">R, ON</td><td rowspan="2">R, BLK</td><td rowspan="2">R, BLK</td></tr>
<tr><td>Error during the Firmware download or corrupted Firmware flash memory</td><td>13, ≥100</td></tr>
<tr><td>CPLD programming aborted due to an error</td><td>19</td><td>R, ON</td><td>x, OFF</td><td>R, ON</td><td>R, ON</td><td>R, BLK</td><td>R, BLK</td></tr>
<tr><td>One or more incompatibility between boot, hw and fw</td><td>13, 16, 17, 18</td><td>G, BLK</td><td>x, OFF</td><td>R, ON</td><td>R, ON</td><td>R, BLK</td><td>R, BLK</td></tr>
</table>

***Table 26.1. Description of the leds during the boot and the firmware startup.***

## 26.1. MotionDrive updating

To update MotionDrive you only need to install the updated version of the programme by choosing one of the procedures proposed in Section 3.2, *Installation*.

## 26.2. Updating the motors database

The motors database is a file that's released by Bonfiglioli Mechatronic Research S.p.a and contains the data of the standard motors for the *i*BMD drive series. The motor database is used by MotionDrive to recognize the motor type that's wired to the drive with which you are connected and it's important to keep it updated.

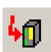
To update the motor database of MotionDrive it's sufficient to the MotionDrive program (see Section 3.2, *Installation*)

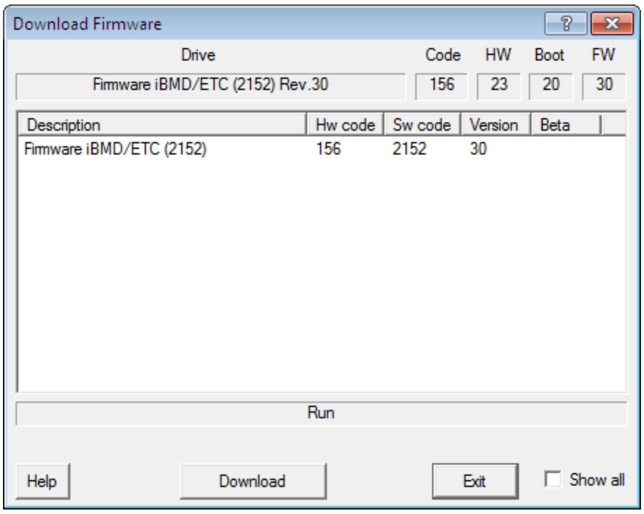
## 26.3. Firmware updating

IMPORTANT

**The firmware updating does not cancel any data saved in the permanent memory.**

To update the firmware, connect the drive with MotionDrive and open the Download firmware window. Access with MotionDrive:

> **Main menu** > **Drive** > **Download firmware ...**
>
> **Toolbar** > 




In the Download firmware window, choose the desired firmware and press Download. If the firmware is not present in the proposed list it's sufficient to install the updated version of MotionDrive by selecting one of the procedures that are proposed Section 3.2, *Installation*.

**ADVICE**

**If at the end of the download, the firmware does not start up, check what reported in the window and in the FirmwareStatus parameter.**

## 26.4. Updating the boot

**⚠ NOTICE**

**If during the boot updating the control section runs out of power, the drive cannot be used anymore and must be sent back to Bonfiglioli Mechatronic Research S.p.a. During the updating, supply power to the drive with an uninterruptible power supply (UPS).**

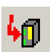
To update the boot, connect to the drive with MotionDrive and open the Download firmware window.

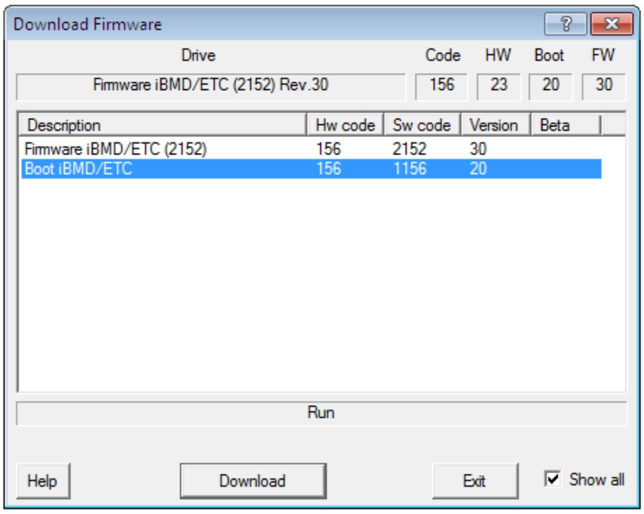
**IMPORTANT**

**Update the boot software only if really necessary.**

Access with MotionDrive:

**Main menu** > **Drive** > **Download firmware ...**

**Toolbar** > 




Activate the Show all option in the download firmware window, select the desired boot and push Download. If the boot is not present in the proposed list it's sufficient to install the updated version of MotionDrive by selecting one of the procedures that are proposed Section 3.2, *Installation*.

NOTE

**After downloading the boot it is necessary to download the firmware again. The data saved in the permanent memory are not cancelled.**

## 26.5. Updating the Configuration File

The Configuration files are files xml used by MotionDrive to communicate with the drive.

To update the Configuration files you only need to install the updated version of the MotionDrive by selecting one of the procedures that are proposed in Section 3.2, *Installation*.

## 26.6. ESI EEPROM updating procedure on ETC drive

### 26.6.1. ESI EEPROM updating from MotionDrive through debug serial port

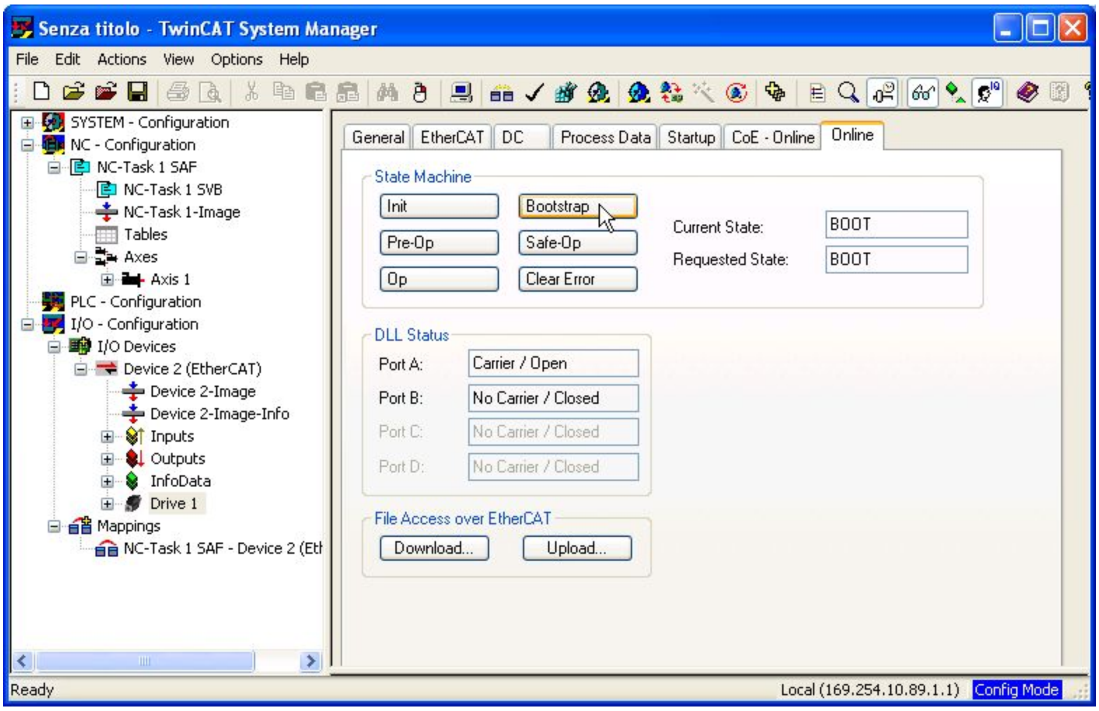
The procedure is commanded/executed by MotionDrive that connects to the drive through the debug serial port.

The EtherCAT master (e.g. TwinCAT Beckhoff) must be connected to the drive through the EtherCAT port, it is assumed to have already installed on the PC the EtherCAT master development environment and to have a project that describes all the peripherals in the EtherCAT network.

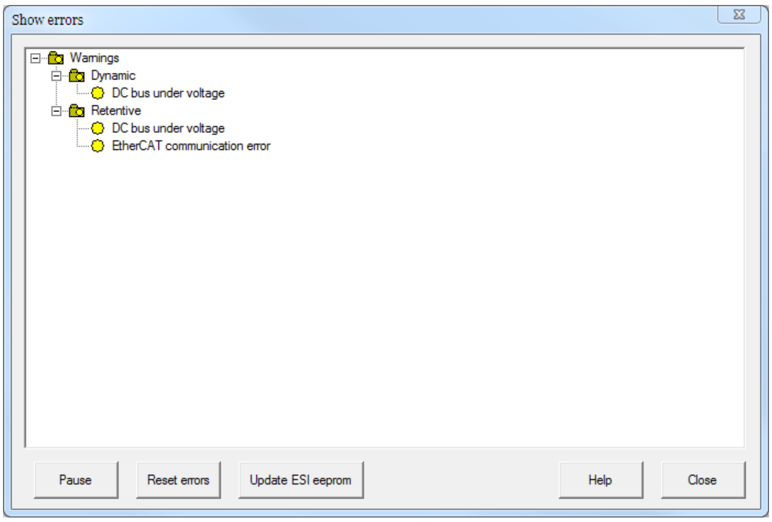
The sequence is the following:

1. start the EtherCAT master development environment and connect to the drive

2. take the EtherCAT status of the drive to BOOTSTRAP




3. start MotionDrive and connect to the drive through the debug serial port
4. on MotionDrive select the "Show Error" button and push the "Update esi eeprom" button




5. at the and of the operation check that the "ESI eeprom may not be updated" Warning is no more present
6. execute a power-up cycle on the drive (in case there are more than one drive to update, this operation can be executed only once at the end of the sequence)

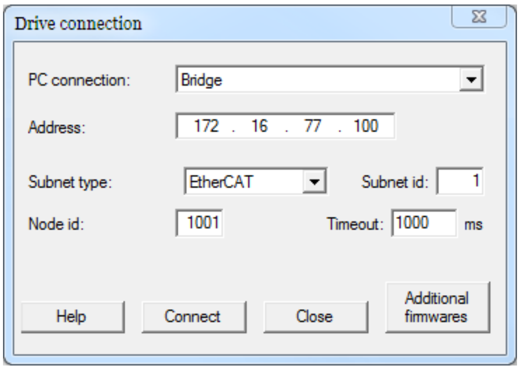
## 26.6.2. ESI EEPROM updating from MotionDrive through gateway CoDeSys

The procedure is commanded/executed by MotionDrive that connects to the drive through the Ethernet port to a EtherCAT CoDeSys master..

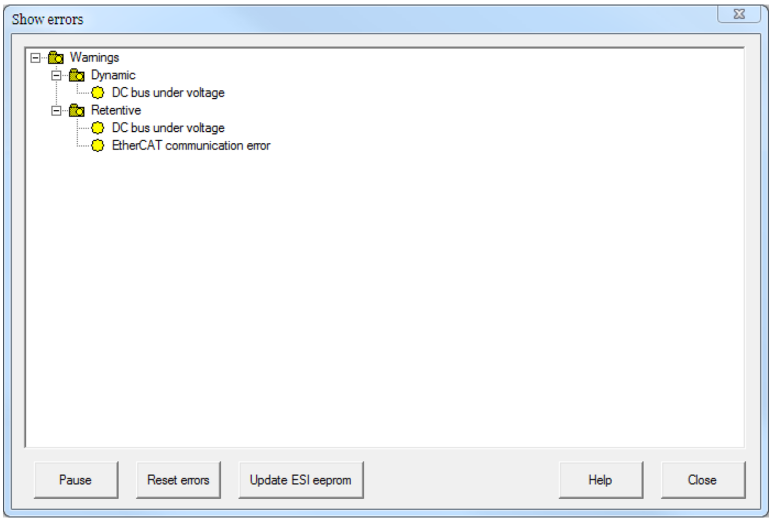
The EtherCAT CoDeSys master must be connected to the drive through the EtherCAT port.

The sequence is the following:

1. start MotionDrive and connect to the drive through the CoDeSys master




2. on MotionDrive select the "Show Error" button and push the "Update esi eeprom" button




3. at the and of the operation check that the "ESI eeprom may not be updated" Warning is no more present
4. execute a power-up cycle on the drive (in case there are more than one drive to update, this operation can be executed only once at the end of the sequence)

## 26.6.3. ESI EEPROM updating from EtherCAT master in manual mode

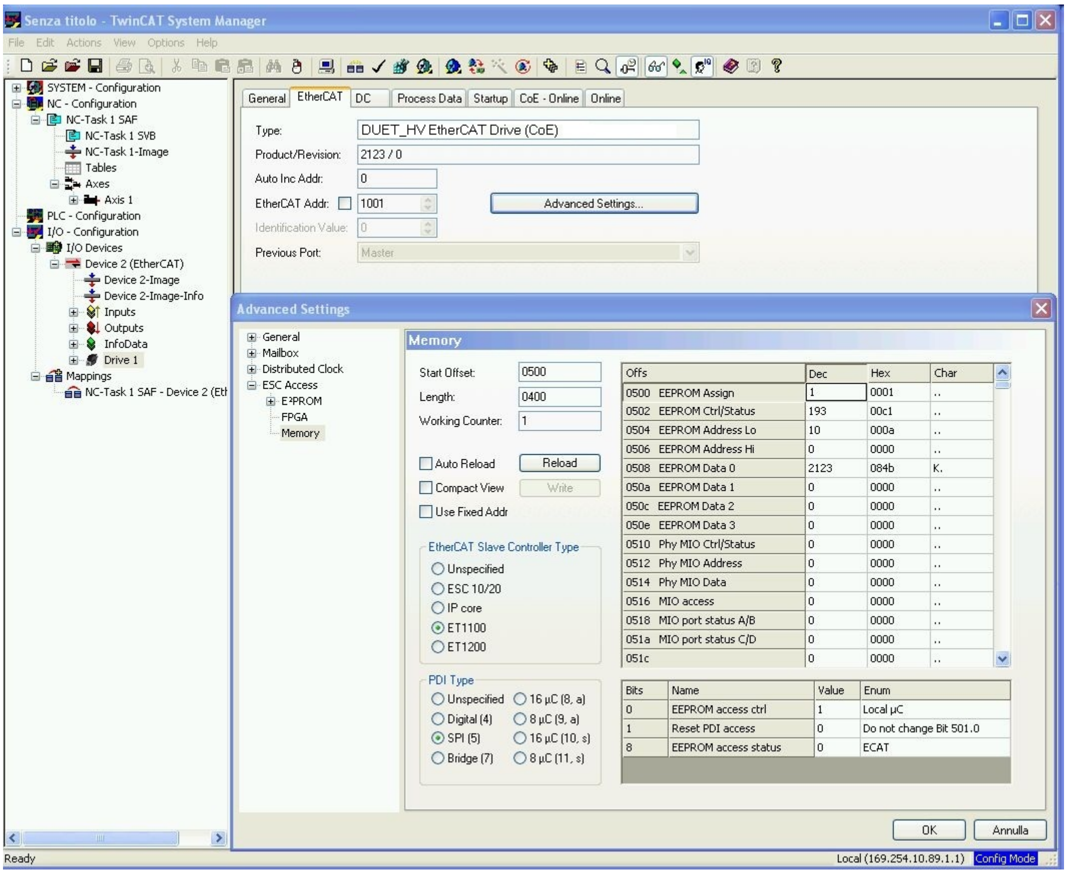
The procedure is commanded/executed by the EtherCAT master.

The EtherCAT master must be connected to the drive through the EtherCAT port, it is assumed to have already installed on the PC the EtherCAT master development environment and to have a project that describes all the peripherals in the EtherCAT network.

The folowing described procedure can be manually executed by an operator or can be inserted at the end of a firmware updating operation made by the master.

The sequence is the following:

1. start the master development environment
2. connect to the drive
3. write 1 on the 0x500 register of the ET1100 (enables the SII_EEPROM access to the DSP)




4. write the command 5400 on the SysMngCommand cell (0x5FF7.01 CoE)
5. execute the reading of the SystemManagerStatus cell (0x5FF7.02 CoE) to read the operation status, wait the end operation code, if it ends with an error read the error code on SysMngError (0x5FF7.03 CoE)
6. write 0 on the 0x500 register of the ET1100 (disabes the SII_EEPROM access to the DSP)
7. execute a power-up cycle on the drive (in case there are more than one drive to update, this operation can be executed only once at the end of the sequence)

## 26.6.4. ESI EEPROM updating from EtherCAT master in automatic mode

The procedure permits to execute the update without an EtherCAT master.

At the end of a firmware download the message: *ESI eeprom will be updated at the next power-up cycle* will appair in the “Show error” window of the MotionDrive tool. By executing the drive turn off/turn on sequence, the ESI eeprom update is automatically executed and at the firmware start-up it's sufficient to check that there is no error signals related to the update. If at the start-up the message: *ESI eeprom updating at the power-up failed* appairs, it means that the automatic procedure has failed due to a drive HW problem.

NOTE

**it's anyway possible to execute always the manual procedure for the update.**

The eventual warning messages are related to the bits in the EtcErrorRetentCommMsg parameter.

# Chapter 27
# Parameters vocabulary

The exchange of data with the drive takes place through a list of parameters, called *Parameters vocabulary*. The parameters define and control every single function of the drive.

## 27.1. Agreements on the parameters description

Every drive parameter is described in this chapter by the fields in the following table:

| Field | Description |
|---|---|
| - | *Desc* means that the field information can be found in the following description. |
| Modbus | Parameter address which is accessible through protocol Modbus. The number is expressed on a decimal basis. |
| CANopen | Parameter address which is accessible through CANopen protocol. The value is expressed on a hexadecimal basis in the format 0xYYYY.ZZ with the following meaning:<br>• YYYY: parameter index;<br>• ZZ: parameter subindex.<br>This field has a meaning only for drives CAN. For drives of ETC version this field refers to the address of the parameter in the vocabulary accessible through the CANopen over EtherCAT protocol. |
| Range | Range of values accepted for the parameter. If it is not specified it means that all values represented by the type of datum associated to the parameter are considered as valid. |
| Default | Parameter default value. |
| Type | Type of datum associated to the parameter: |

| Field | Description |
|---|---|
| | • U8: 8 bits without sign<br>• U16: 16 bits without sign;<br>• U32: 32 bits without sign;<br>• S8: 8 bits with sign;<br>• S16: 16 bits with sign;<br>• S32: 32 bits with sign;<br>• STR: string;<br>• IQN: notation at fixed point at 32 bits with sign and N bits after the point;<br>• FLT: floating point single precision. |
| Units | Unit of measurement of the parameter (see Section 13.1, *Units of measurement of the parameters*). |
| Acc | Type of access to the parameter:<br>• RW (read/write): reading and writing;<br>• WO (write only): only writing;<br>• RO (read only): only reading;<br>• CST (constant): only reading (constant parameter). |
| Pdo | Parameter mapping in a PDO:<br>• *YES*: mappable parameter;<br>• -: not mappable parameter.<br>This field has a meaning only for drives CAN. |
| Mem | Type of parameter saving in the permanent memory:<br>• -: parameters non savable in the permanent memory<br>• ES: parameters savable in the permanent memory that can be restored on command with the default values;<br>• EM: parameters savable in the permanent memory that cannot be restored on command with default values. |

***Table 27.1. Fields describing the parameters***

## 27.2. Reading and writing a parameter

The dimension of every Modbus register is 1 Word (2 byte). Therefore each parameter takes a minimum of 2 bytes of memory. For example:

- if a parameter is 8 bit long (1 byte = 1/2 Word) it takes 1 word anyway, therefore if it is on 4100 Modbus address, the next parameter is on 4101;

- if a parameter is 16 bit long (2 byte = 1 Word) it takes 1 word, therefore if it is on 1201 Modbus address, the next parameter is on 1202;

- if a parameter is 32 bit long (4 byte = 2 Word) it takes 2 words, therefore if it is on 4110 Modbus address, the contained data take even the 4111 register and, in consequence, the next parameter is on 4112;

NOTE

**To read and write a parameter via EtherCAT, using the SDOs expedited (Section 10.1, *Protocol CANopen over EtherCAT (CoE)*).**

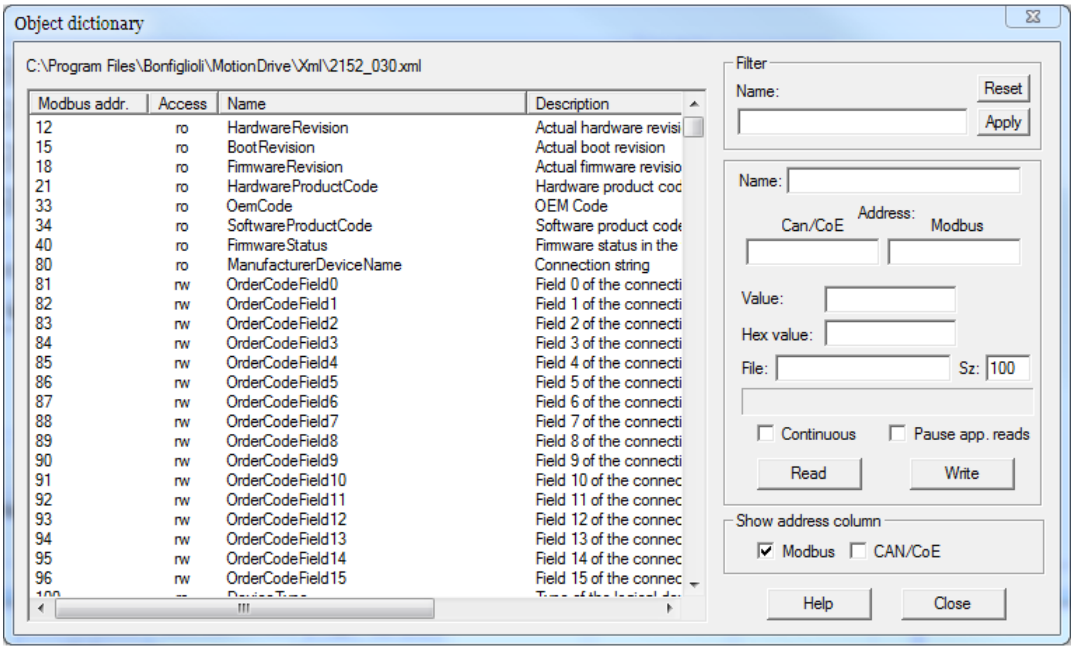
**To read and write a parameter via CANopen, use the SDOs expedited according to the specifications CiA-301.**

**To read and write a parameter via Modbus, send a frame by using the function codes written in Table 11.1.**

Access with MotionDrive:

**Main menu** > **Drive** > **Object dictionary ...**

**Toolbar** > 




To select the parameter to read or write in the Object dictionary window, you can click on the proposed list, write the name and the address or use the search by name functions in the box Filter.

| ADVICE |
|---|
| **To interpret any error messages please see Section 8.4, *Errors in reading / writing of the parameters*.** |

# 27.3. Uploading/downloading

To upload/download via EtherCAT, use the SDOs normal (Section 10.1, *Protocol CANopen over EtherCAT (CoE)*).

To upload/download via CANopen, use the SDOs normal according to the specifications CiA-301.

Via Modbus it is not possible to run correctly an upload/download.

# 27.4. Initial configuration, update and board identity (0-999)

### DriveInformation

Informations related to the drive.

### Number of entries

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x5FFD.00 | 15 | 15 | U8 | - | CST | - | - |

Number of parameters in this group.

### HardwareRevision

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 12 | 0x5FFD.01 | - | - | S16 | - | RO | - | - |

Drive hardware revision.

### BootRevision

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 15 | 0x5FFD.04 | - | - | S16 | - | RO | - | - |

Boot firmware revision.

### FirmwareRevision

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 18 | 0x5FFD.07 | - | - | S16 | - | RO | - | - |

Firmware revision. If -1 is valid, only the boot firmware is present.

### HardwareProductCode

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 21 | 0x5FFD.0A | - | - | U32 | - | RO | - | - |

Product hardware code.

### OemCode

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 33 | 0x5FFD.0E | 6873 | 6873 | U16 | - | CST | - | - |

Code that identifies the constructor.

### SoftwareProductCode

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 34 | 0x5FFD.0F | - | - | U16 | - | RO | - | - |

Product software code.

### *FirmwareStatus*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 40 | 0x5FFE.01 | - | - | U8 | - | RO | - | - |

Status of the firmware.

| FirmwareStatus | Message | Solution |
|---|---|---|
| 0 | CRC has not been checked yet | Wait the end of the download procedure. |
| 1 | Do not launch firmware | |
| 10 | Run | Firmware is executing. |
| 11 | Permanent memory error | Error in the permanent memory, turn off and on again the drive. If the problem persists, please contact Bonfiglioli. |
| 12 | Reserved | - |
| 13 | CRC error | The firmware is corrupted, try again the download procedure. If the problem persists, please contact Bonfiglioli. |
| 14 | Hardware is not compatible with firmware | The hardware is not compatible with firmware. Try a new download procedure with a compatible firmware or substitute the *i*BMD drive with one that has a compatible hardware. In the "Download Firmware" window, choose the desired firmware and press |
| 15 | Boot is not compatible with firmware | The Boot is not compatible with the firmware. Try a new download procedure with a compatible firmware or update the boot. The “Download Firmware” window automatically shows the firmwares and the boots that are compatible. |
| 16 | Firmware is not compatible with hardware | The firmware is not compatible with the hardware. Try a new download procedure with a compatible firmware or substitute the *i*BMD drive with one that has a compatible hardware. In the "Download Firmware" window, choose the desired firmware and press |
| 17 | Firmware is not compatible with boot | Firmware non compatible with the boot. Try a new download procedure with a compatible firmware or update the boot. The “Download Firmware” window automatically shows the firmwares and the boots that are compatible. |
| 18 | Reserved | - |
| 19 | CPLD error | Error during the internal memory programming, try to download again the firmware. If the problem persists, please contact Bonfiglioli. |
| 20 | Firmware exception error | Due to an error that cannot be reset, the firmware start has been blocked. Try to download another firmware and then restore the parameters with the default. |
| 106 | Download: generic time out | Firmware download procedure interrupted. Check the wirings (see Section 7.2, *Electrical connections*), the parameters of the connection (see Chapter 8, *Communicating with the drive*) and, then, try a new download procedure. |
| 113 | Download: memory is busy | The memory of the drive is busy because other procedures are executing on another communication channel, wait that these are ended and try another download procedure. |
| 151 | Download: file corrupted (code 5101) | The firmware file is corrupted, update the files as described in Section 26.5, *Updating the Configuration File* and try again the download procedure. |
| 153 | Download: file requires unsupported features (code 5103) | The firmware download requires some functionalities that are not supported by the boot. Try to download another firmware or update the boot. |

| FirmwareStatus | Message | Solution |
|---|---|---|
| 154 | Download: file requires unsupported features (code 5104) | |
| 155 | Download: file requires unsupported features (code 5105) | |
| 156 | Download: file requires unsupported features (code 5106) | |
| 157 | Download: file requires unsupported features (code 5107) | |
| 158 | Download: file requires unsupported features (code 5108) | |
| 161 | Download: file corrupted (code 5111) | The firmware file is corrupted, update the files as described in Section 26.5, *Updating the Configuration File* and try again the download procedure. |
| 162 | Download: file corrupted (code 5112) | |
| 163 | Download: file corrupted (code 5113) | |
| 164 | Download: file corrupted (code 5114) | |
| 165 | Download: file corrupted (code 5115) | |
| 166 | Download: file requires unsupported features (code 5116) | The firmware download requires some functionalities that are not supported by the boot. Try to download another firmware or update the boot. |
| 167 | Download: file requires unsupported features (code 5117) | |
| 168 | Download: file requires unsupported features (code 5118) | |
| 169 | Download: file requires unsupported features (code 5119) | |
| 170 | Download: file requires unsupported features (code 5120) | |
| 171 | Download: file requires unsupported features (code 5121) | |
| 172 | Download: memory error (code 5122) | Error during the memory programming, try to download again the firmware. If the problem persists, please contact Bonfiglioli. |
| 173 | Download: file corrupted (code 5123) | The firmware file is corrupted, update the files as described in Section 26.5, *Updating the Configuration File* and try again the download procedure. |
| 174 | Download: file corrupted (code 5124) | |
| 175 | Download: memory error (code 5125) | Error during the memory programming, try to download again the firmware. If the problem persists, please contact Bonfiglioli. |
| 200 | Download: memory error (code 5150) | |
| 201 | Download: memory error (code 5151) | |
| 202 | Download: memory error (code 5152) | |
| 203 | Download: memory error (code 5153) | |
| 204 | Download: memory error (code 5154) | |
| 210 | Download: memory error (code 5160) | |
| 211 | Download: memory error (code 5161) | |
| 212 | Download: memory error (code 5162) | |
| 213 | Download: memory error (code 5163) | |
| 214 | Download: memory error (code 5164) | |
| 220 | Download: memory error (code 5170) | |
| 221 | Download: memory error (code 5171) | |
| 230 | Download: memory error (code 5180) | |

### *ManufacturerDeviceName*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 80 | 0x1008.00 | - | - | STR | - | CST | - | - |

Reading of the ManufacturerDeviceName. For further informations see Section 29.1, *OrderCode*

### *DeviceType*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 100 | 0x1000.00 | 0x00020192 | 0x00020192 | U32 | - | CST | - | - |

Code of the type of device and of its functionalities:

- The value in the two less important bytes (0x0192) shows that the device is a drive in compliance with the specification CANopen CiA-402;
- The value in the two most important bytes (0x0002) shows that the drive can control the motor in closed chain.

### *ErrorRegister*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 102 | 0x1001.00 | Desc | - | U8 | - | RO | *YES* | - |

Concisely indicates the status of the drive alarms that are described in the Table 24.1. If a particular type of alarm is present the corresponding bit value is 1, otherwise its value is 0:

| Bit | Description | Related alarms |
|---|---|---|
| 0 | Generic alarm (the value of this bit is 1 if at least one of the following bits value is 1, in other words if it has been detected at least one alarm. Otherwise it's equal to 0). | |
| 1 | Current fault. | Power or motor short circuit, Power or motor over current |
| 2 | Voltage fault | DC bus over voltage, DC bus under voltage, Logic voltage error |
| 3 | Temperature fault | Thermal management |
| 4 | Communication Warning or Fault | - If it is a BD/ETC: EtherCAT communication error<br>- If it is a BD/CAN: CAN communication error, CANopenEmcyService and CANopenEmcyProcess. |
| 5 | Fault of the Device Profile (related to the CiA-402 regulations) | Parameters serious error, Position following error, I2T limit reached, Digital IO configuration error |
| 6 | Reserved | |
| 7 | Manufacturer fault | Real time mode error, Last command requested failed, /STO Management Error, User Fault, Feedback sensor error |

*Table 27.2. Bit coding of ErrorRegister*

The value of this parameter is sent with the emergency messages (see Section 9.2, *Emergency object (EMCY)* ).

### ManufacturerHwVersion

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 110 | 0x1009.00 | - | - | STR | - | CST | - | - |

String in ASCII characters showing the hardware version of the drive.

### ManufacturerSwVersion

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 114 | 0x100A.00 | - | - | STR | - | CST | - | - |

String in ASCII characters showing the software version of the drive.

## Identity

Drive Identity.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x1018.00 | 5 | 5 | U8 | - | CST | - | - |

Number of parameter in this group.

### *VendorID*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 120 | 0x1018.01 | | | U32 | - | RO | - | - |

Number code given to Bonfiglioli Mechatronic Research S.p.a as manufacturer of CANopen or EtherCAT devices.

### *ProductCode*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 122 | 0x1018.02 | - | - | U32 | - | RO | - | - |

Product code.

### *RevisionNumber*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 124 | 0x1018.03 | - | - | U32 | - | RO | - | - |

Revision of the product.

### *SerialNumber*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 126 | 0x1018.04 | - | - | U32 | - | RO | - | - |

Drive serial number.

## CpuInfo

Information about the CPU.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x5FFA.00 | 2 | 2 | U8 | - | CST | - | - |

Number of parameters in this group.

### *SwResetCode*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 560 | 0x5FFA.03 | - | 0 | U16 | - | RO | - | - |

Software reset: reset code. Please contact Bonfiglioli if it is different from 0.

### *SwResetInfo*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 561 | 0x5FFA.04 | - | - | U32 | - | RO | - | - |

Software reset: RPC register value. It identifies a firmware internal problem.

### *CPUSiliconRevision*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 580 | 0x5FFA.01 | - | - | U16 | - | RO | - | - |

CPU revision.

### *ResetCause*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 581 | 0x5FFA.02 | - | - | U32 | - | RW | - | - |

Code of the cause that has caused the firmware reset.

| ResetCause | Description |
|---|---|
| 1 | Reserved |
| 2 | Power-up reset (switching on the drive, command 5000 of SysMngCommand, firmware download Chapter 26, *Software updating*) |
| 3 | Reserved |
| 4 | Reset from watchdog |
| 5 | Reset from NMT reset CANopen |
| 6 | Reset from any other communication channel (command 5001 of SysMngCommand). |

# 27.5. Communication port CANopen (1000-1099)

NOTE

**This vocabulary section can be found only in the drive version CAN**

## CANopenPortStatus

Parameters for the management of the CANopen communication port status.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x5100.00 | 2 | 2 | U8 | - | CST | - | - |

Number of parameters in this group.

### *CANopenPortEnable*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1000 | 0x5100.01 | - | 1 | U16 | - | RW | - | - |

Enabling/disabling the CANopen communication port (0 = disabled port, otherwise enabled port).

### *CANopenPortSwitchSetup*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1001 | 0x5100.02 | 0 - 1 | 0 | U16 | - | RW | - | EM |

Selector of the settings used to configure the CANopen communication port. This parameter is part of the Enabled parameters after reset group.

| CANopen-PortSwitchSetup | Description |
|---|---|
| 0 | Configuration of the node number and baud rate from dip switch (Configuration from dip switch) |
| 1 | Configuration node number and baud rate with CANopenPortSetup. |

NOTE

**If the node number set with the dip switches is 0, you can use the configuration of the parameters CANopenPortSetup.**

## CANopenPortSetup

Parameters used to configure the CANopen communication port from the permanent memory.

#### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x5102.00 | 2 | 2 | U8 | - | CST | - | - |

Number of parameters in this group.

#### *CANopenPortSetupNodeID*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1002 | 0x5102.01 | 1 - 127 | 1 | U16 | - | RW | - | EM |

Setting of the permanent memory of the node number of the CANopen communication port. This parameter is part of the group Enabled parameters after reset.

#### *CANopenPortSetupBaudRate*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1003 | 0x5102.02 | Desc | 1000000 | U32 | bit/s | RW | - | EM |

Setting of the permanent memory of the baud rate of the CANopen communication port. This parameter belongs to the group Enabled parameters after reset. The accepted values are:

- 20000;
- 50000;
- 100000;
- 125000;
- 250000;
- 500000;
- 1000000.

### CANopenPortActual

Parameters showing the current configuration of the CANopen communication port.

#### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x5108.00 | 2 | 2 | U8 | - | CST | - | - |

Number of parameters in this group.

#### *CANopenPortActualNodeID*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1005 | 0x5108.01 | 1 - 127 | - | U16 | - | RO | - | - |

Current setting of the node number of the CANopen communication port.

### CANopenPortActualBaudRate

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1006 | 0x5108.02 | Desc | - | U32 | bit/s | RO | - | - |

Current setting of the baud rate of the CANopen communication port.

### SYNC_CobID

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1010 | 0x1005.00 | Desc | 0x80 | U32 | - | RW | - | - |

Identification code (COB-ID) of the SYNC message. The bit meaning is shown in Table 27.3.

| Bit | Description |
|---|---|
| 0 - 10 | Field at 11 bits for the standard ID (CAN 2.0). |
| 11 - 28 | Field at 18 bits for the extended ID, not supported. They must be set at 0. |
| 29 | Format identifier. It must be equal to 0 (standard identifier). |
| 30 | It must always be 0 because the *i*BMD series drives can be only *consumer* of the SYNC service (see Section 9.3, *Synchronization object (SYNC)*). |
| 31 | Reserved |

***Table 27.3. Bit coding of SYNC_CobID***

### CommunicCyclePeriod

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1012 | 0x1006.00 | 0 - 32000 | 0 | U32 | µs | RW | - | - |

Time period $T_{SYNC}$ during which the SYNC message is transmitted from the *producer* (see Section 9.3, *Synchronization object (SYNC)*). By setting a value equal to 0 the SYNC receiving is disabled by the drive.

NOTE

**In case of interpolated mode use, pay attention to Section 22.10, *Interpolated Position Mode*.**

### GuardTime

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1014 | 0x100C.00 | - | 0 | U16 | ms | RW | - | - |

Period during which the master sends the requests to read the state of the *NMT state machine* to the drive. The drive will answer to every request with a message of *node guarding* showing the internal status of the communication. See Error control services.

### LifeTimeFactor

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1015 | 0x100D.00 | - | 0 | U8 | - | RW | - | - |

By multiplying this parameter for the GuardTime you get the time-out value tolerated by the drive between two consecutive requests of the state of the *NMT state machine*. If this time-out elapses, the drive switches to Fault (see Error control services).

### *EMCY_CobID*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1016 | 0x1014.00 | Desc | 0x80+NodeID | U32 | - | RW | - | - |

Identification code (COB-ID) of the EMCY message. With *NodeID* it is meant the current node number of the drive, which can be read by the CANopenPortActualNodeID parameter. The bit meaning is shown in Table 27.4.

| Bit | Description |
|---|---|
| 0 - 10 | Field at 11 bits for the standard ID (CAN 2.0). |
| 11 - 28 | Field at 18 bits for the extended ID, not supported. They must be set at 0. |
| 29 | Format identifier. It must be equal to 0 (standard identifier). |
| 30 | It must always be 0. |
| 31 | COB-ID Validation: if equal to 0 it is valid and the EMCY service is enabled. |

***Table 27.4. Bit coding of EMCY_CobID***

NOTE

**It is not possible to modify the bits 0 - 29 when the bit 31 is equal to 0 (EMCY service enabled).**

## CANopenCounters

Error frame counters of the CANopen communication port.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x5FFE.00 | 5 | 5 | U8 | - | CST | - | - |

Number of parameters in this group.

### *CANopenStatusRegister*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1060 | 0x5FFE.02 | Desc | - | U32 | - | RO | - | - |

Status register of the CANopen communication port. In the Table 27.5 you can find the meaning of the single bits: for further information see also the specification CAN 2.0.

| Bit | Description |
|---|---|
| 0 | *Warning state*. Bit set when one of the two error counters (TEC or REC) is higher or equal to 96. |
| 1 | *Error-passive state*. Bit set when the communication port CANopen goes to the error-passive state (error counter TEC higher or equal to 128). |
| 2 | *Bus-off state*. Bit set with many errors on the communication port CANopen (error counter TEC higher or equal to 256). During the bus-off no message can be received or transmitted. |

| Bit | Description |
|---|---|
| 3 | *Acknowledge error.* |
| 4 | *Stuff error.* |
| 5 | *Crc error.* |
| 6 | *Stuck at dominant error.* |
| 7 | *Bit error.* |
| 8 | *Form error.* |
| 9 - 15 | Reserved |
| 16 - 23 | *Transmit error counter (TEC).* Error counter in transmission. |
| 24 - 31 | *Receive error counter (REC).* Receiving error counter. |

**Table 27.5. Bit coding of CANopenStatusRegister**

### *CANopenCountTxErrFrame*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1062 | 0x5FFE.04 | - | - | U32 | - | RW | - | - |

Counter of error frame in transmission found by the CANopen communication port.

### *CANopenCountRxErrFrame*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1064 | 0x5FFE.05 | - | - | U32 | - | RW | - | - |

Counter of error frame in transmission found by the CANopen communication port.

## CANopenEmcyRegister

Register of the emergencies sent by the drive in the communication port CANopen.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1080 | 0x5FF9.00 | 3 | 3 | U8 | - | RW | - | - |

Number of parameters in this group. By writing this parameter all parameters associated to the subindexes 1-3 are reset.

### *CANopenEmcyService*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1082 | 0x5FF9.01 | - | - | U32 | - | RO | - | - |

Specific details of the CAN communication error related to the emergencies created by the objects NMT, SYNC, EMCY and SDO.

### *CANopenEmcyProcess*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1084 | 0x5FF9.02 | - | - | U32 | - | RO | - | - |

Specific details of the CAN communication error related to the emergencies created from the objects PDO.

### *CANopenEmcyDevice*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1086 | 0x5FF9.03 | - | - | U32 | - | RO | - | - |

Emergencies created because of the MainError (excluding CAN communication error). The bit coding is shown in Table 24.1.

# 27.6. Communication port EtherCAT (1000-1099)

NOTE

**This vocabulary section can be found only in the drive versions ETC**

## EtherCATPortSetup

Parameters used to configure the EtherCAT communication port from the permanent memory.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x5102.00 | 1 | 1 | U8 | - | CST | - | - |

Number of parameters in this group.

### *EtcConfiguredStationAlias*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1002 | 0x5102.01 | - | - | U16 | - | RO | - | - |

Node number with which the slave can be identified in an EtherCAT network. The parameter reports the value that's contained in the "Configured Station Alias" register of the ET1100 chip (register with address 0x0012-0x0013). For further details see Section 8.2, *Communicate with master EtherCAT*.

## EtherCAT_PortActual

Current configuration of the EtherCAT communication port.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x5108.00 | 1 | 1 | U8 | - | CST | - | - |

Number of parameters in this group.

### *EtcConfiguredStation*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1005 | 0x5108.01 | - | - | U16 | - | RO | - | - |

Node number with which the master recognize the drive through the Fixed addressing (Node Address). The parameter reports the value that's contained in the "Configured Station Address" register of the ET1100 chip (register with address 0x0010-0x0011). For further details see Section 8.2, *Communicate with master EtherCAT*.

### ***CommunicCyclePeriod***

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1012 | 0x1006.00 | 0 - 32000 | 0 | U32 | µs | RW | - | - |

Synchronization time of the PDOs with the synchronization method Soft sync.

NOTE

**In case of interpolated mode use, pay attention to Section 22.10, *Interpolated Position Mode*.**

## EtherCAT_Diagnostics

Specific details of EtherCAT communication error.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x5FF6.00 | 15 | 15 | U8 | - | CST | - | - |

Number of parameters in this group.

### *EtcRegDllStatus*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1020 | 0x5FF6.08 | Desc | - | U16 | - | RO | - | - |

0x110:0x111 register of the ESC

### *EtcRegAlStatus*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1021 | 0x5FF6.09 | Desc | - | U16 | - | RO | - | - |

0x130:0x131 register of the ESC

### *EtcResetPdoRxLostMaxConsecReset*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1084 | 0x5FF6.10 | - | 0 | U16 | - | RW | - | - |

Counter of the total number of consecutively lost PDOs RX. It's automatically reset on the transition from SAFE-OPERATIONAL state to the OPERATIONAL state and it works only in the OPERATIONAL state.

### *EtcRegAlStatusCode*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1022 | 0x5FF6.0A | Desc | - | U16 | - | RO | - | - |

0x134:0x135 register of the ESC

### *EtcRegEEpromConfiguration*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1023 | 0x5FF6.0B | Desc | - | U16 | - | RO | - | - |

0x500:0x501 register of the ESC

### *EtcRegSyncOutUnit*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1024 | 0x5FF6.0C | Desc | - | U16 | - | RO | - | - |

0x980:0x981 register of the ESC

### *EtcRegSyncPulseLenght*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1025 | 0x5FF6.0D | Desc | - | U16 | - | RO | - | - |

0x982:983 register of the ESC

### *EtcRegSyncActivationStatus*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1026 | 0x5FF6.0E | Desc | - | U16 | - | RO | - | - |

0x984 register of the ESC

### *EtcRegSync0CycleTime*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1027 | 0x5FF6.0F | Desc | - | U32 | - | RO | - | - |

0x9A0:9A3 register of the ESC

### *EtcErrorRetentCommMsg*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1070 | 0x5FF6.01 | Desc | - | U32 | - | RO | - | - |

Detail of the errors of the EtherCAT communication error.

IMPORTANT

**The Faults can be generated only if the drive is in the Operation enable (CiA-402) status. In any case it is possible to analyze the causes of an unexpected EtherCAT status change, by reading the EtcRegAlStatusCode parameter.**

### *EtcPdoRxMissingTolerance*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1072 | 0x5FF6.02 | 0 - 128 | 1 | U16 | - | RW | - | - |

Tolerance on the number of PDO RX that can be consecutively lost before the drive generates an error.

ATTENTION

**We suggest to set a greater than 4 tolerance, because every not received and not interpreted PDO RX implies that the movement is not controlled by the Master (see Section 10.3.2, *Missing or corrupted PDO RX management*).**

### *EtcPdoRxLostConsecutive*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1073 | 0x5FF6.03 | - | 0 | U16 | - | RW | - | - |

Counter of the maximum number of consecutively lost PDOs RX (only updated on the OPERATIONAL state).

### *EtcPdoRxLostTotal*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1074 | 0x5FF6.04 | - | 0 | U32 | - | RW | - | - |

Counter of the total number of consecutively lost PDOs RX (only updated on the OPERATIONAL state).

### *EtcPdoRxLostTotalReset*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1076 | 0x5FF6.05 | - | 0 | S32 | - | RW | - | - |

Counter of the total number of lost PDOs RX. It's automatically reset on the transition from SAFE-OPERATIONAL state to the OPERATIONAL state and it works only in the OPERATIONAL state.

### *EtcDcPllResetOnOpe*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1078 | 0x5FF6.06 | Desc | - | S32 | - | RW | - | - |

Difference between the detected Sync Signal number and the correctly received PDO RX messages number (the value is updated only if the drive is on OPERATIONAL state and if the synchronization mode is HardSync, see Section 10.4, *Synchronization*).

### *EtcPdoRxTotal*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1080 | 0x5FF6.07 | Desc | 0 | U32 | - | RW | - | - |

Number of total frames that have been correctly received. The counter dosen't overflow, when the 32bit maximum value is reached the counting stops until its value is modified by a writing operation. It is possible to write any value, it will be reset on the SAFEOPERATIONAL -> OPERATIONAL transition.

# 27.7. Auxiliary communication port (1100-1199)

## AuxiliaryPortSetup

Parameters used to configure the auxiliary communication port.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x5120.00 | 4 | 4 | U8 | - | CST | - | - |

Number of parameters in this group.

### *AuxiliaryPortSetupWordOrder*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1100 | 0x5120.01 | 0 - 1 | 0 | U16 | - | RW | - | EM |

Order of the words used by the drive, through the auxiliary port, to receive or send the parameters of 32 bits (the byte order of the words is big-endian, as defined by the specification of the Modbus protocol, implemented in the auxiliary port).

| Auxiliary Port Setup Word Order | Description | Example |
|---|---|---|
| 0 | Word sent in little-endian format. | The value 0x12345678 is sent in the order 0x5678 0x1234. |
| 1 | Word sent in big-endian format. | The value 0x12345678 is sent in the order 0x1234 0x5678. |

### *AuxiliaryPortSetupTimeout*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1101 | 0x5120.02 | 20 - 65000 | 50 | U16 | ms | RW | - | EM |

Timeout of the auxiliary port. If the time between two consecutive characters overcomes this value, the interface cancels the ongoing receiving of the whole frame and it prepares to receive a new frame.

### AuxiliaryPortSetupBaudRateImmediate

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1110 | 0x5120.03 | 19200 or 57600 | 57600 | U32 | bit/s | RW | - | - |

Parameters used for the immediate exchange of the baud rate of the auxiliary port. Once received the request to change the baud rate, the drive sends the answer with the precedent baud rate and only after it configures the communication interface with the new baud rate.

### AuxiliaryPortSetupBaudRate

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1112 | 0x5120.04 | 19200 o 57600 | 57600 | U32 | bit/s | RW | - | EM |

Baud rate of the auxiliary port. This parameter is included in the Enabled parameters after reset group.

## AuxiliaryPortError

Parameters to read the last error condition in writing or reading carried out with the auxiliary communication port.

### Number of entries

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x5124.00 | 2 | 2 | U8 | - | CST | - | - |

Number of parameters in this group.

### AuxiliaryPortErrorParam

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1120 | 0x5124.01 | - | 0 | U16 | - | RW | - | - |

Modbus address of the parameter that generated the last error condition during the writing/reading phase with the auxiliary communication port. An access in writing causes the resetting of this parameter and of the parameter AuxiliaryPortErrorCode.

### AuxiliaryPortErrorCode

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1121 | 0x5124.02 | Desc | 0 | U16 | - | RW | - | - |

Error code of the last error condition found during the writing/reading phase with the auxiliary communication port. An access in writing causes the resetting of this parameter and of the parameter AuxiliaryPortErrorParam. The meaning of the codes can be found in Table 8.9.

## 27.8. Motor, drive and I2T (1200-1299)

### MotorParameters

Motor parameters.

#### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x6410.00 | 15 | 15 | U8 | - | CST | - | - |

Number of parameters in this group. For any further information, see Section 15.1, *Motor parametrization*.

#### *MotorStallCurrent*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1201 | 0x6410.01 | - | - | U16 | 100 = 1A | RW | - | EM |

It is the motor stall current, that corresponds to the current of the maximum motor torque, with a close to 0 velocity, without its thermal limits are exceeded.

#### *MotorPeakCurrent*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1202 | 0x6410.02 | - | - | U16 | 100 = 1A | RW | - | EM |

Motor peak current.

#### *CoggingTorque(CoggingForce)*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1203 | 0x6410.03 | - | - | U16 | Desc | RW | - | EM |

Motor cogging torque. The measurement unit depends on the motor type: rotative motor [1000 = 1Nm], linear motor [1000 = 1N].

#### *MotorInductance*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1204 | 0x6410.04 | - | - | U16 | 100 = 1mH | RW | - | EM |

Phase-phase motor inductance.

#### *MotorResistance*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1205 | 0x6410.05 | - | - | U16 | mΩ | RW | - | EM |

Phase-phase motor resistance.

### MotorInertia(MotorMass)

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1206 | 0x6410.06 | - | - | U16 | Desc | RW | - | EM |

Motor moment of inertia. The unit of measurement depends on the motor type: rotary motor [1 = 10g $cm^2$ ], linear motor [1 = 10g].

### TorqueConstant(ForceConstant)

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1212 | 0x6410.08 | - | - | U16 | Desc | RW | - | EM |

Motor torque constant. The unit of measurement depends on the motor type: rotary motor [1000 = 1Nm/A], linear motor [10 = 1N/A].

### MotorRatedSpeed

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1213 | 0x6410.09 | - | - | U32 | Desc | RW | - | EM |

Motor nominal velocity. The unit of measurement depends on the motor type: rotary motor [rpm], linear motor [mm/s].

### MotorPoles

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1215 | 0x6410.0A | - | - | U16 | - | RW | - | EM |

Number of motor poles.

### FaultTemperatureThrs

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1216 | 0x6410.0B | - | - | U16 | - | RW | - | EM |

Motor temperature fault threshold.

For further details please see Table 24.3.

### MotorMotionType

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1217 | 0x6410.0C | 0 ÷ 1 | - | U16 | - | RW | - | EM |

Motor type: 0 = rotary, 1 = linear.

### PolePitch

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1218 | 0x6410.0D | - | - | U16 | mm | RW | - | EM |

Linear motor pole pitch.

### *MotorFaultTemperatureThrsOhm*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1219 | 0x6410.0E | - | - | U32 | Ω | RW | - | EM |

Resistance value that causes the motor temperature Fault, if the motor temperature sensor is PTC.

### *MotorTemperatureSensorType*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1221 | 0x6410.0F | 0-2 | - | U16 | - | RW | - | EM |

Temperature sensor type built on the motor. For the temperature value reading, if this parameter value is 1 please refer to MotorTemperature, if it is 2 refer to MotorTemperaturePTC.

| Code | Sensor type |
|---|---|
| 0 | No sensor – the temperature measurement is disabled |
| 1 | KTY84 |
| 2 | PTC SWITCH |

NOTE

**The temperature sensor is present only since the HardwareRevision 5. Drives with previous hardware versions can be configured only if MotorTemperatureSensorType = 0.**

## DriveParameters

Drive parameters.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x6510.00 | - | 4 | U8 | - | CST | - | - |

Number of parameters in this group.

### *UserDriveName*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1250 | 0x6510.04 | - | - | - | - | RW | - | EM |

Drive name, given by the user. For an easier drive identification, it's possible to insert an alphanumeric string of up to 16 characters. The UserDriveName parameter must be considered as any other parameter: it's saved in the permanent memory and it's managed by the parameters file as the others parameters. UserDriveName is showed on the connection status and in the main page heading.

### MaxRatedCurrent

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1253 | 0x6510.01 | - | - | U16 | 100 = 1A | RO | - | - |

Drive nominal current, power section.

### MaxPeakCurrent

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1254 | 0x6510.02 | - | - | U16 | 100 = 1A | RO | - | - |

Drive peak current, power section.

### MaxSupplyVoltage

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1255 | 0x6510.03 | - | - | U16 | V | RO | - | - |

Maximum supply voltage, power section.

## I2TParameters

Parameters of the I2T limitation.

### Number of entries

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x3405.00 | - | 6 | U8 | - | CST | - | - |

Number of parameters in this group. For further details see Section 14.6, *I2T*.

### UserPeakCurrent

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1258 | 0x3405.06 | - | Desc | U16 | 100 = 1A | RW | *YES* | ES |

Peak current that can be set by the user to limit the current supply to the motor. It contributes to determine UserMaxI2T. UserPeakCurrent must be different from zero and lower or equal to MotorPeakCurrent and MaxPeakCurrent. Its default value is the lowest value between MotorPeakCurrent and MaxPeakCurrent.

### I2TTime

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1259 | 0x3405.01 | - | Desc | U16 | ms | RW | - | ES |

Maximum time the drive can keep the motor current at the value of PeakCurrent. It contributes to determine UserMaxI2T. Its default value is so that UserMaxI2T is lower than DriveMaxI2T, with a maximum value of 5s.

### *UserMaxI2T*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1260 | 0x3405.02 | - | - | U32 | $A^2s$ | RO | - | - |

Maximum value of I2T calculated depending on PeakCurrent and I2TTime. Its value must be lower than DriveMaxI2T.

### *DriveMaxI2T*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1262 | 0x3405.03 | - | - | U32 | $A^2$ ms | RO | - | - |

Maximum value of drive I2T.

### *I2TWarningThreshold*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1264 | 0x3405.04 | 0-100 | 80 | U16 | % | RW | - | ES |

Warning threshold enabling the I2T Warning threshold reached error.

## 27.9. Tuning (1400-1499)

### ***ResetWatchdogTimeout***

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1400 | 0x3500.00 | - | - | U16 | - | WO | - | - |

Write in this parameter the SysMngCommand value to execute the Reset of the Watchdog of the System manager.

### **TuningConfigurations**

TuningConfigurations.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x3502.00 | - | 4 | U8 | - | CST | - | - |

Number of parameters in this group.

### *DynamicResponse*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1401 | 0x3502.01 | - | 150 | U16 | - | RW | *YES* | ES |

Tuning configuration used to select the dynamic response of the motor.

| DynamicResponse | Description |
|---|---|
| 120 | Lowest |
| 130 | Very low |
| 140 | Low |
| 150 | Medium |
| 160 | High |
| 170 | Very high |
| 180 | Highest |

### *Stiffness*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1402 | 0x3502.02 | - | 150 | U16 | - | RW | *YES* | ES |

Tuning configuration used to select the motor stiffness at low velocities. Low speeds are 30% lower than the HighSpeed parameter.

| Stiffness | Description |
|---|---|
| 130 | Very low |
| 140 | Low |
| 150 | Medium |
| 160 | High |
| 170 | Very high |

### *VelocityLoopFilter1*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1403 | 0x3502.03 | - | 2 | U16 | - | RW | *YES* | ES |

Tuning configuration that defines the filtering action of the velocity loop. For further details see *"Tuning configuration modifications"* in the Section 20.3, *"Fast tuning guide"*.

| VelocityLoopFilter1 | Description |
|---|---|
| 1 | User |
| 2 | Noise filter |
| 3 | Disable |
| 51 | Soft filter |

### *VelocityLoopFilter2*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1404 | 0x3502.04 | - | 3 | U16 | - | RW | *YES* | ES |

Tuning configuration selecting the second filter of the speed loop. For further details please see *"Tuning configuration modifications"* in the Section 20.3, *"Fast tuning guide"*.

| VelocityLoopFilter2 | Description |
|---|---|
| 1 | User |
| 2 | Resonance filter |

| VelocityLoopFilter2 | Description |
|---|---|
| 3 | Disable |
| 50 | Debounce filter |

## EstimatedLoopsBandwidth

Loops estimated bandwidths.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x3501.00 | - | 3 | U8 | - | CST | - | - |

Number of parameters in this group.

### *CurrentLoopEstimatedBandwidth*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1407 | 0x3501.01 | - | - | U16 | Hz | RO | - | - |

CurrentLoop estimated bandwidth.

### *VelocityLoopEstimatedBandwidth*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1408 | 0x3501.02 | - | - | U16 | Hz | RO | - | - |

VelocityLoop estimated bandwidth.

### *PositionLoopEstimatedBandwidth*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1409 | 0x3501.03 | - | - | U16 | Hz | RO | - | - |

PositionLoop Estimated Bandwidth.

## *TuningEndOption*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1450 | 0x3515.08 | - | 22 | S16 | - | RW | - | - |

Operation that is executed when the End button of the MotionDrive (command 100 of the System Manager) is pushed.

| TuningEndOption | Description |
|---|---|
| 0 | Immediately disable, the motor is stopped with maximum deceleration resetting RequestedSpeed to zero and then the drive has the *Switch On Disabled* state |
| 10 | Zero speed and disable, the motor is stopped with maximum deceleration resetting RequestedSpeed at zero and then the drive has the *Switch On Disabled* state |
| 11 | Zero speed and switched on, the motor is stopped with maximum deceleration, by resetting RequestedSpeed at zero, and then the drive has the *Switched On* state |

| TuningEndOption | Description |
|---|---|
| 12 | Zero speed and previous state, the motor is stopped with maximum deceleration resetting RequestedSpeed at zero and then the drive has the same status before the requested tuning command |
| 20 | Deceleration ramp and disable, the motor is stopped with a deceleration equal to TuningEndDeceleration and then the drive has the *Switch On Disabled* state |
| 21 | Deceleration ramp and switched on, the motor is stopped with a deceleration equal to TuningEndDeceleration and then the drive has the *Switched On* state |
| 22 | Deceleration ramp and previous state, the motor is stopped with a deceleration equal to TuningEndDeceleration and then the drive has the same status before the requested tuning command |

### *TuningEndDeceleration*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1451 | 0x3515.09 | Desc | Desc | U32 | inc/$s^2$ | RW | - | - |

Motor deceleration value when the End button in MotionDrive is pushed (command 100 of the System Manager). The default value is 1000.0 rad/$s^2$ while the range of the accepted values goes from 0.32 to 205887.3 rad/$s^2$.

## InertiaEstimator

Parameters for the inertia estimation

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x3503.00 | 6 | 6 | U8 | - | CST | - | - |

Number of parameters in this group.

### *InertiaEstimatorDirection*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1460 | 3503.01 | - | 0 | U16 | - | RW | - | - |

Rotation direction of the drive shaft using the inertia estimator (0 = positive, 192 = negative).

### *EstimatorTorque*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1461 | 3503.02 | 0 ÷ 32767 | 1000 | U16 | 10 = 1%IS | RW | - | - |

Torque requested to the motor using the inertia estimator.

### *InertiaEstimatorVelocity*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1462 | 3503.03 | - | Desc | U32 | inc/s | RW | - | - |

Velocity requested to the motor using the inertia estimator. The default value is 100.0 rad/s.

### *EstimatedDampingFactor*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1464 | 3503.04 | - | Desc | U16 | 0,0001 A s | RW | - | EM |

Estimated value between the viscous friction and TorqueConstant(ForceConstant).

### *EstimatedInertia*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1466 | 0x3503.05 | - | 10 | U16 | 10 = 1Jm | RW | - | EM |

Total inertia moment, calculated compared to the motor shaft. EstimatedInertia must take into consideration the moments of motor inertia, brake mechanical transmission and load.

### *InertiaReductionFactor*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1467 | 0x3503.06 | 0 ÷ 1000 | 1000 | U16 | Desc | RW | - | EM |

Reduction factor of EstimatedInertia (0 = maximum reduction, 1000 = no reduction). The result of the reduction is used to calculate the regulation loops gains.

## RLEstimator

Parameters for the phase resistance and the motor inductance

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x3504.00 | 12 | 12 | U8 | - | CST | - | - |

Number of parameters in this group.

IMPORTANT

**The EstimatedPhaseResistance, EstimatedLDNominalP, EstimatedLDNominalN, EstimatedLDPeakP, EstimatedLDPeakN, EstimatedLQNominalP, EstimatedLQNominalN, EstimatedLQPeakP, EstimatedLQPeakN parameters cannot be downloaded through the parameters file because they are specific for every axis. On the contrary, the parameters file download reset them to the default value.**

### *EstimatedPhaseResistance*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1470 | 0x3504.01 | - | - | U16 | mΩ | RO | - | EM |

Phase resistance estimated with RLEstimator(see Section 20.7, *RL estimator*).

### *MotorPhaseResistance*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1471 | 0x3504.02 | - | - | U16 | mΩ | RO | - | EM |

Theoretical phase resistance obtained with the motor nameplate data (MotorResistance).

### MotorSynchronousInductance

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1473 | 0x3504.04 | - | - | U16 | 100 = 1mH | RO | - | EM |

Theoretical synchronous inductance obtained with the motor nameplate data (MotorInductance).

### EstimatedLDNominalP

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1474 | 0x3504.05 | - | - | U16 | 100 = 1mH | RO | - | EM |

Synchronous inductance, estimated with RLEstimator and positive ActualFieldCurrent equal to NominalCurrent.

### EstimatedLDNominalN

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1475 | 0x3504.06 | - | - | U16 | 100 = 1mH | RO | - | EM |

Synchronous inductance, estimated with RLEstimator and negative ActualFieldCurrent equal to NominalCurrent.

### EstimatedLDPeakP

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1476 | 0x3504.07 | - | - | U16 | 100 = 1mH | RO | - | EM |

Synchronous inductance, estimated with RLEstimator and positive ActualFieldCurrent equal to PeakCurrent.

### EstimatedLDPeakN

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1477 | 0x3504.08 | - | - | U16 | 100 = 1mH | RO | - | EM |

Synchronous inductance, estimated with RLEstimator and negative ActualFieldCurrent equal to PeakCurrent.

### EstimatedLQNominalP

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1478 | 0x3504.09 | - | - | U16 | 100 = 1mH | RO | - | EM |

Synchronous inductance, estimated with RLEstimator and positive ActualTorqueCurrent equal to NominalCurrent.

### EstimatedLQNominalN

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1479 | 0x3504.0A | - | - | U16 | 100 = 1mH | RO | - | EM |

Synchronous inductance, estimated with RLEstimator and negative ActualTorqueCurrent equal to NominalCurrent.

### EstimatedLQPeakP

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1480 | 0x3504.0B | - | - | U16 | 100 = 1mH | RO | - | EM |

Synchronous inductance, estimated with RLEstimator and positive ActualTorqueCurrent equal to PeakCurrent.

### EstimatedLQPeakN

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1481 | 0x3504.0C | - | - | U16 | 100 = 1mH | RO | - | EM |

Synchronous inductance, estimated with RLEstimator and positive ActualTorqueCurrent equal to PeakCurrent.

### *EstimatedResonanceFrequency*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1483 | 0x3505.01 | - | - | U16 | Hz | RW | - | EM |

Estimated value of the mechanical resonance frequency.

### *VelocityStandStill*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1490 | 0x3523.00 | - | - | U16 | - | RW | - | ES |

Regulation of the gains of the speed regulator for low speeds (it modifies both KVp_LS and KVi_LS).

## 27.10. Loop (1500-1599)

### *ResetSpeedIntegrator*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1516 | 0x60F9.22 | - | - | U16 | - | WO | - | - |

By writing any value in this parameter, the integrative memory of the velocity regulator is reset to 0.

## LoopConfiguration

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1529 | 0x3522.00 | 0 ÷ 2 | 2 | U16 | - | RW | - | ES |

Loops configuration.

| LoopConfiguration | Description |
|---|---|
| 0 | Gains not set |
| 1 | Basic configuration (available with every FirmwareRevision). |
| 2 | Smith configuration (available since FirmwareRevision ≥ 14). |

## VelocityLoop

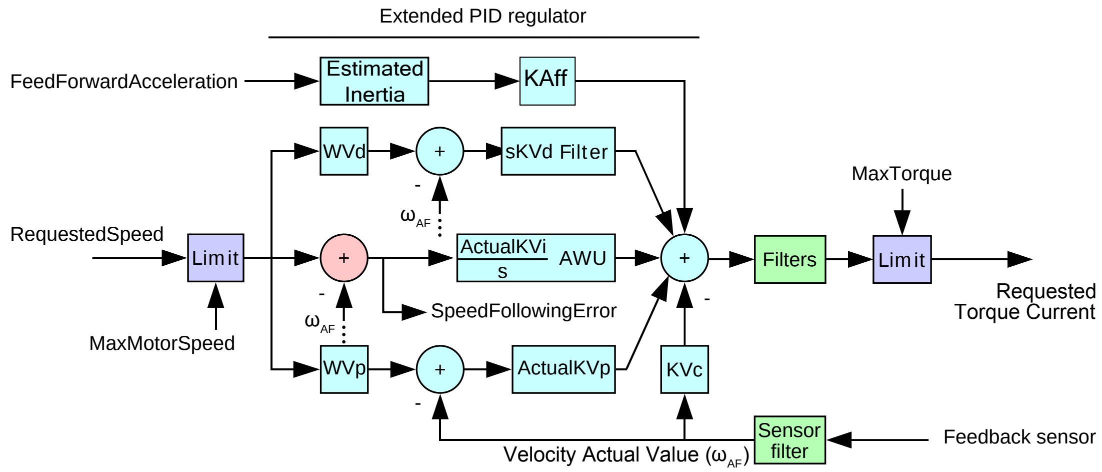



*Figure 27.1. Flow chart of the speed regulator.*

Velocity loop.

### Number of entries

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x60F9.00 | - | 33 | U8 | - | CST | - | - |

Number of parameters in this group. The VelocityLoop is composed by the following blocks:

- input speed limiter
- PID regulator with more degrees of freedom made up by five components: acceleration feed forward, derivative with weight and filter, integral with persistence limit (*AWU*), proportional with weight, damping.
- three filters in the regulator output
- one filter on the sensor of the feedback position
- limiter of the output torque.

### EnableVelocityStandStill

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1500 | 0x60F9.17 | - | - | U16 | - | RW | *YES* | ES |

Enabling of the VelocityStandStill: (0 = disabled, 1 = enabled).

### *LowSpeed*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1501 | 0x60F9.09 | - | - | U32 | inc/s | RW | *YES* | ES |

Speed threshold to use only the Stand still parameters.

### *HighSpeed*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1505 | 0x60F9.08 | - | - | U32 | inc/s | RW | *YES* | ES |

Out of this speed threshold the Stand still parameters do not have any effect.

### *KVp_LS*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1509 | 0x60F9.04 | - | - | U16 | - | RW | *YES* | ES |

Proportional gain of the speed regulator for low speeds.

### *KVi_LS*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1511 | 0x60F9.05 | - | - | U16 | - | RW | *YES* | ES |

Integral gain of the speed regulator for low speeds.

### *ActualKVp*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1513 | 0x60F9.0C | - | - | U16 | - | RO | *YES* | - |

Proportional gain currently used by the speed regulator.

### *ActualKVi*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1515 | 0x60F9.0D | - | - | U16 | - | RO | *YES* | - |

Integral gain currently used by the speed regulator.

### *KVp*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1517 | 0x60F9.01 | - | - | U16 | - | RW | *YES* | ES |

Proportional gain of the speed regulator.

### KVi

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1519 | 0x60F9.03 | - | - | U16 | - | RW | *YES* | ES |

Integral gain of the speed regulator.

### KVdFilterFrequency

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1524 | 0x60F9.10 | - | - | U16 | - | RW | *YES* | ES |

Typical filter frequency on the derivative component of the speed regulator. By increasing this parameter, the filtering action decreases.

### KVd

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1525 | 0x60F9.11 | - | - | U16 | - | RW | *YES* | ES |

Derivative gain of the speed regulator.

### WVd

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1526 | 0x60F9.12 | - | - | U16 | 1000 = 1 | RW | *YES* | ES |

Weighting coefficient of the speed reference in the calculation of the derivative component.

### WVp

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1527 | 0x60F9.13 | - | - | U16 | 1000 = 1 | RW | *YES* | ES |

Weighting coefficient of the speed reference in the calculation of the proportional component.

### KVc

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1528 | 0x60F9.14 | - | - | U16 | - | RW | *YES* | ES |

Damping gain of the speed regulator.

### VFilter1Frequency

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1540 | 0x60F9.0E | - | - | U16 | Hz | RW | *YES* | ES |

Typical frequency of the first filter on the output of the speed regulator.

## *VFilter1Type*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1541 | 0x60F9.0F | - | - | U16 | - | RW | *YES* | ES |

Type of the first filter on the output of the speed regulator.

| VFilterType1 | Description |
|---|---|
| 0 | All-stop filter |
| 1 | Low-pass filter of the first order |
| 2 | Low-pass filter of the second order |
| 3 | Band-eliminating filter |
| 65535 | All-pass filter |

## *VFilter1QFactor*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1542 | 0x60F9.18 | - | - | U16 | 10 = 1 | RW | *YES* | ES |

Quality Q factor of the first filter on the output of the speed regulator.

## *VFilter2Frequency*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1543 | 0x60F9.19 | - | - | U16 | Hz | RW | *YES* | ES |

Typical frequency of the second filter on the output of the speed regulator.

## *VFilter2Type*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1544 | 0x60F9.1A | - | - | U16 | - | RW | *YES* | ES |

Type of the second filter on the output of the speed regulator.

| VFilterType2 | Description |
|---|---|
| 0 | All-stop filter |
| 1 | Low-pass filter of the first order |
| 2 | Low-pass filter of the second order |
| 3 | Band-eliminating filter |
| 65535 | All-pass filter |

## *VFilter2QFactor*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1545 | 0x60F9.1B | - | - | U16 | 10 = 1 | RW | *YES* | ES |

Quality Q factor of the second filter on the output of the speed regulator.

### VFilter3Frequency

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1546 | 0x60F9.1C | - | - | U16 | Hz | RW | *YES* | ES |

Typical frequency of the third filter on the output of the speed regulator.

### VFilter3Type

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1547 | 0x60F9.1D | - | - | U16 | - | RW | *YES* | ES |

Type of the third filter on the output of the speed regulator.

| VFilterType3 | Description |
|---|---|
| 0 | All-stop filter |
| 1 | Low-pass filter of the first order |
| 2 | Low-pass filter of the second order |
| 3 | Band-eliminating filter |
| 65535 | All-pass filter |

### VFilter3QFactor

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1548 | 0x60F9.1E | - | - | U16 | 10 = 1 | RW | *YES* | ES |

Quality Q factor of the third filter on the output of the speed regulator.

### DegaussTypeFilter

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1557 | 0x3520.06 | Desc | 2 | U16 | - | RW | - | ES |

Selector that allows to activate/deactivate the field weakening functionality, by acting on the filter type on the homonymous regulator output and that is not part of any loop in the Loop settings page. The values that can be inserted are listed in the following table:

| DegaussTypeFilter | Description |
|---|---|
| 0 | Disable weakening - All-stop filter |
| 1 | Enable weakening - Low-pass filter of the first order |
| 2 | Enable weakening - Low-pass filter of the second order |
| 3 | Enable weakening - Band-eliminating filter |
| 65535 | Enable weakening - All-pass filter |

### VFilterSensorFrequency

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1570 | 0x60F9.1F | - | - | U16 | Hz | RW | *YES* | ES |

Typical frequency of the filter on the feedback position sensor.

### *VFilterSensorType*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1571 | 0x60F9.20 | - | - | U16 | - | RW | *YES* | ES |

Filter type on the sensor of the feedback position.

| VFilterTypeSensor | Description |
|---|---|
| 0 | All-stop filter |
| 1 | Low-pass filter of the first order |
| 2 | Low-pass filter of the second order |
| 3 | Band-eliminating filter |
| 65535 | All-pass filter |

### *VFilterSensorQFactor*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1572 | 0x60F9.21 | - | - | U16 | 10 = 1 | RW | *YES* | ES |

Quality Q factor of the filter on the feedback position sensor.

### *KAff*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1590 | 0x60F9.16 | - | - | U16 | 1000 = 1 | RW | *YES* | ES |

Acceleration feed forward gain.

## PositionLoop

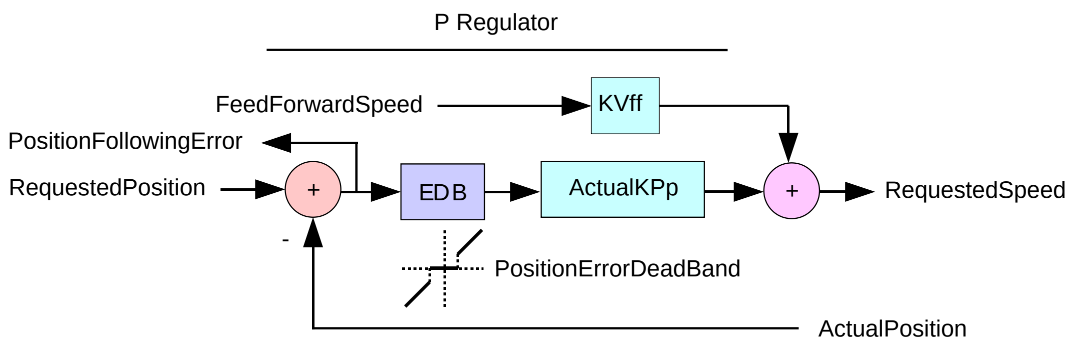



***Figure 27.2. Flow chart of the position regulator.***

Position loop.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x60FB.00 | - | 6 | U8 | - | CST | - | - |

Number of parameters in this group. The position regulator is of P type, made up of two components, proportional and speed feed forward. Just at its input there is the resetting block of the PositionFollowingError when it is included in the PositionErrorDeadBand (EDB) located near the zero.

### *KPp*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1560 | 0x60FB.01 | - | - | U16 | - | RW | *YES* | ES |

Proportional gain of the position regulator.

### *KVff*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1561 | 0x60FB.02 | - | - | U16 | 1000 = 1 | RW | *YES* | ES |

Velocity feed forward gain.

### *PositionStandStill*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1562 | 0x60FB.03 | - | - | U16 | - | RW | *YES* | ES |

Proportional gain of the position regulator for low velocities.

### *EnablePositionStandStill*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1563 | 0x60FB.04 | - | - | U16 | - | RW | *YES* | ES |

Enabling of the PositionStandStill: (0 = disabled, 1 = enabled).

### *ActualKPp*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1564 | 0x60FB.05 | - | - | U16 | - | RO | *YES* | - |

Proportional gain currently used by the position regulator.

### *ClosePositionLoop*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1565 | 0x60FB.06 | - | 1 | U16 | - | RW | *YES* | ES |

It enables the internal closure of the drive of the position loop.

### ***PositionErrorDeadBand***

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1565 | 0x4281.01 | - | 0 | U16 | inc | RW | - | ES |

Half width of the dead band of the PositionFollowingError.

## CurrentLoop

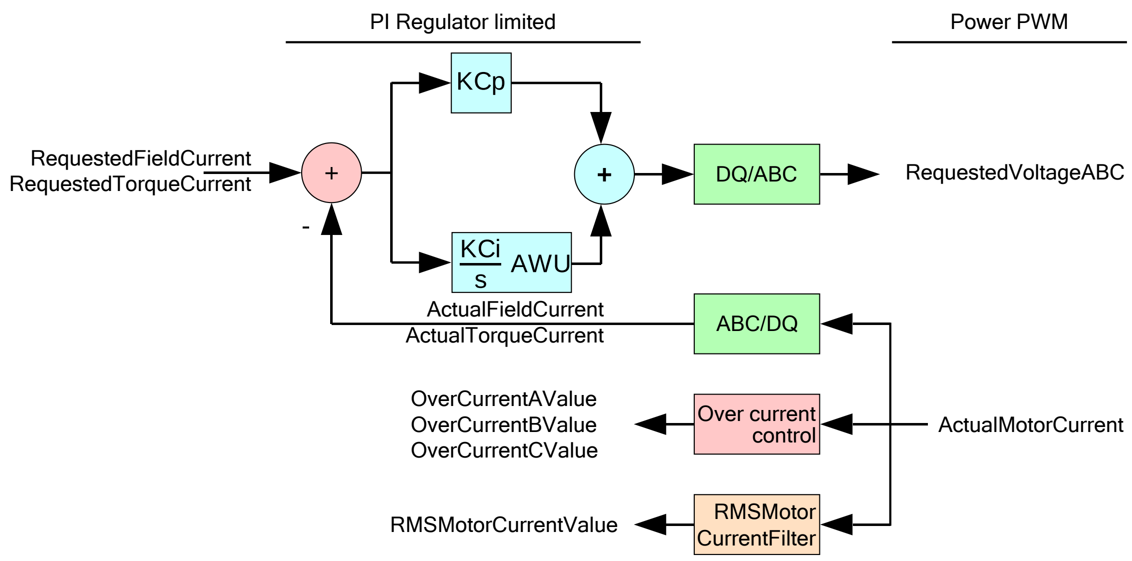



*Figure 27.3. Flow chart of the logical part of the CurrentLoop. Number of parameters in this group.*

Current loop.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x60F6.00 | 10 | 10 | U8 | - | CST | - | - |

Number of parameters in this group. The current regulator is of PI type, made up of the two components, proportional and integral with persistence limit of the integral part (Anti Wind Up). For the regulator you can use two different gain torques: one for the torque component and one for the field component.

### *KCp_Q*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1580 | 0x60F6.01 | - | - | U16 | - | RW | *YES* | ES |

Proportional gain of the torque current regulator.

### *KCi_Q*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1581 | 0x60F6.02 | - | - | U16 | - | RW | *YES* | ES |

Integral gain of the torque current regulator.

## KCp_D

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1582 | 0x60F6.03 | - | - | U16 | - | RW | *YES* | ES |

Proportional gain of the field current regulator.

## KCi_D

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1583 | 0x60F6.04 | - | - | U16 | - | RW | *YES* | ES |

Integral gain of the field current regulator.

## KC_FilterFrequency

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1584 | 0x60F6.06 | - | - | U16 | Hz | RW | *YES* | ES |

Typical frequency of the filter on the output of the current regulators.

## KC_FilterType

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1585 | 0x60F6.07 | - | - | U16 | - | RW | *YES* | ES |

Type of the filter on the output of the current regulator.

| KC_FilterFrequency | Description |
|---|---|
| 0 | All-stop filter |
| 1 | Low-pass filter of the first order |
| 2 | Low-pass filter of the second order |
| 3 | Band-eliminating filter |
| 65535 | All-pass filter |

## KC_FilterQFactor

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1586 | 0x60F6.08 | - | - | U16 | 10 = 1 | RW | *YES* | ES |

Quality factor of the filter on the current regulators output.

## KC_QReduction

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1587 | 0x60F6.09 | 0 ÷ 1000 | - | U16 | Desc | RW | *YES* | ES |

Gains reduction factor to compensate the Lq saturation (0 = no reduction, 1000 = maximum reduction).

### *EnableLoopCompensation*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1588 | 0x60F6.05 | - | - | U16 | - | RW | *YES* | ES |

Each bit of this parameter enables the related functionality. The functionalities are listed in the following table.

| Bit | Name | Description |
|---|---|---|
| 0 | EMF Compensation | enable the counterelectromotive force compensation. |
| 1 | crossDQ | Cross coupling contributions compensation of the current regulators. |
| 2 | Reserved | - |
| 3 | Predictive current measurement | enable the predictive current measurement. |
| 4 | KC_QReduction | enable the gains reduction to compensate the Lq saturation. |

### *AngleObserverBandwidth*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1589 | 0x60F6.0A | - | - | U16 | Hz | RW | *YES* | ES |

Natural frequency of the observer that estimates the rotor position.

### ***LoopType***

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1595 | 0x3080.00 | 0 ÷ 1 | 0 | U16 | - | RW | - | - |

Loops control mode (0 = Servo mode, 1 = Micro step).

# 27.11. Power Pwm (1600-1699)

## PowerPwmParameters

Power pwm parameters.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x3521.00 | - | 7 | U8 | - | CST | - | - |

Number of parameters in this group. For further details see Section 14.5, *Power PWM*.

### *PwmBridgeFrequency*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1600 | 0x3521.01 | 1500-30000 | 5000 | U16 | Hz | RO | - | ES |

Three-phase bridge modulation frequency.

### *PwmModulationMethod*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1601 | 0x3521.02 | 2 | 2 | U16 | - | RO | - | ES |

Modulation type of the three-phase bridge; 2 = asymmetrical.

### *PwmMotionLoopDivider*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1602 | 0x3521.03 | - | 1 | U16 | - | RO | - | ES |

Reduction factor of the loop motion period compared to the current loop period.

### *PwmMotionLoopCode*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1603 | 0x3521.07 | - | 0 | U16 | - | WO | - | - |

Unique code to set the frequency of the three-phase bridge and of the loop period. Writable only when the motor is disabled.

| PwmMotion-LoopCode | PwmBridge-Frequency | PwmModula-tionMethod | PwmMotion-LoopDivider | Motion-LoopPeriod | Current-LoopPeriod |
|---|---|---|---|---|---|
| 0 | 5000 | 2 | 1 | 100 | 100 |

### *MotionLoopPeriod*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1610 | 0x3521.04 | - | 100 | U16 | µs | RO | - | - |

Motion loop period.

### *CurrentLoopPeriod*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1611 | 0x3521.05 | - | 100 | U16 | µs | RO | - | - |

Current loop period.

## 27.12. Drive status (1800-1999)

### TemperatureStatus

Temperature status.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x3300.00 | - | 5 | U8 | - | CST | - | - |

Number of parameters in this group.

### *PowerTemperature*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1800 | 0x3300.01 | - | - | S16 | 10 = 1°C | RO | *YES* | - |

Power section actual temperature.

### *LogicTemperature*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1801 | 0x3300.02 | - | - | S16 | 10 = 1°C | RO | *YES* | - |

Control section actual temperature.

### *MotorTemperature*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1802 | 0x3300.03 | - | - | S16 | 10 = 1°C | RO | *YES* | - |

Actual motor temperature (if the selected sensor is KTY, see MotorTemperatureSensorType).

### *FeedbackSensorTemperature*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1803 | 0x3300.04 | - | - | S16 | 1 = 1°C | RO | *YES* | - |

Feedback sensor temperature. This parameter returns the temperature only if it is installed an Absolute encoder Hiperface position sensor (see Section 15.4, *Absolute feedback position sensor*). With the other sensor types this cell returns 0.

### *MotorTemperaturePTC*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1804 | 0x3300.05 | - | - | S32 | Ω | RO | *YES* | - |

Resistor value for the motor temperature when the selected sensor is PTC (see MotorTemperatureSensorType).

### ***DCBusVoltage(+HV)***

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1840 | 0x3310.01 | - | - | U16 | 10 = 1V | RO | *YES* | - |

DC bus voltage feeding the drive power section.

### ***I2TValue***

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1880 | 0x3405.05 | - | - | U16 | % | RO | *YES* | - |

Actual value of I2T.

## CurrentStatus

Current status.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x3320.00 | - | 9 | U8 | - | CST | - | - |

Number of parameters in this group.

### *ActualMotorCurrent*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1881 | 0x3320.01 | - | - | U16 | 100 = 1A | RO | *YES* | - |

Motor actual current.

### *ActualFieldCurrent*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1883 | 0x3320.02 | - | - | U16 | 100 = 1A | RO | *YES* | - |

Motor actual field current (Id).

### *ActualTorqueCurrent*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1884 | 0x3320.03 | - | - | U16 | 100 = 1A | RO | *YES* | - |

Motor actual torque current (Iq).

### *OverCurrentAValue*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1885 | 0x3320.05 | - | - | U16 | 100 = 1A | RO | *YES* | - |

Motor A phase current in Power or motor over current conditions.

### *OverCurrentBValue*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1886 | 0x3320.06 | - | - | U16 | 100 = 1A | RO | *YES* | - |

Motor B phase current in Power or motor over current conditions.

### *OverCurrentCValue*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1887 | 0x3320.07 | - | - | U16 | 100 = 1A | RO | *YES* | - |

Motor C phase current in Power or motor over current conditions.

### RMSMotorCurrent

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1888 | 0x3320.08 | - | - | U16 | 100 = 1A | RO | *YES* | - |

Motor RMS current.

### RMSMotorCurrentFilter

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1889 | 0x3320.09 | - | 100 | U16 | 100 = 1s | RW | - | ES |

Filtering time to get the RMSMotorCurrent. RMSMotorCurrentFilter must be different from 0. Too low values of RMSMotorCurrentFilter can generate swinging in RMSMotorCurrent. Too high values of RMSMotorCurrentFilter slow down the convergence of RMSMotorCurrent. Writing in the parameter RMSMotorCurrentFilter, the time value of machine cycle the drive belongs to, it is possible to get a stable and convergent RMSMotorCurrent in ten machine cycles.

## AI0AcquiringStatus

Acquiring through the analog input 0.

### Number of entries

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x3330.00 | - | 2 | U8 | - | CST | - | - |

Number of parameters in this group. For further details see Section 17.1, *Capture*.

### AI0Voltage

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1930 | 0x3330.01 | - | - | S16 | mV | RO | *YES* | - |

Actual value of the not filtered analog input 0. The updating time of this parameter is CurrentLoopPeriod.

### AI0FilteredVoltage

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 1931 | 0x3330.02 | - | - | S16 | mV | RO | *YES* | - |

Actual value of the filtered analog input 0. The updating time of this parameter is MotionLoopPeriod.

# 27.13. Fault and Warning (2000-2199)

## FaultMask

Masks representing the Faults features. The relation between the masks bits and the Faults is shown in Table 24.1.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x3000.00 | 3 | 3 | U8 | - | CST | - | - |

Number of parameters in this group. For further details see Section 24.4, *Reaction to the Faults*

### *FaultMaskAutoErase*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2000 | 0x3000.01 | Desc | 0 | U32 | - | RW | - | ES |

Auto-restoring Fault mask the Fault Reset command is automatically run for. The Fault Reset runs when the Fault cause has been removed and there are no more retentive Faults. In Table 24.1 you can find the errors that can become auto-restoring through this parameter.

### *FaultMaskEnable*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2002 | 0x3000.02 | Desc | Desc | U32 | - | RW | - | ES |

Error mask the Fault signal has been enabled for. In Table 24.1 you can find the Faults that can be enabled/disabled through this parameter. All Fatal Faults, all bits for future uses and the Faults Real time mode error, CAN communication error, EtherCAT communication error, Position following error, User Fault, I2T limit reached and Logic voltage error are enabled by default.

⚠ NOTICE

**If the Fault signal is disabled, the related bits in the Fault registers will always be 0 and the drive will consequently not reach the Fault state because of that error cause.**

### *FaultMaskSafetyPrfExecute*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2004 | 0x3000.03 | Desc | 0x180 | U32 | - | RW | - | ES |

Fault mask generating the Safety profile. In Table 24.1 you can find the Faults that can generate the Safety profile.

### ***FaultReactionOptionCode***

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2049 | 0x605E.00 | Desc | -1 | S16 | - | RW | - | ES |

Action run in case of Non fatal Fault and with disabled Safety profile. For further information see Section 24.4, *Reaction to the Faults*.

| FaultReactionOptionCode | Action |
|---|---|
| -1 | The motor is stopped with maximum deceleration by resetting RequestedSpeed and then the drive reaches the *Fault* state. |
| 1 | The motor is stopped with deceleration equal to ProfileDeceleration and then the drive reaches the *Fault* state. |
| 2 | The motor is stopped with deceleration equal to QuickStopDeceleration and then the drive reaches the *Fault* state. |

## SafetyPrfConfiguration

Parameters to configure the Safety profile.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x300C.00 | 4 | 4 | U8 | - | CST | - | - |

Number of parameters in this group.

### *SafetyPrfTargetPosition*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2050 | 0x300C.01 | - | 0 | S32 | inc | RW | - | ES |

Absolute position target to reach when the safety profile is run.

### *SafetyPrfVelocity*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2052 | 0x300C.02 | Desc | Desc | U32 | inc/s | RW | - | ES |

Running speed of the safety profile. The default value and the adjustable maximum value are respectively 3.0 rad/s and 3216.9 rad/s. It cannot be set with a zero value.

### *SafetyPrfAcceleration*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2054 | 0x300C.03 | Desc | Desc | U32 | $inc/s^2$ | RW | - | ES |

Acceleration of the safety profile. The default value is 125.0 $rad/s^2$ while the range of the accepted values goes from 0.32 to 205887.3 $rad/s^2$.

### *SafetyPrfDeceleration*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2056 | 0x300C.04 | Desc | Desc | U32 | $inc/s^2$ | RW | - | ES |

Deceleration of the safety profile. The default value is 125.0 $rad/s^2$ while the range of the accepted values goes from 0.32 to 205887.3 $rad/s^2$.

### *SafetyPrfCommand*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2065 | 0x3010.01 | Desc | 0 | U16 | - | RW | - | - |

Command to enable/disable the safety profile. For further information see Section 24.4, *Reaction to the Faults*.

| SafetyPrfCommand | Description |
|---|---|
| 0 | Safety profile not enabled. |
| 1 | Safety profile enabled and configured with the same parameters of the *Profile Position Mode*. |
| 2 | Safety profile enabled and configured with the parameters defined in SafetyPrfConfiguration. The speed of the profile beginning and end are not valid. |

ADVICE

**It is advisable to choose 2 for SafetyPrfCommand to have more flexibility and less restrictions.**

## MainError

Drive main errors. The bit encoding is shown in Table 24.1.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x3014.00 | 4 | 4 | U8 | - | CST | - | - |

Number of parameters in this group.

### *WarnRetentive*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2070 | 0x3014.01 | Desc | 0 | U32 | - | RW | - | - |

Main retentive warnings.

### *WarnDynamic*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2072 | 0x3014.02 | Desc | 0 | U32 | - | RO | - | - |

Main dynamic warnings.

### *FaultRetentive*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2074 | 0x3014.03 | Desc | 0 | U32 | - | RO | *YES* | - |

Main retentive faults.

### *FaultDynamic*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2076 | 0x3014.04 | Desc | 0 | U32 | - | RO | *YES* | - |

Main dynamic faults.

### ***ErrorCode***

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2078 | 0x603F.00 | Desc | 0 | U16 | - | RO | *YES* | - |

Register that reports the error code of the last detected error. The possible codes are reported in the Table 9.3 for the CAN versions and in the Table 10.3 for the ETC versions. The parameter is reset on the Fault Reset command (see Table 8.13).

### ***RealTimeModeError***

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2080 | 0x3018.00 | Desc | 0 | U16 | - | RO | - | - |

Specific details of the Real time mode error.

### ***LimitReachedError***

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2081 | 0x3019.00 | Desc | 0 | U16 | - | RO | - | - |

Specific details of the Limit reached.

### ***ParamSoftError***

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2082 | 0x301A.00 | Desc | 0 | U16 | - | RO | - | - |

Specific details of the Parameters soft error.

### ***MotionParamLimitedError***

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2083 | 0x301B.00 | Desc | 0 | U16 | - | RO | - | - |

Specific details of the Motion parameter limited.

### **ThermalManageError**

Specific details of the Thermal management.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x302C.00 | 4 | 4 | U8 | - | CST | - | - |

Number of parameters in this group.

### *ThermalManageWarnRetentive*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2100 | 0x302C.01 | Desc | 0 | U16 | - | RO | - | - |

Details of the retentive Warnings of the Thermal management.

### *ThermalManageWarnDynamic*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2101 | 0x302C.02 | Desc | 0 | U16 | - | RO | - | - |

Details of the dynamic Warnings of the Thermal management.

### *ThermalManageFaultRetentive*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2102 | 0x302C.03 | Desc | 0 | U16 | - | RO | - | - |

Details of the retentive Faults of the Thermal management.

### *ThermalManageFaultDynamic*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2103 | 0x302C.04 | Desc | 0 | U16 | - | RO | - | - |

Details of the dynamic Faults of the Thermal management.

## ParamSeriousError

Specific details of the Parameters serious error.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x302D.00 | 4 | 4 | U8 | - | CST | - | - |

Number of parameters in this group.

### *ParamSeriousWarnDynamic*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2106 | 0x302D.02 | Desc | 0 | U16 | - | RO | - | - |

Details

of the dynamic Warnings of the Parameters serious error.

### *ParamSeriousFaultRetentive*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2107 | 0x302D.03 | Desc | 0 | U16 | - | RO | - | - |

Details of the retentive Faults of the Parameters serious error.

### *ParamSeriousFaultDynamic*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2108 | 0x302D.04 | Desc | 0 | U16 | - | RO | - | - |

Details of the dynamic Faults of the Parameters serious error.

## DigitalIoConfigError

Specific details of the Digital IO configuration error.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x302E.00 | 4 | 4 | U8 | - | CST | - | - |

Number of parameters in this group.

### *DigitalIoConfigWarnRetentive*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2110 | 0x302E.01 | Desc | 0 | U16 | - | RO | - | - |

Details of the retentive Warnings of the Digital IO configuration error.

### *DigitalIoConfigFaultRetentive*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2112 | 0x302E.03 | Desc | 0 | U16 | - | RO | - | - |

Details of the retentive Faults of the Digital IO configuration error.

### *DigitalIoConfigFaultDynamic*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2113 | 0x302E.04 | Desc | 0 | U16 | - | RO | - | - |

Details of the dynamic Faults of the Digital IO configuration error.

## UserError

Specific details of the User Fault.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x302F.00 | 4 | 4 | U8 | - | CST | - | - |

Number of parameters in this group.

### *UserFaultDynamic*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2115 | 0x302F.04 | Desc | 0 | U16 | - | RW | - | - |

Details of the dynamic Faults of the User Fault.

## LogicVoltageError

Specific details of the Logic voltage error.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x3030.00 | 4 | 4 | U8 | - | CST | - | - |

Number of parameters in this group.

### *LogicVoltageWarnRetentive*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2120 | 0x3030.01 | Desc | 0 | U16 | - | RO | - | - |

Details of the retentive Warnings of the Logic voltage error.

### *LogicVoltageWarnDynamic*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2121 | 0x3030.02 | Desc | 0 | U16 | - | RO | - | - |

Details of the dynamic Warnings of the Logic voltage error.

### *LogicVoltageFaultRetentive*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2122 | 0x3030.03 | Desc | 0 | U16 | - | RO | - | - |

Details of the retentive Faults of the Logic voltage error.

### *LogicVoltageFaultDynamic*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2123 | 0x3030.04 | Desc | 0 | U16 | - | RO | - | - |

Details of the dynamic Faults of the Logic voltage error.

## FeedbackSensorError

Specific details of the Feedback sensor error.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x3031.00 | 4 | 4 | U8 | - | CST | - | - |

Number of parameters in this group.

### *FeedbackSensorWarnRetentive*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2125 | 0x3031.01 | Desc | 0 | U16 | - | RO | - | - |

Details of the retentive Warnings of the Feedback sensor error.

### *FeedbackSensorWarnDynamic*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2126 | 0x3031.02 | Desc | 0 | U16 | - | RO | - | - |

Details of the dynamic Warnings of the Feedback sensor error.

### *FeedbackSensorFaultRetentive*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2127 | 0x3031.03 | Desc | 0 | U16 | - | RO | - | - |

Details of the retentive Faults of the Feedback sensor error.

### *FeedbackSensorFaultDynamic*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2128 | 0x3031.04 | Desc | 0 | U16 | - | RO | - | - |

Details of the dynamic Faults of the Feedback sensor error.

## STOError

Specific details of /STO Management Error.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x3032.00 | 2 | 2 | U8 | - | CST | - | - |

Number of parameters in this group.

### *STOFaultRetentive*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2132 | 0x3032.03 | Desc | 0 | U16 | - | RO | - | - |

Retentive Faults details of the /STO Management Error.

### *STOFaultDynamic*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2133 | 0x3032.04 | Desc | 0 | U16 | - | RO | - | - |

Dynamic Faults details of the /STO Management Error.

## InternalError

Specific details of Internal Error.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x303F.00 | 2 | 2 | U8 | - | CST | - | - |

Number of parameters in this group.

### *InternalErrorFaultRetentive*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2192 | 0x303F.03 | Desc | 0 | U16 | - | RO | - | - |

Details of the retentive Faults of the Internal Error

### *InternalErrorFaultDynamic*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2193 | 0x303F.04 | Desc | 0 | U16 | - | RO | - | - |

Details of the dynamic Faults of the Internal Error

# 27.14. CiA402 state machine (2400-2449)

## *Controlword*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2400 | 0x6040.00 | Desc | - | U16 | - | RW | *YES* | - |

Parameter to manage the CiA402 State Machine and the specific commands offered by the operating modes. For further details see what is reported in Section 8.5, *CiA402 state machine*. The bits are divided in this way:

- *Bit 0 - 3 and 7* to command each Transition of the CiA402 state machine.
- *Bit 8*: bits to manage the command of *Halt*.
- *Bit 4 - 6*: bits to request specific commands that can vary depending on the value of ModesOfOperationDisplay.
- *Bit 9 - 15*: bits not used.

For further details see what is reported in Section 8.5, *CiA402 state machine*.

### Statusword

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2401 | 0x6041.00 | Desc | - | U16 | - | RO | *YES* | - |

State of the CiA402 State Machine and of the specific commands of the operative modes. For further details see what is reported in Section 8.5, *CiA402 state machine*. In the following chart you can find the encoding of the status of the Statusword. The bits shown with 'x' are not important to determine the status.

| Statusword | Name | Description |
|---|---|---|
| xxxx xxxx x0xx 0000 | *Not Ready to Switch On* | Initializing |
| xxxx xxxx x1xx 0000 | *Switch On Disabled* | Idle |
| xxxx xxxx x01x 0001 | *Ready to Switch On* | Preparation to enabling |
| xxxx xxxx x01x 0011 | *Switched On* | The drive can be enabled or disabled, depending on the SwitchedOnOptionCode parameter value. |
| xxxx xxxx x01x 0111 | Operation enable | Drive enabled and possibility to command the motor motion |
| xxxx xxxx x00x 0111 | *Quick Stop Active* | Running a command of *Quick stop* |
| xxxx xxxx x0xx 1111 | *Fault Reaction Active* | Reaction to a Fault situation. The drive can be enabled or not, depending on the situation before the error occurred |
| xxxx xxxx x0xx 1000 | *Fault* | Fault state, finished reaction |

### QuickStopConfiguration

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2402 | 0x605A.00 | Desc | 6 | S16 | - | RW | - | ES |

Action run after a *Quick Stop* command.

| QuickStop-Configuration | Action |
|---|---|
| -1 | The motor is stopped with maximum deceleration by resetting RequestedSpeed and then the drive reaches the *Switch On Disabled* state. |
| 1 | The motor is stopped with deceleration equal to ProfileDeceleration and later the drive enters the *Switch On Disabled* state. |
| 2 | The motor is stopped with deceleration equal to QuickStopDeceleration and later the drive enters the *Switch On Disabled* state. |
| 5 | The motor is stopped with deceleration equal to ProfileDeceleration and the drive remains in the *Quick Stop Active* state. |
| 6 | The motor is stopped with deceleration equal to QuickStopDeceleration and the drive remains in the *Quick Stop Active* state. |

### SwitchedOnOptionCode

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2410 | 0x42E0.00 | Desc | 0 | S16 | - | RW | - | ES |

Value that determines the CiA402 state machine behaviour when it's in the *Switched On* state (see Section 8.5, “*CiA402 state machine*”).

| SwitchedOnOptionCode | Action |
|---|---|
| 0 | torque not present in the motor if the drive is in the *Switched On* state. |
| 1 | torque present in the motor if the drive is in the *Switched On* state. |

## 27.15. System manager (2450-2499)

### SysMngCommand

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2450 | 0x5FF7.01 | Desc | 0 | U16 | - | RW | - | - |

Command of the System Manager requested to the drive.

| SysMngCommand | Description |
|---|---|
| 0 | No command |
| 100 | End tuning command |
| 600 | End download parameters file |
| 601 | End export parameters file |
| 620 | End digital I/O setup |
| 1001 | Tuning: extended inertia estimator (requires the writing of ResetWatchdogTimeout) |
| 1002 | Tuning: inertia estimator (requires the writing of ResetWatchdogTimeout) |
| 1003 | Tuning: RL estimator (requires the writing of ResetWatchdogTimeout) |
| 1010 | Function Generator current D (requires the writing of ResetWatchdogTimeout) |
| 1015 | Function Generator current Q (requires the writing of ResetWatchdogTimeout) |
| 1020 | Function Generator velocity (requires the writing of ResetWatchdogTimeout) |
| 1030 | Function Generator position (requires the writing of ResetWatchdogTimeout) |
| 1040 | Function Generator profile velocity (requires the writing of ResetWatchdogTimeout) |
| 1050 | Function Generator profile position (requires the writing of ResetWatchdogTimeout) |
| 1101 | Set all loops, tuning and estimated parameters at default |
| 1102 | Parameter recalculation of all loops |
| 1103 | Parameter recalculation of motion loops |
| 1110 | Parameter recalculation of CurrentLoop |
| 1120 | Parameter recalculation of speed loop |
| 1130 | Parameter recalculation of position loop |
| 1140 | Parameter recalculation of flux weakening loop |
| 2001 | Permanent memory: save all parameters |
| 2200 | Permanent memory: restore to default of all parameters (permanent) |
| 2201 | Reset to default of all parameters (temporary) |

| SysMngCommand | Description |
|---|---|
| 2250 | Permanent memory: delete motor and sensor data |
| 2300 | Permanent memory: reload value of all parameters |
| 2301 | Permanent memory: reload value of loops parameters and tuning configuration |
| 5000 | Hard firmware reset |
| 5001 | Soft firmware reset |
| 5100 | Request download firmware |
| 5300 | Phasing of feedback position sensor |
| 5310 | Test phasing of feedback position sensor |
| 5320 | Save phasing of feedback position sensor |
| 5400 | Update ESI eeprom |
| 6000 | Downloading parameters file |
| 6001 | Export parameters file |
| 6200 | Setup digital I/O |
| 7200 | Start Analog input 0 offset calibration |
| 7201 | Start Analog input 0 gain calibration |
| 8000 | Request download PLC program |
| 8100 | Request erase PLC program |
| 8200 | Request upload PLC program |

## SysMngStatus

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2451 | 0x5FF7.02 | Desc | - | U16 | - | RO | - | - |

Status of the System Manager.

| SysMngStatus | Description |
|---|---|
| 5400 | Comparing of the EEPROM data with the expected ones |
| 5401 | EEPROM data writing in progress |
| 5402 | EEPROM data verification in progress |
| 5403 | Procedure closing phase |
| 5404 | The procedure is finished with an error |
| 5405 | The procedure is correctly finished (eeprom updated with new values) |
| 5406 | The procedure is correctly finished (eeprom verified, no data updated) |

NOTE

**The reported codes are valid only for the EtherCAT drives.**

## SysMngError

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2452 | 0x5FF7.03 | Desc | 0 | U16 | - | RO | - | - |

Error of the last command of the System Manager, requested to the drive.

| SysMngError | Message | Solution |
|---|---|---|
| 0 | No error | - |
| 1 | Unrecognized command | Check that the value that's written in the SysMngCommand parameter is present in the table that describes it. If the value is present and the error remains, update the firmware to the latest available version. |
| 2 | Test function are active | Please contact Bonfiglioli. |
| 3 | Test enabling key is required | The command cannot be executed by the user. |
| 4 | Enabling key is required | The command cannot be executed by the user. |
| 5 | Safety condition not satisfied: drive is enabled | Disable the drive before to give the command. |
| 6 | Generic time out | The command has taken more than the expected time. Repeat the command and check that's correctly executed. |
| 7 | No active command to stop | There are not active commands to stop. it requires the command interruption only if it is in progress. |
| 9 | Safety condition not satisfied: one or more digital outputs are active | Disable the digital outputs and try again. |
| 10 | Safety condition not satisfied: one or more digital outputs are not configured as Generic Output | Configure all the digital outputs as Generic Output (I/O X - Out X). |
| 11 | Safety condition not satisfied: capture units are active | Stop the capture peripherals. |
| 12 | Generic error during upload/download | Check the connection status and repeat the command. |
| 13 | Dynamic memory is busy | Stop the oscilloscope and try again. |
| 1000 | User has stopped the command | It has been required the interruption of the last command in progress. If the end of the command is not desired, check that the drive is not commanded by other master devices. |
| 1001 | Command watch dog is expired | Check if the connection is active and the ResetWatchdogTimeout parameter writing timing requiring. |
| 1002 | Switched on state has been required | If the Transition of the CiA402 State Machine is not desired, check that the drive is not commanded by other master devices (see Section 8.5, “*CiA402 state machine*”). |
| 1003 | Position limits are reached | Disengage the limit switch and repeat the command. Be sure that the required mechanical movement doesn't engage the limit switch. |
| 1004 | Quick stop has been required | If the Transition of the CiA402 State Machine is not desired, check that the drive is not commanded by other master devices (see Section 8.5, “*CiA402 state machine*”). |
| 1005 | Halt has been required | If the Transition of the CiA402 State Machine is not desired, check that the drive is not commanded by other master devices (see Section 8.5, “*CiA402 state machine*”). |
| 1006 | Disable has been required | If the Transition of the CiA402 State Machine is not desired, check that the drive is not commanded by other master devices (see Section 8.5, “*CiA402 state machine*”). |

| SysMngError | Message | Solution |
|---|---|---|
| 1007 | Drive is in Fault state | Verify the cause that has generated the Fault and, once it's solved, give a Reset command, see Table 8.13. |
| 1008 | Unknown transition has been required | If the Transition of the CiA402 State Machine is not desired, check that the drive is not commanded by other master devices (see Section 8.5, *"CiA402 state machine"*). |
| 1009 | Estimator torque is out of range | EstimatorTorque must be lower than ActualTorqueLimitP. |
| 1010 | Estimator speed is out of range | InertiaEstimatorVelocity must be lower than MaxMotorSpeed. |
| 1011 | Motor shaft is blocked | The applied load blocks the motor movement; check the mechanical. |
| 1012 | Servo mode is not active | Internal error, repeat the command or reset the drive (Hard reset). |
| 1013 | Tuning mode is not achievable | Internal error, repeat the command or reset the drive (Hard reset). Check if the drive is not piloted by other master devices. |
| 1014 | Motion is enable | The drive is in Operation enable; take the drive to the *Switched On* state. |
| 1015 | Motor and feedback sensor are not aligned | With incremental encoder, turn the motor for at least a half mechanical revolution. |
| 1016 | Estimated inertia is lower than motor inertia | Load with low inertia moment; try again to confirm the estimation. |
| 1017 | Estimate inertia is too high: bandwidth is limited | The inertia moment of the mechanical load doesn't allow to obtain a high VelocityLoopEstimatedBandwidth; try again to confirm the estimation. |
| 1018 | Estimated inertia limit reached | Inertia moment too large to be estimated. Retry to confirm the estimation; if the estimation is reliable, verify the good functioning of the mechanical. If the performances are not satisfactory, it's necessary to increase the motor size. |
| 1019 | Motor Parameters are not correctly set | Check the Warnings. |
| 1020 | Feedback position sensor is not set | Check the Warnings. |
| 1021 | Estimator security position limit reached | Decrease the value of InertiaEstimatorVelocity or increase the value of EstimatorTorque and retry. if the problem persists, try to execute a not extended estimation of the inertia moment (command 1002 in the place of command 1001 of the SysMngCommand). Check the mechanical parts work properly. If the problem persists, the estimator cannot be used, then try with what's described in Section 20.4, *Gains calculation*. |
| 1022 | Estimator torque greater than nominal current | Reduce EstimatorTorque. |
| 1023 | Estimator speed is too low: InertiaEstimatorVelocity is lower than 15rad/s. | Increase the value of InertiaEstimatorVelocity. |
| 1024 | Requested estimator speed is not reached | Retry the estimation; Try to execute a not extended estimation of the inertia moment (command 1002 in the place of command 1001 of the SysMngCommand). |
| 1025 | I2T Warning threshold reached | Increase I2TWarningThreshold. |

| SysMngError | Message | Solution |
|---|---|---|
| 1026 | DC bus voltage (+HV) is too low | Increase the supply voltage DCBusVoltage(+HV) and retry. |
| 2000 | Permanent memory address is out of range | Restore the permanent memory through the command 2200 (SysMngCommand). Reset the drive through the command 5000. |
| 2001 | Permanent memory data length is not valid | Restore the permanent memory through the command 2200 (SysMngCommand). Reset the drive through the command 5000. |
| 2002 | Message: "Dati non salvabili perché ADC di corrente non sono calibrati" | Please contact Bonfiglioli. |
| 2003 | Message: "Dati non salvabili perché ADC di tensione non sono calibrati" | Please contact Bonfiglioli. |
| 2004 | Last permanent memory writing was not completed correctly. Permanent memory data may be corrupted | Restore the permanent memory through the command 2200 (SysMngCommand). Reset the drive through the command 5000. |
| 2005 | Message: "Dati non salvabili perche' la calibrazione dell'Analog Input 0 non e' completa" | Completely execute the analog input calibration (see Section 17.2, *Calibration*). |
| 2100 | No error for file system | - |
| 2101 | File system ID not exist | Restore the permanent memory through the command 2200 (SysMngCommand). Reset the drive through the command 5000. If the problem persists, please contact Bonfiglioli. |
| 2102 | File system ID not present in permanent memory | |
| 2103 | File system data length mismatch | |
| 2104 | File system CRC is invalid | |
| 2105 | File system command is refused by I2C driver | |
| 2106 | File system dynamic memory is busy | |
| 2107 | File system dimension limit reached | |
| 2108 | File system ID is zero | |
| 2109 | File system data length is zero | |
| 2110 | File system operation not exist | |
| 2200 | No error for I2C driver | |
| 2201 | I2C driver is busy | |
| 2202 | I2C time out in reading | |
| 2203 | I2C time out in writing | |
| 2204 | I2C driver bus error | |
| 2205 | I2C driver has detected an odd permanent memory address | |
| 2300 | Hiperface internal memory: no error | - |
| 2301 | Hiperface internal memory: driver is busy | Turn the drive off and then on again. If the problem persists, please contact Bonfiglioli. |
| 2302 | Hiperface internal memory: operation not exist | Turn the drive off and then on again. If the problem persists, please contact Bonfiglioli. |
| 2303 | Hiperface internal memory: data size is too large | Turn the drive off and then on again. If the problem persists, please contact Bonfiglioli. |
| 2304 | Hiperface internal memory: for details, see Feedback sensor error | Check Feedback sensor error. |
| 2305 | Hiperface internal memory: dynamic memory is busy | Stop the oscilloscope and try again. Turn the drive off and then on again. If the problem persists, please contact Bonfiglioli. |
| 2306 | Hiperface internal memory: CRC is invalid | Check the wirings of the feedback sensor and that the FeedbackSensorTemperature is correct- |

| SysMngError | Message | Solution |
|---|---|---|
| | | ly read. Turn the drive off and then on again. Repeat the phasing and the saving. |
| 5000 | Module ID is not present in ID table | Turn the drive off and then on again. If the problem persists, please contact Bonfiglioli. |
| 5101 | Download: file corrupted | See the solution in FirmwareStatus (value 151). |
| 5103 | Download: file requires unsupported features | See the solution in FirmwareStatus (value 153). |
| 5104 | Download: file requires unsupported features | See the solution in FirmwareStatus (value 154). |
| 5105 | Download: file requires unsupported features | See the solution in FirmwareStatus (value 155). |
| 5106 | Download: file requires unsupported features | See the solution in FirmwareStatus (value 156). |
| 5107 | Download: file requires unsupported features | See the solution in FirmwareStatus (value 157). |
| 5108 | Download: file requires unsupported features | See the solution in FirmwareStatus (value 158). |
| 5111 | Download: file corrupted | See the solution in FirmwareStatus (value 161). |
| 5112 | Download: file corrupted | See the solution in FirmwareStatus (value 162). |
| 5113 | Download: file corrupted | See the solution in FirmwareStatus (value 163). |
| 5114 | Download: file corrupted | See the solution in FirmwareStatus (value 164). |
| 5115 | Download: file corrupted | See the solution in FirmwareStatus (value 165). |
| 5116 | Download: file requires unsupported features | See the solution in FirmwareStatus (value 166). |
| 5117 | Download: file requires unsupported features | See the solution in FirmwareStatus (value 167). |
| 5118 | Download: file requires unsupported features | See the solution in FirmwareStatus (value 168). |
| 5119 | Download: file requires unsupported features | See the solution in FirmwareStatus (value 169). |
| 5120 | Download: file requires unsupported features | See the solution in FirmwareStatus (value 170). |
| 5121 | Download: file requires unsupported features | See the solution in FirmwareStatus (value 171). |
| 5122 | Download: memory error | See the solution in FirmwareStatus (value 172). |
| 5123 | Download: file corrupted | See the solution in FirmwareStatus (value 173). |
| 5124 | Download: file corrupted | See the solution in FirmwareStatus (value 174). |
| 5125 | Download: memory error | See the solution in FirmwareStatus (value 175). |
| 5150 | Download: memory error | See the solution in FirmwareStatus (value 200). |
| 5151 | Download: memory error | See the solution in FirmwareStatus (value 201). |
| 5152 | Download: memory error | See the solution in FirmwareStatus (value 202). |
| 5153 | Download: memory error | See the solution in FirmwareStatus (value 203). |
| 5154 | Download: memory error | See the solution in FirmwareStatus (value 204). |
| 5160 | Download: memory error | See the solution in FirmwareStatus (value 210). |
| 5161 | Download: memory error | See the solution in FirmwareStatus (value 211). |
| 5162 | Download: memory error | See the solution in FirmwareStatus (value 212). |
| 5163 | Download: memory error | See the solution in FirmwareStatus (value 213). |
| 5164 | Download: memory error | See the solution in FirmwareStatus (value 214). |
| 5170 | Download: memory error | See the solution in FirmwareStatus (value 220). |
| 5171 | Download: memory error | See the solution in FirmwareStatus (value 221). |
| 5180 | Download: memory error | See the solution in FirmwareStatus (value 230). |
| 5300 | Feedback position sensor is not compatibile with command required. | It's not necessary to execute the required command. |
| 5301 | Feedback position sensor is not phased: angle error is out of range. | Turn the drive off and then on again. |

| SysMngError | Message | Solution |
|---|---|---|
| 5302 | Read/Write error of the internal memory of feedback position sensor. | Check the wirings of the feedback sensor and that the FeedbackSensorTemperature is correctly read. |
| 5400 | Update ESI eeprom: the EtherCAT master does not allow access to the ESI eeprom[a] | Check that the master is connected to the EtherCAT port, repeat the procedure according to the operation sequence. If the error persists, please contact Bonfiglioli. |
| 5401 | Update ESI eeprom: procedure internal error[a] | Turn the drive off and then on again, repeat the procedure according to the operation sequence. If the error persists, please contact Bonfiglioli. |
| 5402 | Update ESI eeprom: procedure in timeout[a] | Turn the drive off and then on again, repeat the procedure according to the operation sequence. If the error persists, please contact Bonfiglioli. |
| 5403 | Update ESI eeprom: error reading data[a] | Turn the drive off and then on again, repeat the procedure according to the operation sequence. If the error persists, please contact Bonfiglioli. |
| 5404 | Update ESI eeprom: error writing data[a] | Turn the drive off and then on again, repeat the procedure according to the operation sequence. If the error persists, please contact Bonfiglioli. |
| 6200 | Impossible to change the IO configuration because an IO overload Warning is active (Digital output overtemperature or overload) | Reset the Warning and try again. |
| 7100 | Message: "VGATE non presente (manca /STO o abilitazione software)" | Please contact Bonfiglioli. |

[a]This error is valid only for EtherCAT drives.

### SysMngEnForcing

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2455 | 0x5FF7.06 | 0 ÷ 1 | 0 | U16 | - | RW | - | - |

It enables to force the System manager safety conditions, for the next command of the System Manager requested to the drive (0 = disabled forcing, 1 = enabled forcing).

## 27.16. Capture peripherals (2800-2899)

⚠ NOTICE

**In this section the parameters of both the capture interfaces are contained (selectable through the CaptureInterfaceMode parameter). When an interface has been selected, the parameters that are related to the other one are ignored and any try to access them returns a Warning. For further details see Section 18.2, *Configuration interface selection*.**

IMPORTANT

**The parameters listed in this section are divided according to the interface to which they belong. Taking as reference the CaptureInterfaceMode parameter:**
**- the previous parameters are related to the CUSTOM interface**
**- The successive parameters are related to the CiA-402 interface**

## CaptureParam_A

Parameters of capture peripheral (A).

### Number of entries

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x4000.00 | 7 | 7 | U8 | - | CST | - | - |

Number of parameters in this group.

### CaptureUnitCommand_A

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2801 | 0x4000.01 | 0÷3 | 0 | U16 | - | RW | *YES* | - |

Commands for the capture peripheral A.

| UnitCommand | Description |
|---|---|
| 0 | *Stop*<br>Disabling the capture peripheral. |
| 1 | *Single Run*<br>Enabling the single capture. When the selected trigger event takes place (CaptureTriggerInput_A ), the selected values of the quantities to capture (CaptureSource0_A, CaptureSource1_A, CaptureSource2_A) are copied in the parameters CapturedValue0_A, CapturedValue1_A, CapturedValue2_A. After the capture, CaptureUnitState_A goes from *Single capture armed* to *Single stop*. The capture results will remain valid until the next rearmament. Any further trigger events will be ignored by the capture peripheral. |
| 2 | *Repetitive Run*<br>It enables the repeating capture. When the selected trigger event takes place (CaptureTriggerInput_A ), the selected values of the quantities to capture (CaptureSource0_A, CaptureSource1_A, CaptureSource2_A) are copied in the parameters CapturedValue0_A, CapturedValue1_A, CapturedValue2_A. After the capture, CaptureUnitState_A goes from *Repetitive capture armed* to *Repetitive capture done*. A further trigger event will cause a new capture by overwriting the capture values of the previous event. When at least one of the parameters of the results are read, CaptureUnitState_A goes to *Repetitive capture armed*. |
| 3 | *Capture now*<br>Enable the single capture and force the trigger. This mode is useful in debug phase to check that the configurations of the capture peripheral are correct, even if the physical source of the trigger is not available. The behaviour of the capture peripheral, in this case, is equal to that of the mode *Single Run*, with the only difference that the trigger event is simulated. |

***Table 27.6. Codes for UnitCommand***

### CaptureTriggerInput_A

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2802 | 0x4000.02 | Desc | 0 | U16 | - | RW | - | ES |

Trigger signal that causes the capture for the peripheral A.

| TriggerInput | Description |
|---|---|
| 0 | Digital input In9 |
| 1 | Digital input In8 |

| TriggerInput | Description |
|---|---|
| 3 | Index pulse of the feedback encoder |
| 4 | Index pulse of the master encoder |

***Table 27.7. TriggerInput codes.***

## CaptureTriggerEdge_A

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2803 | 0x4000.03 | 0 ÷ 2 | 0 | U16 | - | RW | - | ES |

Trigger edge that causes the capture for the peripheral A.

| TriggerEdge | Description |
|---|---|
| 0 | Falling edge |
| 1 | Rising edge |
| 2 | Both the edges |

***Table 27.8. Codes for TriggerEdge.***

## CaptureInhibitTime_A

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2804 | 0x4000.04 | 0-65500 | 0 | U16 | ms | RW | - | ES |

Inhibition time of the capture for the peripheral A, after the trigger event, to avoid repeated captures, in case of not clean triggers. When the trigger event happens, the peripheral executes the capture and then ignore the trigger signal for the specified time.
In case of capture on both edges (CaptureTriggerEdge_A = 2), CaptureInhibitTime_A is applied "for edge". In case of repetitive capture (CaptureUnitCommand_A = 2), when a capture event happens, the capture is inhibited for the time that has been set in this parameter in relation to the specific capture edge. The other capture edge is not inhibited until it will not happen at least one time.

## CaptureValidationFilterMode_A

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2809 | 0x4000.0A | 0-1 | 0 | U16 | - | RW | - | ES |

Filtering mode of the first quantity to capture for the peripheral A (0=symmetric, 1=asymmetric). In the symmetric mode the filtering value is expressed by the parameter CaptureActiveSlopeValidationFilter_A and it is the same for the edge carrying out the capture and for the restore edge. In the asymmetric mode, the filtering value applied to the capture edge is expressed by the parameter CaptureActiveSlopeValidationFilter_A, while the filtering value applied to the restore edge is expressed by the parameter CaptureRestoreSlopeValidationFilter_A. For further information on the working of this type of filter please see Section 18.4, *Filter on CaptureSource0_A and CaptureSource0_B*.

## CaptureRestoreSlopeValidationFilter_A

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2810 | 0x4000.09 | - | 0 | U32 | Desc | RW | - | ES |

Enables the filter on the value of the first quantity of the peripheral A for the capture edge not enabled (restoring). This parameter is valid only if CaptureValidationFilterMode_A is equal to 1 (asymmetric mode). When different, it is ignored. The value shows the filtering entity and the unit of measurement is the same of the first quantity of the peripheral A. For further information please see Section 18.4, *Filter on CaptureSource0_A and CaptureSource0_B*.

### *CaptureActiveSlopeValidationFilter_A*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2812 | 0x4000.08 | - | 0 | U32 | Desc | RW | - | ES |

Enables the filter on the value of the first quantity of the peripheral A for the capture edge. If CaptureValidationFilterMode_A is equal to 0 (symmetric mode), the filtering expressed in this parameter is applied on the capture edge and on the restore edge, on the contrary the filtering expressed by this parameter is applied only to the capture edge; the filtering set by CaptureRestoreSlopeValidationFilter_A is applied to the restore edge.

## CaptureState_A

Status of the capture peripheral (A).

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x4001.00 | 2 | 2 | U8 | - | CST | - | - |

Number of parameters in this group.

### *CaptureUnitState_A*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2800 | 0x4001.01 | - | 0 | U16 | - | RO | *YES* | - |

Status of the capture peripheral A.

| UnitState | Description |
|---|---|
| 0 | *Capture stop*<br>Capture peripheral in stop. The trigger source is ignored. |
| 1 | *Single capture armed*<br>The capture peripheral is waiting for the trigger event to carry out the capture of type *Single Run*. When the trigger event takes place, the requested data will be stored and the capture peripheral will enter the *Single stop* state. |
| 2 | *Repetitive capture armed*<br>The capture peripheral is waiting for the trigger event to carry out a capture of repetitive type. When the trigger event takes place, the requested data will be stored and the capture peripheral will enter the *Repetitive capture done* state. |
| 3 | *Single stop*<br>The capture peripheral captured the selected data after the trigger event. After the captured data have been read, the peripheral can be disabled. The trigger source is ignored. |
| 4 | *Repetitive capture done* |

| UnitState | Description |
|---|---|
| | The peripheral captured the selected data after the trigger event, every new trigger event causes a new capture of the selected data and the overwriting of the previous values. This possibility is not notified in any way by the drive. When at least one of the results of the capture is read, the peripheral enters the status *Repetitive capture armed*. |
| 19 | *Single capture done on falling edge*<br>The capture peripheral has captured the selected data after the trigger event. After the captured data have been read, the peripheral can be disabled. The trigger source is ignored. |
| 35 | *Single capture done on rising edge*<br>The capture peripheral has captured the selected data after the trigger event. After the captured data have been read, the peripheral can be disabled. The trigger source is ignored. |
| 51 | *Single capture done on both edges*<br>The capture peripheral has captured the selected data after the trigger event. After the captured data have been read, the peripheral can be disabled. The trigger source is ignored. |
| 20 | *Repetitive capture done on falling edge*<br>The peripheral captured the selected data after the trigger event, every new trigger event causes a new capture of the selected data and the overwriting of the previous values. This possibility is not notified in any way by the drive. When at least one of the results of the capture is read, the peripheral enters the status *Repetitive capture armed*. |
| 36 | *Repetitive capture done on rising edge*<br>The peripheral captured the selected data after the trigger event, every new trigger event causes a new capture of the selected data and the overwriting of the previous values. This possibility is not notified in any way by the drive. When at least one of the results of the capture is read, the peripheral enters the status *Repetitive capture armed*. |

***Table 27.9. Codes for UnitState.***

### *NumberCapturesRecorded_A*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2814 | 0x4001.02 | - | - | U16 | - | RO | *YES* | - |

Counter of the capture number of the capture peripheral A.

### *NumberCapturesRecordedRising_A*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2815 | 0x4001.03 | - | - | U16 | - | RO | *YES* | - |

Counter of the events that have been captured on the rising edge of the capture peripheral A.

### *NumberCapturesRecordedFalling_A*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2816 | 0x4001.04 | - | - | U16 | - | RO | *YES* | - |

Counter of the events that have been captured on the falling edge of the capture peripheral A.

## CapturedValues_Rising_A

Captured value on the rising edge of the capture peripheral (A).

### Number of entries

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x4007.00 | 3 | 3 | U8 | - | CST | - | - |

Number of parameters in this group.

### CapturedValueRising0_A

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2820 | 0x4007.01 | - | - | S32 | - | RO | *YES* | - |

32 bit value of the first variable of the capture peripheral A, captured on the rising edge.

### CapturedValueRising1_A

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2822 | 0x4007.02 | - | - | S32 | - | RO | *YES* | - |

32 bit value of the second variable of the capture peripheral A, captured on the rising edge.

### CapturedValueRising2_A

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2824 | 0x4007.03 | - | - | S32 | - | RO | *YES* | - |

32 bit value of the third variable of the capture peripheral A, captured on the rising edge.

## CapturedValues_Falling_A

Captured value on the falling edge of the capture peripheral (A).

### Number of entries

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x4008.00 | 3 | 3 | U8 | - | CST | - | - |

Number of parameters in this group.

### CapturedValueFalling0_A

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2826 | 0x4008.01 | - | - | S32 | - | RO | *YES* | - |

32 bit value of the first variable of the capture peripheral A, captured on the falling edge.

### CapturedValueFalling1_A

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2828 | 0x4008.02 | - | - | S32 | - | RO | *YES* | - |

32 bit value of the second variable of the capture peripheral A, captured on the falling edge.

### *CapturedValueFalling2_A*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2830 | 0x4008.03 | - | - | S32 | - | RO | *YES* | - |

32 bit value of the third variable of the capture peripheral A, captured on the falling edge.

## CaptureSources_A

Selection of the quantity to capture with the capture peripheral (A).

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x4003.00 | 3 | 3 | U8 | - | CST | - | - |

Number of parameters in this group.

### *CaptureSource0_A*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2835 | 0x4003.01 | Desc | 2 | U16 | - | RW | - | ES |

Code of the first quantity to capture for the peripheral A.

| CaptureX | Description |
|---|---|
| 0 | No quantity |
| 1 | AuxiliaryEncoderPosition |
| 2 | PositionActualValue |
| 4 | PositionFollowingError |

***Table 27.10. Codes of the quantities to capture.***

### *CaptureSource1_A*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2836 | 0x4003.02 | Desc | 1 | U16 | - | RW | - | ES |

Code of the second quantity to capture for the peripheral A. In Table 27.10 you can find the available codes.

### *CaptureSource2_A*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2837 | 0x4003.03 | Desc | 0 | U16 | - | RW | - | ES |

Code of the third quantity to capture for the peripheral A. In Table 27.10 you can find the available codes.

## CapturedValues_A

Capture peripheral (A) captured values, saved in memory locations of 1 Long size.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x4004.00 | 3 | 3 | U8 | - | CST | - | - |

Number of parameters in this group.

### *CapturedValue0_A*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2840 | 0x4004.01 | - | - | S32 | - | RO | *YES* | - |

Captured value of the first quantity for the peripheral A (4 bytes).

### *CapturedValue1_A*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2842 | 0x4004.02 | - | - | S32 | - | RO | *YES* | - |

Captured value of the second quantity for the peripheral A (4 bytes).

### *CapturedValue2_A*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2844 | 0x4004.03 | - | - | S32 | - | RO | *YES* | - |

Captured value of the third quantity for the peripheral A (4 bytes).

## CapturedValues_Word_A

Capture peripheral (A) captured values, saved in memory locations of 1 Word size.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x4005.00 | 3 | 3 | U8 | - | CST | - | - |

Number of parameters in this group.

### *CapturedValue0_Word_A*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x4005.01 | - | - | S16 | - | RO | *YES* | - |

Captured value of the first quantity for the peripheral A (2 bytes).

### *CapturedValue1_Word_A*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x4005.02 | - | - | S16 | - | RO | *YES* | - |

Captured value of the second quantity for the peripheral A (2 bytes).

### *CapturedValue2_Word_A*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x4005.03 | - | - | S16 | - | RO | *YES* | - |

Captured value of the third quantity for the peripheral A (2 bytes).

## CapturedValues_Byte_A

Capture peripheral (A) captured values, saved in memory locations of 1 Byte size.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x4006.00 | 3 | 3 | U8 | - | CST | - | - |

Number of parameters in this group.

### *CapturedValue0_Byte_A*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x4006.01 | - | - | S8 | - | RO | *YES* | - |

Captured value of the first quantity for the peripheral A (1 byte).

### *CapturedValue1_Byte_A*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x4006.02 | - | - | S8 | - | RO | *YES* | - |

Captured value of the second quantity for the peripheral A (1 byte).

### *CapturedValue2_Byte_A*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x4006.03 | - | - | S8 | - | RO | *YES* | - |

Captured value of the third quantity for the peripheral A (1 byte).

## CaptureState_B

Capture peripheral (B) status.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x4011.00 | 4 | 4 | U8 | - | CST | - | - |

Number of parameters in this group.

### *CaptureUnitState_B*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2850 | 0x4011.01 | - | 0 | U16 | - | RO | *YES* | - |

Status of the capture peripheral B. In Table 27.9 you can find the available codes.

### *NumberCapturesRecorded_B*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2864 | 0x4011.02 | - | - | U16 | - | RO | *YES* | - |

Counter of the capture number of the capture peripheral B.

### *NumberCapturesRecordedRising_B*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2865 | 0x4011.03 | - | - | U16 | - | RO | *YES* | - |

Counter of the events that have been captured on the rising edge of the capture peripheral B.

### *NumberCapturesRecordedFalling_B*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2866 | 0x4011.04 | - | - | U16 | - | RO | *YES* | - |

Counter of the events that have been captured on the falling edge of the capture peripheral B.

## CaptureParam_B

Parameters of capture peripheral (B).

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x4010.00 | 7 | 7 | U8 | - | CST | - | - |

Number of parameters in this group.

### *CaptureUnitCommand_B*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2851 | 0x4010.01 | 0÷3 | 0 | U16 | - | RW | *YES* | - |

Commands for the capture peripheral B. In Table 27.6 you can find the available codes.

### *CaptureTriggerInput_B*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2852 | 0x4010.02 | Desc | 0 | U16 | - | RW | - | ES |

Trigger signal that causes the capture for the peripheral B. In Table 27.7 you can find the available codes.

### CaptureTriggerEdge_B

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2853 | 0x4010.03 | 0 ÷ 2 | 0 | U16 | - | RW | - | ES |

Trigger edge that causes the capture for the peripheral B. In Table 27.8 you can find the available codes.

### CaptureInhibitTime_B

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2854 | 0x4010.04 | 0-65500 | 0 | U16 | ms | RW | - | ES |

Inhibition time of the capture for the peripheral B, after the trigger event, to avoid repeated captures, in case of not clean triggers. When the trigger event happens, the peripheral executes the capture and then ignore the trigger signal for the specified time.
In case of capture on both edges (CaptureTriggerEdge_B = 2), CaptureInhibitTime_B is applied "for edge". In case of repetitive capture (CaptureUnitCommand_B = 2), when a capture event happens, the capture is inhibited for the time that has been set in this parameter in relation to the specific capture edge. The other capture edge is not inhibited until it will not happen at least one time.

### CaptureValidationFilterMode_B

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2859 | 0x4010.0A | 0-1 | 0 | U16 | - | RW | - | ES |

Filtering mode of the first quantity to capture for the peripheral B (0=symmetric, 1=asymmetric). In the symmetric mode the filtering value is expressed by the parameter CaptureActiveSlopeValidationFilter_B and it is the same for the edge carrying out the capture and for the restore edge. In the asymmetric mode, the filtering value applied to the capture edge is expressed by the parameter CaptureActiveSlopeValidationFilter_B, while the filtering value applied to the restore edge is expressed by the parameter CaptureRestoreSlopeValidationFilter_B. For further information on the working of this type of filter please see Section 18.4, *Filter on CaptureSource0_A and CaptureSource0_B*.

### CaptureRestoreSlopeValidationFilter_B

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2860 | 0x4010.09 | - | 0 | U32 | Desc | RW | - | ES |

Enabling the filter on the value of the first quantity of the peripheral B for the capture edge not enabled (restoring). This parameter is valid only if CaptureValidationFilterMode_B is equal to 1 (asymmetric mode). When different, it is ignored. The value shows the filtering entity and the unit of measurement is the same of the first quantity of the peripheral B. For further information please see Section 18.4, *Filter on CaptureSource0_A and CaptureSource0_B*.

### *CaptureActiveSlopeValidationFilter_B*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2862 | 0x4010.08 | - | 0 | U32 | Desc | RW | - | ES |

Enabling the filter on the value of the first quantity of the peripheral B for the capture edge. If CaptureValidationFilterMode_B is equal to 0 (symmetric mode), the filtering expressed in this parameter is applied on the capture edge and on the restore edge, on the contrary the filtering expressed by this parameter is applied only to the capture edge; the filtering set by CaptureRestoreSlopeValidationFilter_B is applied to the restore edge.

## CapturedValues_Rising_B

Captured value on the rising edge of the capture peripheral (B).

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x4017.00 | 3 | 3 | U8 | - | CST | - | - |

Number of parameters in this group.

### *CapturedValueRising0_B*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2870 | 0x4017.01 | - | - | S32 | - | RO | *YES* | - |

32 bit value of the variable 0 that has been captured on the rising edge of the capture peripheral B.

### *CapturedValueRising1_B*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2872 | 0x4017.02 | - | - | S32 | - | RO | *YES* | - |

32 bit value of the variable 1 that has been captured on the rising edge of the capture peripheral B.

### *CapturedValueRising2_B*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2874 | 0x4017.03 | - | - | S32 | - | RO | *YES* | - |

32 bit value of the variable 2 that has been captured on the rising edge of the capture peripheral B.

## CapturedValues_Falling_B

Captured value on the falling edge of the capture peripheral (B).

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x4018.00 | 3 | 3 | U8 | - | CST | - | - |

Number of parameters in this group.

### *CapturedValueFalling0_B*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2876 | 0x4018.01 | - | - | S32 | - | RO | *YES* | - |

32 bit value of the variable 0 that has been captured on the falling edge of the capture peripheral B.

### *CapturedValueFalling1_B*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2878 | 0x4018.02 | - | - | S32 | - | RO | *YES* | - |

32 bit value of the variable 1 that has been captured on the falling edge of the capture peripheral B.

### *CapturedValueFalling2_B*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2880 | 0x4018.03 | - | - | S32 | - | RO | *YES* | - |

32 bit value of the variable 2 that has been captured on the falling edge of the capture peripheral B.

## CaptureSources_B

Selection of the quantity to capture with the capture peripheral (B).

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x4013.00 | 3 | 3 | U8 | - | CST | - | - |

Number of parameters in this group.

### *CaptureSource0_B*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2885 | 0x4013.01 | Desc | 2 | U16 | - | RW | - | ES |

Code of the first quantity to capture for the peripheral B. In Table 27.10 you can find the available codes.

### *CaptureSource1_B*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2886 | 0x4013.02 | Desc | 1 | U16 | - | RW | - | ES |

Code of the second quantity to capture for the peripheral B. In Table 27.10 you can find the available codes.

### *CaptureSource2_B*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2887 | 0x4013.03 | Desc | 0 | U16 | - | RW | - | ES |

Code of the third quantity to capture for the peripheral B. In Table 27.10 you can find the available codes.

## CapturedValues_B

Capture peripheral (B) captured values, saved in memory locations of 1 Long size.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x4014.00 | 3 | 3 | U8 | - | CST | - | - |

Number of parameters in this group.

### *CapturedValue0_B*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2890 | 0x4014.01 | - | - | S32 | - | RO | *YES* | - |

Captured value of the first quantity for the peripheral B (4 bytes).

### *CapturedValue1_B*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2892 | 0x4014.02 | - | - | S32 | - | RO | *YES* | - |

Captured value of the second quantity for the peripheral B (4 bytes).

### *CapturedValue2_B*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2894 | 0x4014.03 | - | - | S32 | - | RO | *YES* | - |

Captured value of the third quantity for the peripheral B (4 bytes).

## CapturedValues_Word_B

Capture peripheral (B) captured values, saved in memory locations of 1 Word size.

### Number of entries

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x4015.00 | 3 | 3 | U8 | - | CST | - | - |

Number of parameters in this group.

### CapturedValue0_Word_B

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x4015.01 | - | - | S16 | - | RO | *YES* | - |

Captured value of the first quantity for the peripheral B (2 bytes).

### CapturedValue1_Word_B

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x4015.02 | - | - | S16 | - | RO | *YES* | - |

Captured value of the second quantity for the peripheral B (2 bytes).

### CapturedValue2_Word_B

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x4015.03 | - | - | S16 | - | RO | *YES* | - |

Captured value of the third quantity for the peripheral B (2 bytes).

## CapturedValues_Byte_B

Capture peripheral (B) captured values, saved in memory locations of 1 Byte size.

### Number of entries

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x4016.00 | 3 | 3 | U8 | - | CST | - | - |

Number of parameters in this group.

### CapturedValue0_Byte_B

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x4016.01 | - | - | S8 | - | RO | *YES* | - |

Captured value of the first quantity for the peripheral B (1 byte).

### CapturedValue1_Byte_B

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x4016.02 | - | - | S8 | - | RO | *YES* | - |

Captured value of the second quantity for the peripheral B (1 byte).

### CapturedValue2_Byte_B

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x4016.03 | - | - | S8 | - | RO | *YES* | - |

Captured value of the third quantity for the peripheral B (1 byte).

### CaptureInterfaceMode

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2949 | 0x402F.00 | 0 ÷ 1 | 0 | U16 | - | RW | *YES* | ES |

Switch to select the interface to be used to manage the capture peripherals A and B (0=custom mode, 1=CiA-402 mode). See Section 18.2, *"Configuration interface selection"*

IMPORTANT

**It's not possible to change the interface if the capture functionality is active:**
**- in case CaptureInterfaceMode value is 0 (CUSTOM interface): if at least one of the CaptureUnitState_A and CaptureUnitState_B parameters value is 1**
**- in case CaptureInterfaceMode value is 1 (CiA-402 interface): if both the bits 0 and/ or both the bits 8 of TouchProbeFunction and of TouchProbeStatus are set to 1.**

### TouchProbeFunction

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2950 | 0x60B8.00 | Desc | 0 | U16 | - | RW | *YES* | - |

Settings of the capture peripherals A and B.

| Bit | Description |
|---|---|
| 0 | 0 = Capture peripheral A disable<br>1 = Capture peripheral A enable |
| 1 | 0 = Single capture<br>1 = Repetitive capture |
| 2 | 0 = Trigger on Digital Input A<br>1 = Trigger on Index pulse of the feedback encoder or PositionActualValue |
| 3 | Reserved |
| 4 | 0 = Capture on rising edge of the capture peripheral A disable<br>1 = Capture on rising edge of the capture peripheral A enable |
| 5 | 0 = Capture on falling edge of the capture peripheral A disable<br>1 = Capture on falling edge of the capture peripheral A enable |
| 6, 7 | At disposal of the user (e.g. for test) |
| 8 | 0 = Capture peripheral B disable<br>1 = Capture peripheral B enable |
| 9 | 0 = Single capture<br>1 = Repetitive capture |
| 10 | 0 = Trigger on Digital Input B<br>1 = Trigger on Index pulse of the feedback encoder or PositionActualValue |
| 11 | Reserved |

| Bit | Description |
|---|---|
| 12 | 0 = Capture on rising edge of the capture peripheral B disable<br>1 = Capture on rising edge of the capture peripheral B enable |
| 13 | 0 = Capture on falling edge of the capture peripheral B disable<br>1 = Capture on falling edge of the capture peripheral B enable |
| 14, 15 | At disposal of the user (e.g. for test) |

***Table 27.11. Meaning of the bits related to the capture peripherals function.***

## *TouchProbeStatus*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2951 | 0x60B9.00 | - | - | U16 | - | RO | *YES* | - |

Status of the capture peripherals A and B.

| Bit | Description |
|---|---|
| 0 | 0 = Capture peripheral A disabled<br>1 = Capture peripheral A enabled |
| 1 | 0 = No value captured on rising edge of the capture peripheral A<br>1 = Value captured on falling edge of the capture peripheral A |
| 2 | 0 = No value captured on falling edge of the capture peripheral A<br>1 = Value captured on rising edge of the capture peripheral A |
| 3 - 5 | Reserved |
| 6, 7 | At disposal of the user (e.g. for test) |
| 8 | 0 = Capture peripheral B disabled<br>1 = Capture peripheral B enabled |
| 9 | 0 = No value captured on rising edge of the capture peripheral B<br>1 = Value captured on falling edge of the capture peripheral B |
| 10 | 0 = No value captured on falling edge of the capture peripheral B<br>1 = Value captured on rising edge of the capture peripheral B |
| 11 - 13 | Reserved |
| 14, 15 | At disposal of the user (e.g. for test) |

***Table 27.12. Meaning of the bits related to the capture peripherals status.***

## TouchProbesSource

Selection of the capture source of the capture peripheral.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x60D0.00 | 2 | 2 | U8 | - | CST | - | - |

Number of parameters in this group.

### *TouchProbe1Source*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2952 | 0x60D0.01 | Desc | 1 | S16 | - | RW | *YES* | - |

Capture source of the capture peripheral A. he values that this parameter can take are reported in the following table.

| Value | Trigger Event |
|---|---|
| -1 | Master encoder Index pulse |
| 1 | Touch probe input 1 (Digital input In8) |
| 2 | Touch probe input 2 (Digital input In9) |
| 5 | Feedback encoder Index pulse |

***Table 27.13. Codes for the selection of the trigger event of the capture peripheral.***

### *TouchProbe2Source*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2953 | 0x60D0.02 | Desc | 0 | S16 | - | RW | *YES* | - |

Capture source of the capture peripheral B. The values that this parameter can take are reported in the Table 27.13.

### ***TouchProbePosition1PosValue***

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2960 | 0x60BA.00 | - | - | S32 | - | RO | *YES* | - |

32 bit value captured on the rising edge of the capture peripheral A.

### ***TouchProbePosition1NegValue***

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2962 | 0x60BB.00 | - | - | S32 | - | RO | *YES* | - |

32 bit value captured on the falling edge of the capture peripheral A.

### ***TouchProbePosition2PosValue***

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2964 | 0x60BC.00 | - | - | S32 | - | RO | *YES* | - |

32 bit value captured on the rising edge of the capture peripheral B.

### ***TouchProbePosition2NegValue***

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2966 | 0x60BD.00 | - | - | S32 | - | RO | *YES* | - |

32 bit value captured on the falling edge of the capture peripheral B.

### ***TouchProbe1PosEdgeCounter***

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2980 | 0x60D5.00 | - | - | U16 | - | RO | *YES* | - |

Counter of the number of the values that have been captured on the rising edge of the capture peripheral A.

### *TouchProbe1NegEdgeCounter*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2981 | 0x60D6.00 | - | - | U16 | - | RO | *YES* | - |

Counter of the number of the values that have been captured on the falling edge of the capture peripheral A.

### *TouchProbe2PosEdgeCounter*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2982 | 0x60D7.00 | - | - | U16 | - | RO | *YES* | - |

Counter of the number of the values that have been captured on the rising edge of the capture peripheral B.

### *TouchProbe2NegEdgeCounter*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 2983 | 0x60D8.00 | - | - | U16 | - | RO | *YES* | - |

Counter of the number of the values that have been captured on the falling edge of the capture peripheral B.

## 27.17. Feedback sensor (3000-3999)

### FeedbackSensor

Feedback sensor related values.

#### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x36C0.00 | 5 | 5 | U8 | - | CST | - | - |

Number of parameters in this group.

#### *FeedbackSensorResolution*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 3000 | 0x36C0.02 | - | 8000 | U32 | Desc | *RO* | - | - |

Feedback sensor resolution, valid only for the sensors that declare it. The unit of measurement depends on the motor type: rotary motor [count/rev], linear motor [count/PolePitch].

## *FeedbackSensorCode*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 3004 | 0x36C0.04 | - | - | U16 | - | *Desc* | - | EM |

Code of the feedback sensor

This parameter is only for reading.

NOTE

**If the value of this parameter is modified, then all the Non-resettable parameters reset.**

| SensorCode | Description |
|---|---|
| 100 | Absolute Hiperface encoder multiturn SEL37 16sin/rev, 4096rev. |
| 101 | Absolute Hiperface encoder multiturn SKM36 128sin/rev, 4096rev. |
| 150 | Absolute Hiperface encoder singleturn SEK37 16sin/rev. |

NOTE

**In the drives of the *i*BMD series with Hiperface encoders, the Index is simulated.**

NOTE

**The default value, once it has been modified, is no more restorable.**

## *FeedbackSensorFaultTemperatureThrs*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 3005 | 0x36C0.05 | - | - | U16 | °C | RO | - | EM |

Feedback sensor temperature Fault threshold. In case the FeedbackSensorFaultTemperatureThrs value is 65535, it means that the encoder overheating temperature alarm management is disabled. In this case the only monitoring that remains active is, therefore, the motor temperature one, that can be read through the MotorTemperature parameter, that protects even the sensor.

## *FeedbackSensorAbsMode*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 3010 | 0x36C0.08 | - | - | S16 | - | RW | - | EM |

Absolute sensor functioning mode (0=incremental, 1=absolute). This parameter only makes sense if the physical sensor installed on the system is an absolute sensor (encoder hiperface mono or multi turn, resolver 2P=1, pure SinCos 2P=1). Since this parameter determines the mode to reconstruct the position, when it is written through the vocabulary, it will even be immediately and automatically saved in the permanent memory, so that even in case of NMT / SOFT reset (see Section 21.8, *Resetting* ) there will be no undesired changes on the functioning mode.

NOTE

**As for the FeedbackSensorCode parameter, the default value, once it has been modified, is no more restorable.**

NOTE

**If the value of this parameter is modified, then all the Non-resettable parameters reset.**

## FeedbackSensorPhasing

Feedback sensor phasing parameters related values.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x36C2.00 | 3 | 3 | U8 | - | CST | - | - |

Number of parameters in this group.

### *FeedbackSensorPhasingStatus*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 3020 | 0x36C2.01 | - | - | U16 | - | RO | - | - |

Feedback sensor phasing status (0=not phased, 1=phasing in progress, 2=phased, 3=phasing error).

### *FeedbackSensorPhasingAngleTest*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 3021 | 0x36C2.02 | 0-3600 | - | S16 | 10 = 1deg | RW | - | - |

Value of the electric angle used to execute the phasing test through the command 5300 of the SysMngCommand.

### *FeedbackSensorPhasingAngleError*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 3022 | 0x36C2.03 | - | - | S16 | 10 = 1deg | RW | - | - |

Error of the electric angle used to execute the phasing test through the command 5300 of the SysMngCommand.

## HallFeedbackSensor

Hall sensors features related values.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x3680.00 | 2 | 2 | U8 | - | CST | - | - |

Number of parameters in this group.

### *HallValidationWindow*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 3066 | 0x3680.0E | - | - | S16 | Count | RO | - | - |

Accuracy of the Hall sensors of the feedback sensor, valid only for the sensors that declare it. If the sensor doesn't respect this accuracy, the *Phasing: Validation window not respected* warning is reported (see Table 24.13).

# 27.18. Motion (4000-4999)

## PositionResolution

Number of increments related to a motor axis revolution (for further details see Section 13.2, *Revolution resolution*). This parameter is calculated as EncoderIncrements / MotorRevolutions. The unit of measurement depends on the motor type: rotary motor [inc/rev], linear motor [inc/PolePitch].

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x608F.00 | 2 | 2 | U8 | - | CST | - | - |

Number of parameters in this group.

### *EncoderIncrements*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4000 | 0x608F.01 | 256 - 1048576 | 8000 | U32 | Desc | RW | - | ES |

Number of increments of the feedback position sensor, used to calculate the PositionResolution.

NOTE

**This parameter is part of the group Non-resettable on the reset parameters.**

### *MotorRevolutions*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4003 | 0x608F.02 | 1 | 1 | U32 | Desc | RW | - | - |

Parameter that's necessary to calculate the PositionResolution. For rotative motors, it indicates the motor revolutions number. For linear motors, it indicates the motor polar steps number. Actually, this parameter is set to 1, so the PositionResolution coincides with the value of EncoderIncrements.

## *Polarity*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4002 | 0x607E.00 | - | 0 | U8 | - | RW | *YES* | ES |

Motor axis rotation direction in which the position increments. For further details see Section 13.3, *Polarity*.

| Polarity | Name | Description |
|---|---|---|
| 0 | Forward | With rotary motor: positive direction clockwise towards the motor flange<br>With linear motor: the positive direction is defined by the motor constructor, supposing the correct wiring of the motor phases. |
| 192 | Reverse | With rotary motor: positive direction counterclockwise towards the motor flange<br>With linear motor: the positive direction is the opposite one than the direction defined by the motor constructor, supposing the correct wiring of the motor phases. |

NOTE

**This parameter is part of the group Non-resettable on the reset parameters.**

## *ModesOfOperation*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4100 | 0x6060.00 | Desc | 3 | S8 | - | RW | *YES* | ES |

Selection of the operating mode of drive functioning. The accepted values are shown in the Table 27.14.

| Value | Name | Description |
|---|---|---|
| 1 | *Profile Position Mode* | The drive runs positionings with profile and set points configured by the user. |
| 3 | *Profile Velocity Mode (CiA402)* | The drive runs speed motions with profile and set point configured by the user. |
| 4 | *Torque Mode* | The drive runs a motion by following a torque reference configured by the user. Only the current regulation loop is used. |
| 6 | *Homing Mode* | The drive runs a homing procedure (zero searching) with profile configured by the user. |
| 7 | *Interpolated Position Mode* | The drive runs a motion by following a position set point periodically set by the user (positionings in Real-time). |
| 8 | *Cyclic Synchronous Position Mode* | The drive runs a motion by following a position set point periodically set by the user (positionings in Real-time). |
| 9 | *Cyclic Synchronous Velocity Mode* | The drive runs a motion by following a velocity set point periodically set by the user (positionings in Real-time). |
| 10 | *Cyclic Synchronous Torque Mode* | The drive runs a motion by following a torque set point periodically set by the user (positionings in Real-time). |
| -113 | *Profile Velocity Mode (CUSTOM)* | The drive runs a parametrized motion as speed profile but internally checked by the drive even in its position. |
| -111 | *Profile Velocity AI Mode* | The drive runs speed motions with profile configured by the user and set point set through the analog input. |
| -101 | *Torque AI Mode* | The drive executes a movement because it's following a torque reference that's related to the analog input value. Only the current regulation loop is used. |

***Table 27.14. Accepted values for the ModesOfOperation parameter.***

NOTE

**It is possible to change ModesOfOperation only if the drive is not on Quick Stop Active or Fault Reaction Active state. If the drive is on Operation enable state, it's possible to change the operative mode only if the bit 4 of the Controlword is set to 0. The operative mode change, with this method, must be done with the motor stopped, and it's user's responsibility to be sure of this condition.**

### *ModesOfOperationDisplay*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4101 | 0x6061.00 | Desc | 3 | S8 | - | RO | *YES* | - |

Mode of operation active in the drive. The values that can be read are the same of the ModesOfOperation parameter (see Table 27.14) with, in addition, the *Tuning Mode* (-127) used for some tuning and configuration commands.

## ApplyModeOperation

Group of parameter to manage the on-the-fly change of the operative mode.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x42C0.00 | 9 | 9 | U8 | - | CST | - | - |

Number of parameters in this group.

### *ApplyModeOperationCommand*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4108 | 0x42C0.01 | Desc | - | S8 | - | RW | *YES* | - |

Selection of the operating mode for the functionality of on-the-fly mode change.

| ApplyModeOperationComand | Name |
|---|---|
| 1 | *Profile Position Mode* |
| 6 | *Homing Mode* |
| 7 | *Interpolated Position Mode* |
| -113 | *Profile Velocity Mode (CUSTOM)* |

NOTE

**It is possible to change operative mode through this parameter only when the drive is in the Operation enable state. For further details see Section 22.5, *On-the-fly operative mode change*.**

### *ApplyModeOperationStatus*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4109 | 0x42C0.02 | - | - | U8 | - | RO | *YES* | - |

Status of the on-the-fly mode change. The parameter shows if the change was carried out or not and it shows any possible error code.

| ApplyModeOperationStatus | Description |
|---|---|
| 0 | No error, change ApplyModeOperationCommand correctly run. |
| 1 | Error: parameter ApplyModeOperationCommand not correct. |
| 2 | Error: parameter 1 of ApplyModeOperationParameters not correct. |
| 3 | Error: parameter 2 of ApplyModeOperationParameters not correct. |
| 4 | Error: parameter 3 of ApplyModeOperationParameters not correct. |
| 5 | Error: parameter 4 of ApplyModeOperationParameters not correct. |
| 6 | Error: parameter 5 of ApplyModeOperationParameters not correct. |
| 7 | Error: parameter 6 of ApplyModeOperationParameters not correct. |
| 8 | Error: parameter 7 of ApplyModeOperationParameters not correct. |

### *ApplyModeOperationParameters*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4110 - 4122 | 0x42C0.03 - 0x42C0.09 | - | - | U32 | - | RW | *YES* | - |

Group of 7 parameters through which it is possible to set the on-the-fly operating mode change. The meaning of every parameter changes when ApplyModeOperationCommand changes. For further details please see Section 22.5, *On-the-fly operative mode change*.

### ***PositionValidationStatus***

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4150 | 0x42BF.00 | - | - | U32 | - | RW | - | - |

System encoder position validation: it indicates if the position of the sensor, here after listed, is valid for the application. Each bit is related to an Encoder type and must be set (logic state 1) from the user if the procedures that are necessaries to validate the position have been executed. With "validate" we mean that the position can be considered "aligned" with the value that is reported by the sensor (homing, position, ...) If for any motive (reset, turn off, polarity inversion, ...) the position is no more coherent, the corresponding bitis automatically reset. The causes that reset the bit are:

| Bit | Encoder | Cause |
|---|---|---|
| 0 | Feedback encoder | turn off, HardReset by SystemManager command (see SysMngCommand) EncoderIncrements or Polarityparameters value change |
| 3 | Selected auxiliary encoder | NMT reset, turn off, HardReset and SoftReset by SystemManager command (see SysMngCommand) |
| 4 | Real auxiliary encoder | NMT reset, turn off, HardReset and SoftReset by SystemManager command (see SysMngCommand) |
| 5 | Virtual auxiliary encoder | NMT reset, turn off, HardReset and SoftReset by SystemManager command (see SysMngCommand) |

### PositionActualValue

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4156 | 0x6064.00 | - | - | S32 | inc | RO | *YES* | - |

Current drive position.

NOTICE

**This parameter is part of the group Non-resettable parameters. If a firmware reset happens, and the feedback sensor is not absolute, the actual position, read by the PositionActualValue parameter, is maintained only if the code that's returned by the ResetCause parameter is 5 or 6. If the feedback sensor is absolute the actual position is kept even if the drive is turned off (within the sensor functioning range).**

### FollowingErrorWindow

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4160 | 0x6065.00 | - | Desc | U32 | inc | RW | *YES* | ES |

Positioning window (Fault threshold) to check the Error of position tracking (setting used only in the position modes). If PositionFollowingError remains over this threshold longer than FollowingErrorTimeOut, the system signals the Fault of following error, if enabled. The default value is equal to 64 motor revolutions.

### FollowingErrorWindowWarn

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4162 | 0x4282.01 | - | Desc | U32 | inc | RW | - | ES |

Position window (Warning threshold) to check the Error of position tracking (setting used only in the position modes). If PositionFollowingError remains above this threshold longer than FollowingErrorTimeOut, the system signals the Warning of following error. The default value is equal to 64 motor revolutions.

### PositionFollowingError

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4164 | 0x60F4.00 | - | - | S32 | inc | RO | *YES* | - |

Current value of the error of position following.

### FollowingErrorTimeOut

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4166 | 0x6066.00 | - | 0 | U16 | ms | RW | *YES* | ES |

Maximum time period during which the absolute value of PositionFollowingError can get over the error windows (FollowingErrorWindow and FollowingErrorWindowWarn) before respectively a fault or a Warning is notified. Setting used only in the position modes.

### PositionWindow

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4170 | 0x6067.00 | - | 0 | U32 | inc | RW | *YES* | ES |

Tolerance window on the Position reached target (setting used only in the position modes). Once the PositionActualValue reached the window and remains inside it for a time period equal to at least PositionWindowTime, the bit *Target reached* of the Statusword is set. Vice versa the same bit is immediately reset as soon as the difference between the two positions (target and current) gets over the window. If this parameter is set to 0, the position target is reached only if the theoretical value of the position (set point of the position loop) is equal to the position target for a time period at least equal to PositionWindowTime.

### PositionWindowTime

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4172 | 0x6068.00 | - | 10 | U16 | ms | RW | *YES* | ES |

Minimum time period to check the reaching of the final position. Setting used only in the position modes.

### VelocityActualValue

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4203 | 0x606C.00 | - | - | S32 | inc/s | RO | *YES* | - |

Actual drive velocity, filtered by the Sensor filter (see Figure 27.1.

### VelocityWindow

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4205 | 0x606D.00 | - | 0 | U16 | inc/s | RW | *YES* | ES |

Tolerance window on the Speed target reached (setting used only in the speed modes). Once the VelocityActualValue reached the window and remains inside it for a time period equal to at least VelocityWindowTime, the bit *Target reached* of the Statusword is set. Vice versa the same bit is immediately reset as soon as the difference between the two speeds (target and current) gets over the window. If this parameter is set to 0, the speed target is reached only if the theoretical value of the speed (set point of the speed loop) is equal to the speed target for a time period at least equal to VelocityWindowTime.

### VelocityThreshold

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4206 | 0x606F.00 | - | 0 | U16 | inc/s | RW | *YES* | ES |

Speed threshold to recognize Stopped motor (setting used only in the speed modes). Once the VelocityActualValue is decreased with a lower value than the threshold and it stays in this for a time period at least equal to VelocityThresholdTime, the bit *Speed* of the Statusword is set. Vice versa the same bit is immediately reset as soon as the current speed gets over the threshold. If this parameter is set to 0, the motor is considered as stopped only if the theoretical value of the speed (set point of the speed loop) is equal to 0 for a time period at least equal to VelocityThresholdTime.

### *VelocityWindowTime*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4208 | 0x606E.00 | - | 0 | U16 | ms | RW | *YES* | ES |

Minimum time period to check the reaching of the final speed. Setting used only in the speed modes.

### *VelocityThresholdTime*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4209 | 0x6070.00 | - | 0 | U16 | ms | RW | *YES* | ES |

Minimum time period to check the condition of the stopped motor. Setting used only in the speed modes.

### *TargetTorque*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4250 | 0x6071.00 | - | 0 | S16 | 10 = 1%IS | RW | *YES* | - |

Target torque the motor has to reach in *Torque Mode* or in *Cyclic Synchronous Torque Mode*.

### *MaxTorque*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4251 | 0x6072.00 | 0 ÷ 32767 | 1000 | U16 | 10 = 1%IS | RW | *YES* | ES |

Symmetrical limit of the torque reference [1000 = Rated current] set by the user. This value is taken as reference in case the TorqueLimitSelector parameter is set to 1.

### *ActualTorqueLimitP*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4252 | 0x420F.00 | - | - | U16 | 10 = 1%IS | RO | *YES* | - |

Actual torque positive limit [1000 = Rated current].

### *ActualTorque*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4253 | 0x6077.00 | - | - | S16 | 10 = 1%IS | RO | *YES* | - |

Applied torque value.

### *TorqueFilterFrequency*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4254 | 0x3321.01 | - | 80 | U16 | Hz | RW | - | ES |

Typical filter frequency to obtain the ActualFilteredTorque parameter.

### ActualFilteredTorque

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4255 | 0x4210.00 | - | - | S16 | 10 = 1%IS | RO | *YES* | - |

Filtered value of ActualTorque. The filter is a Low-pass filter of the first order with typical frequency equal to TorqueFilterFrequency.

### RequestedTorque

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4256 | 0x6074.00 | - | - | S16 | 10 = 1%IS | RO | *YES* | - |

Value of the torque that's required to the motor.

### ActualTorqueLimitN

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4257 | 0x4211.00 | - | - | U16 | 10 = 1%IS | RO | *YES* | - |

Actual torque negative limit [1000 = Rated current].

### PositiveTorqueLimitValue

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4258 | 0x60E0.00 | - | - | U16 | 10 = 1%IS | RW | *YES* | ES |

Torque reference positive limit [1000 = Rated current].

### NegativeTorqueLimitValue

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4259 | 0x60E1.00 | - | - | U16 | 10 = 1%IS | RW | *YES* | ES |

Torque reference negative limit [1000 = Rated current].

### TorqueLimitSelector

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4270 | 0x4202.00 | - | 0 | U16 | - | RW | *YES* | ES |

Selector that allows to select the source to obtain the torque limit.

| TorqueLimitSelector | Description |
|---|---|
| 0 | Peak Current (the torque limit is always active and is equal to UserPeakCurrent) |
| 1 | Max/Positive/Negative (only for ETC version)<br>Max Torque (only for CAN version) Torque limit is active at the MaxTorque value [a] |

| TorqueLimitSelector | Description |
|---|---|
| 2 | Torque limit enabled through analog input (see Section 17.3, *Conversion*) |

[a]In this case the limit can be symmetrical by setting the MaxTorque parameter, or asymmetrical by setting the PositiveTorqueLimitValue and NegativeTorqueLimitValue values.

## *TorqueSlope*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4280 | 0x6087.00 | 0xFFFFFFFF | 0xFFFFFFFF | U32 | - | RW | *YES* | - |

Torque increment per second in the torque profile ramps. This parameter is not used.

## *TorqueProfileType*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4282 | 0x6088.00 | -1 | -1 | S16 | - | RO | *YES* | - |

Torque profile type: -1=Torque step.

## *PositioningOptionCode*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4300 | 0x60F2.00 | Desc | 0 | U16 | - | RW | *YES* | ES |

Necessary options to define the behaviour of the position operating modes. Only the bits related to the behaviour of the *Profile Position Mode* have been currently implemented:

- ***Relative option* (bit 0 - 1)**
  Group of bits used to check the positioning behaviour when it is of relative type (the bit *Absolute / Relative* of the Controlword is equal to 1). The accepted values are:
  0: the positioning is run in relation to the last position target (absolute internal).
  1: the positioning is carried out in relation to the set-point of the position loop.
  2: the positioning is carried out in relation to the PositionActualValue.
- ***Change immediately option* (bit 2 - 3)**
  Group of bits used to check the positioning behaviour when the bit *Change set immediately* of the Controlword is set as equal to 1. The only accepted value is 0 showing that the positioning carried out in this mode immediately readjusts the current motion to the new parameters of the position profile.
- ***Request-response option* (bit 4 - 5)**
  Group of bits used to check the handshake between the drive and the master to start the positionings. The only accepted value is 0 showing that the supported handshake is the standard one, described in Section 22.9, *Profile Position Mode*.

## *ProfilePositionStatus*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4319 | 0x42A0.00 | 0 - 6 | - | U16 | - | RO | *YES* | - |

Status of the position trajectory generator.

| ProfilePositionStatus | Description |
|---|---|
| 0 | The trajectory generator is disabled: ModesOfOperationDisplay is not *Profile Position Mode* or the drive is disabled. |
| 1 | The position profile is in the *stationary phase*: steady references. From this status on ModesOfOperationDisplay is always equal to *Profile Position Mode*, the drive is always in the status Operation enable and the trajectory generator is always enabled. |
| 2 | The position profile is in the *Deceleration phase for direction reversal*. This is the first phase carried out when the motor, already moving, has to reverse the motion. In this phase the deceleration used is always equal to ProfileDeceleration. |
| 3 | The position profile is in the *Acceleration phase*. This is the phase carried out when the trajectory generator must accelerate the motor (or decelerate depending on the set parameters) to reach the cruise speed ProfileVelocity. |
| 4 | The position profile is in the *Cruise phase*. This phase is carried out at a constant speed equal to ProfileVelocity. |
| 5 | The position profile is in the *Deceleration phase*. This phase is carried out when the trajectory generator must decelerate the motor in order to reach the speed EndVelocity. |
| 6 | The position profile is in the *Phase of profile end*. This phase is carried out at a constant speed equal to EndVelocity before stopping the positioning. When this phase is over the trajectory generator enters the status 1, *Stationary phase*. |

## *TargetPosition*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4320 | 0x607A.00 | - | 0 | S32 | inc | RW | *YES* | - |

Target position that must be reached at the end of a positioning carried out in *Profile Position Mode* or in *Cyclic Synchronous Position Mode*. In *Cyclic Synchronous Position Mode* the value is always read as absolute; in *Profile Position Mode* the value is read as absolute or relative depending on the *Absolute / Relative* bit of the Controlword. If the positioning is of absolute type the TargetPosition is understood as a position related to the Zero position, vice versa if the positioning is of relative type the TargetPosition is understood as shown in the bits *Relative option* of the parameter PositioningOptionCode.

NOTE

**If in Profile Position Mode, in the parametrisation of a positioning, the TargetPosition value even includ the space to be travelled with speed equal to EndVelocity at the end of the profile, that is EndIncrements. However if this last value is higher than TargetPosition the motion is carried out with speed equal to EndVelocity and the content of the parameter EndIncrements is ignored.**

## *EndIncrements*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4322 | 0x4284.01 | - | 0 | U32 | inc | RW | *YES* | ES |

Space to cover with speed equal to EndVelocity at the end of the deceleration ramp of a position profile. If the value of EndVelocity is 0 the value of EndIncrements is ignored and the positioning ends after the deceleration ramp.

### *HomeOffset*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4324 | 0x607C.00 | - | 0 | S32 | inc | RW | *YES* | ES |

Difference between Zero position and Home position. Please see Section 22.19, *Homing Mode*.

NOTE

**If the sensor is absolute then the value of HomeOffset must be between 0 and the negative full scale value (for example in a drive that has a single turn absolute encoder of 4096 imp/rev, HomeOffset must be between 0 and -(1*4096).**

### *PositionLimitEnable*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4326 | 0x4280.01 | 0 - 1 | 0 | U16 | - | RW | - | ES |

Parameter used to enable/disable the software position limits. See Section 19.6, *Limits of software position*.

### SoftwarePositionLimit

Software position limits, for details see Section 19.6, *Limits of software position*.

#### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x607D.00 | 2 | 2 | U8 | - | CST | - | - |

Number of parameters in this group.

#### *PositionLimitNegative*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4327 | 0x607D.01 | - | -2147483648 | S32 | inc | RW | *YES* | ES |

Negative software position limit.

#### *PositionLimitPositive*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4329 | 0x607D.02 | - | 2147483647 | S32 | inc | RW | *YES* | ES |

Positive software position limit.

### *MaxProfileVelocity*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4331 | 0x607F.00 | Desc | Desc | U32 | inc/s | RW | *YES* | ES |

Maximum speed of the trajectory generators. Its default value is 100% of MotorRatedSpeed while the maximum settable value is 3216.9 rad/s.

### MaxMotorSpeed

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4333 | 0x6080.00 | - | Desc | U32 | Desc | RW | *YES* | ES |

Motor velocity limit value. The default value is equal to the 120% of MotorRatedSpeed. The unit of measurement depends on the motor type: rotary motor [rpm], linear motor [mm/s].

### ProfileVelocity

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4335 | 0x6081.00 | Desc | Desc | U32 | inc/s | RW | *YES* | ES |

Running speed of the position profile. At the end of the acceleration ramp the motor reaches this speed which is kept until the beginning of the deceleration ramp. If the value of ProfileVelocity is lower than StartVelocity or EndVelocity, its value is internally set with the higher between the two. The default value and the adjustable maximum value are respectively 3.0 rad/s and 3216.9 rad/s. It cannot be set with a zero value.

### EndVelocity

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4337 | 0x6082.00 | Desc | 0 | U32 | inc/s | RW | *YES* | ES |

Speed which the motor has to reach at the end of the profile deceleration ramp. The maximum settable value is 3216.9 rad/s.

### ProfileAccceleration

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4339 | 0x6083.00 | Desc | Desc | U32 | inc/s | RW | *YES* | ES |

Value of the profile acceleration ramp. The default value is 125.0 rad/s$^2$ while the range of accepted values goes from 0.32 to 205887.3 rad/s$^2$.

### ProfileDeceleration

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4341 | 0x6084.00 | Desc | Desc | U32 | inc/s$^2$ | RW | *YES* | ES |

Profile deceleration ramp value. The default value is 125.0 rad/s$^2$ while the range of the accepted values goes from 0.32 to 205887.3 rad/s$^2$.

### QuickStopDeceleration

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4343 | 0x6085.00 | Desc | Desc | U32 | inc/s$^2$ | RW | *YES* | ES |

Quick stop ramp deceleration value. This deceleration is used in the following cases:

- the drive is in the status Operation enable with parameter QuickStopConfiguration equal to 2 or 6 and gets a command of *Quick Stop*.
- The drive is in the status Operation enable, is carrying out a motion and it reaches or gets over a position limit (hardware or software).

The default value is equal to the parameter MaxDeceleration while the range of admitted value goes from 0.32 to 205887.3 rad/$s^2$.

### *MotionProfileType*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4345 | 0x6086.00 | 0 | 0 | S16 | - | RW | *YES* | ES |

Type of acceleration and deceleration ramps used to create the profile. It is currently possible to use only linear ramps (trapeze profile).

### *StartVelocity*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4347 | 0x4244.00 | Desc | 0 | U32 | inc/s | RW | *YES* | ES |

Motor speed at the beginning of the profile. The maximum settable value is 3216.9 rad/s.

### *HomingMethod*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4500 | 0x6098.00 | Desc | 35 | S8 | - | RW | *YES* | ES |

Method used to start the homing procedure. Fur further details please see Section 22.19, *Homing Mode*.

⚠ WARNING

**If the Polarity is of Reverse type, then the roles of Positive limit switch (FC +) and Negative limit switch (FC -) are reversed: Positive limit switch (FC +) behaves like Negative limit switch (FC -) and Negative limit switch (FC -) behaves like Positive limit switch (FC +). This is true both in the text of this manual and in the MotionDrive.**

⚠ WARNING

**If a HomingMethod (see also Table 22.3) with mechanical stop is selected (e.g. mode -1), remember to set the torque limit (see Section 19.3, *Torque limit*).**

| Value | Procedure description |
|---|---|
| 1 | FC- sw, Index pulse |
| 2 | FC+ sw, Index pulse |
| 7 | Home sw, dir +, Index before rise edge, Fc+ reverse |
| 8 | Home sw, dir +, Index after rise edge, Fc+ reverse |
| 9 | Home sw, dir +, Index before fall edge, Fc+ reverse |
| 10 | Home sw, dir +, Index after fall edge, Fc+ reverse |
| 11 | Home sw, dir -, Index before rise edge, Fc+ reverse |
| 12 | Home sw, dir -, Index after rise edge, Fc+ reverse |

| Value | Procedure description |
|---|---|
| 13 | Home sw, dir -, Index before fall edge, Fc+ reverse |
| 14 | Home sw, dir -, Index after fall edge, Fc+ reverse |
| 17 | FC- sw, no Index |
| 18 | FC+ sw, no Index |
| 23 | Home sw, dir +, rise edge, no Index, Fc+ reverse |
| 26 | Home sw, dir +, fall edge, no Index, Fc+ reverse |
| 27 | Home sw, dir -, rise edge, no Index, Fc+ reverse |
| 30 | Home sw, dir -, fall edge, no Index, Fc+ reverse |
| 35 | Current position |
| -1 | Mechanical stop, dir -, Index pulse |
| -2 | Mechanical stop, dir +, Index pulse |
| -7 | Home sw, dir +, Index before rise edge, Fc+ stop |
| -8 | Home sw, dir +, Index after rise edge, Fc+ stop |
| -9 | Home sw, dir +, Index before fall edge, Fc+ stop |
| -10 | Home sw, dir +, Index after fall edge, Fc+ stop |
| -11 | Home sw, dir -, Index before rise edge, Fc- stop |
| -12 | Home sw, dir -, Index after rise edge, Fc- stop |
| -13 | Home sw, dir -, Index before fall edge, Fc- stop |
| -14 | Home sw, dir -, Index after fall edge, Fc- stop |
| -17 | Mechanical stop, dir -, no Index |
| -18 | Mechanical stop, dir +, no Index |
| -23 | Home sw, dir +, rise edge, no Index, Fc+ stop |
| -26 | Home sw, dir +, fall edge, no Index, Fc+ stop |
| -27 | Home sw, dir -, rise edge, no Index, Fc- stop |
| -30 | Home sw, dir -, fall edge, no Index, Fc- stop |
| -35 | Current RequestedPosition |

### *IndexPulseDeadZone*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4501 | 0x4285.02 | Desc | Desc | U32 | inc | RW | - | ES |

Position offset where the drive, after detecting the disengagement of the Home switch or of the limit switch (depending on the chosen method), does not control the Index pulse of the feedback sensor. It represents the measurement of the dead zone after which the drive starts looking for the Index pulse. The resolution of IndexPulseDeadZone is 1° rounded off, the range goes from 0° to 180° and the default value is 1°.

### HomingSpeeds

Speeds used in the *Homing Mode*.

#### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x6099.00 | 2 | 2 | U8 | - | CST | - | - |

Number of parameters in this group.

### *SpeedForSwitch*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4504 | 0x6099.01 | Desc | Desc | U32 | inc/s | RW | *YES* | ES |

Speed used during the search phase of the Home switch or of the limit switch (depending on the chosen method), in the homing procedure. The default value and the maximum settable value are respectively 1.5 rad/s and 3216.9 rad/s. It cannot be set with a zero value.

### *SpeedForZero*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4506 | 0x6099.02 | Desc | Desc | U32 | inc/s | RW | *YES* | ES |

Speed used during the search phase of the Home position. The default value and the maximum settable value are respectively 0.4 rad/s and 3216.9 rad/s. It cannot be set with a zero value.

### ***HomingAcceleration***

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4510 | 0x609A.00 | Desc | Desc | U32 | inc/s$^2$ | RW | *YES* | ES |

Value of the accelerations and decelerations used during the homing procedures. The default value is 15.0 rad/s$^2$ while the range of the accepted values goes from 0.32 to 205887.3 rad/s$^2$.

### ***HomingPosDisengagement***

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4512 | 0x4285.03 | Desc. | Desc. | U32 | inc | RW | - | ES |

Minimum disengagement space used in the mechanical stop HomingMethod without the Index pulse searching, described on Table 22.3 (after the reaching of the mechanical stop, the drive reverses its direction and distances the motor from the mechanical stop of at least of the number of pulses that are written in this parameter). The value of this parameter depends on the feedback sensor resolution and can take the values from 0 to a maximum of 2048*resolution/revolution and by default it's 1/8 of the feedback sensor resolution. For example, if the feedback sensor is 8000 pul/rev then HomingPosDisengagement=2048*8000=16384000 with default=1/8*8000=1000.

### ***HomingAbsRangeMode***

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4514 | 0x4285.04 | Desc. | Desc. | U16 | inc | RW | - | ES |

Selector of the allowed positions range, with absolute encoder (0=from 0 to encoder range, 1=from -1/2 range to +1/2 range). It has to be defined if the homing has to be executed with “zero at centre” or “zero at side”. At the end of the homing procedure the drive, that has an absolute encoder (FeedbackSensorAbsMode = 1), will save the data so that the position can be reconstructed even if the drive is turned off and on again (provided that the axis position is within the range that has been defined for the homing, range that has been defined with this parameter).

### HomingStatus

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4530 | 0x42A1.00 | - | - | S16 | - | RO | *YES* | - |

Homing mode status (see details in the Table 27.15).

| Value | Description |
|---|---|
| -2 | Homing running. |
| -1 | The drive is saving the data in the permanent memory. |
| 0 | The homing has been disabled and/or interrupted (aborted) by another command (e.g. bit *Halt*, reset bit *Homing operation start*, ..., see Table 8.14). |
| 1 | Homing correctly executed. |
| 17 | Error: It has not been found the Home switch between 2 limit switch (it concerns the HomingMethod with limit switches management, see also Table 22.3) and the axis is moving. |
| 18 | Error: it has been reached a limit switch before the Home switch has been found (it regards the HomingMethod without limit switch management, see also Table 22.3) and the axis is on moving. |
| 20 | Error: The Index pulse has already been programmed as capture trigger in the **Capture B** peripheral and the axis is on moving (see IndexPulseNote). |
| 21 | Error: during the homing procedure when it has been selected a mode with the mechanical stop management, the axis is on moving and it has not been activated the torque limit (see Section 19.3, "*Torque limit*"). |
| 49 | Same as 17. Differences: the axis is stationary. |
| 50 | Same as 18. Differences: the axis is stationary. |
| 52 | Same as 20. Differences: the axis is stationary. |
| 53 | Same as 21. Differences: the axis is stationary. |

***Table 27.15. Accepted values for the HomingStatus parameter.***

⚠ NOTICE

**This parameter is part of the group Non-resettable parameters. If a firmware reset happens, and the feedback sensor is not absolute, the homing status, read by the HomingStatus parameter, is maintained only if the code that's returned by the ResetCause parameter is 5 or 6. If the feedback sensor is absolute the homing status is maintained even if the drive is off.**

NOTE

**In some cases the reset of this parameter can also be done by the modification of one of the Non-resettable on the reset parameters.**

### CyclicSynchronousSubMode

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4540 | 0x42D0.00 | Desc | -136 | S16 | - | RW | *YES* | - |

This parameter is used only to change the interpolation type that the drive has to execute when one of the synchronous cyclic modes is active (see Section 22.11, *Cyclic Synchronous Position Mode*, Section 22.12, *Cyclic Synchronous Velocity Mode*, Section 22.13, *Cyclic Synchronous Torque Mode*).
The new sub-mode is applied when the operative mode is changed by writing the value of the new operative mode in the ModesOfOperation. Otherwise the previous sub-mode remains active.
This parameter is applied even if an operative mode change doesn't happen, in case the drive executes a transition to switch the state machine CiA-402 to the Operation enable status: in particular, it is referred to the transitions 4 and 16 (see Table 8.13).

| ModesOfOperationDisplay | CyclicSynchronousSubMode | Interpolation type | Auto KVff[a] | Auto KAff[b] |
|---|---|---|---|---|
| 8 | -136 | none | 0 | 0 |
| | -135 | | 0 | 1 |
| | -132 | | 1 | 0 |
| | -131 | | 1 | 1 |
| | -144 | linear | 0 | 0 |
| | -143 | | 0 | 1 |
| | -140 | | 1 | 0 |
| | -139 | | 1 | 1 |
| | -148 | cubic | 1 | 0 |
| | -147 | | 1 | 1 |
| 9 | -136, -132 | none | - | 0 |
| | -135, -131 | | - | 1 |
| | -144, -140 | linear | - | 0 |
| | -143, -139 | | - | 1 |
| 10 | -136, -135, -132, -131 | none | - | - |
| | -144, -143, -140, -139 | linear | - | - |

[a] 0 = the feed forward velocity is calculated according to the VelocityOffset parameter; 1 = the feed forward velocity calculation is automatically executed internally of the drive; - = not applicable.
[b] 0 = the feed forward acceleration is calculated according to the TorqueOffset parameter; 1 = the feed forward acceleration calculation is automatically executed internally of the drive; - = not applicable.

***Table 27.16. Available interpolation modes.***

### Sub-modes application example

- the drive is turned-on and the ModesOfOperation and CyclicSynchronousSubMode cells values are the defaults, respectively 3 and -136;
- write -139 in the parameter (the new sub-mode is selected but not applied);
- write 8 (CSP) in the parameter (the new sub-mode is applied for the CSP mode);
- write -136 in the parameter (the new sub-mode is selected but not applied);
- write 9 (CSV) in the parameter (the new sub-mode is applied for the CSV mode);
- write 8 (CSP) in the parameter (the new sub-mode is applied for the CSP mode);

## *PositionOffset*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4541 | 0x60B0.00 | - | 0 | S32 | inc | RW | *YES* | - |

Position offset that will be added to the TargetPosition if the *Cyclic Synchronous Position Mode* is active (see Section 22.11, *Cyclic Synchronous Position Mode*).

### *VelocityOffset*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4543 | 0x60B1.00 | Desc | 0 | S32 | inc/s | RW | *YES* | - |

Velocity offset that will be added to the TargetVelocity if the *Cyclic Synchronous Velocity Mode* is active (see Section 22.12, *Cyclic Synchronous Velocity Mode*). Instead, in case of *Cyclic Synchronous Position Mode*, it can be used as velocity feed forward or velocity reference, depending on the interpolation mode (see CyclicSynchronousSubMode). The range of accepted values goes from -3216.9 to 3216.9 rad/s.

### *TorqueOffset*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4545 | 0x60B2.00 | - | 0 | S16 | 10 = 1%IS | RW | *YES* | - |

Torque offset that will be added to the TargetTorque if the *Cyclic Synchronous Torque Mode* is active (see Section 22.13, *Cyclic Synchronous Torque Mode*). Instead, in case of *Cyclic Synchronous Position Mode* or *Cyclic Synchronous Velocity Mode* it can be used as acceleration feed forward (vedere CyclicSynchronousSubMode).

### *IpPosSubModeSelect*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4550 | 0x60C0.00 | Desc | 0 | S16 | - | RW | *YES* | - |

Selector of the interpolation type the drive must run when the *Interpolated Position Mode* is enabled (see Section 22.10, *Interpolated Position Mode*).

| IpPosSub-ModeSelect | Description |
|---|---|
| 0 | Linear interpolation. |
| -1 | Cubic interpolation. |
| -10 | Linear interpolation with speed feed forward. |

## IpTimePeriod

Time gap in which the interpolation data must be transmitted. the final value must be calculated with the following formula: IpTimePeriodValue * $10^{\text{IpTimeIndex}}$[s]. This parameter can be used alternatively to the CommunicCyclePeriod (CAN) or CommunicCyclePeriod (ETC) parameter. If these parameters are used at the same time, they must express the same value.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x60C2.00 | 2 | 2 | U8 | - | CST | - | - |

Number of parameters in this group.

The validation is done as follows:

- With EtherCAT drives, in Soft sync mode : If IpTimePeriod or CommunicCyclePeriod have been implemented with an out of range or not equal value, a configuration error is reported. If IpTimePeriod is right and CommunicCyclePeriod is set to 0, then CommunicCyclePeriod takes the value of IpTimePeriod. If IpTimePeriod has not been set (default) and CommunicCyclePeriod has a correct value, CommunicCyclePeriod is copied in IpTimePeriod. If both the values are within the limits and equals each other, the CommunicCyclePeriod parameter value is taken as cycle time.
- With EtherCAT drives in Hard sync mode : the CommunicCyclePeriod parameter is ignored, in its place the value that's written in the 0x09A0 address, in the ET1100 registers area, is inserted. If IpTimePeriod or ET1100 are set with an out of range or not equal value, a configuration error is reported. If IpTimePeriod has not been set and ET1100 has a correct value, ET1100 is copied in IpTimePeriod. If both the values are within the limits and equals each other, the ET1100 value is taken as cycle time.
- With CANopen drives : If IpTimePeriod or CommunicCyclePeriod have been set with an out of range or not equal value, a configuration error is reported. If IpTimePeriod has not been set (default) and CommunicCyclePeriod has a correct value, CommunicCyclePeriod is copied in IpTimePeriod. If both the values are within the limits and equals each other, the CommunicCyclePeriod parameter value is taken as cycle time.

IMPORTANT

**For the CAN version drives, the parameters of the IpTimePeriod group are writable only if the drive has not the NMT state machine in OPERATIONAL state.**

**For the ETC version drives, the paraemters of the IpTimePeriod group are writable only if the drive has not the EtherCAT state machine in SAFE-OPERATIONAL or OPERATIONAL state.**

### *IpTimePeriodValue*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4551 | 0x60C2.01 | 0-255 | 1 | U8 | s | RW | - | - |

First multiplication factor to calculate IpTimePeriod.

### *IpTimePeriodIndex*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4552 | 0x60C2.02 | -6 ÷ -2 | -3 | S8 | - | RW | - | - |

Exponent (base 10) for the calculation of IpTimePeriod.

## IpPosDataRecord

Group of parameters tat defines the set-point in the *Interpolated Position Mode*. It consists of IpPosFirstParameter and IpPosSecondParameter.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x60C1.00 | 2 | 2 | U8 | - | CST | - | - |

Number of parameters in this group.

### *IpPosFirstParameter*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4560 | 0x60C1.01 | - | 0 | S32 | inc | RW | *YES* | - |

This parameter is the first of a data group that are used all together to define the set-point the interpolator must reach. It contains the position value which must be reached. The parameter is valid only for *Interpolated Position Mode*.

### *IpPosSecondParameter*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4562 | 0x60C1.02 | - | 0 | S32 | Desc | RW | *YES* | - |

This parameter is the second of a data group that are used all together to define the set-point the interpolator must reach. It contains the speed which must be reached. The parameter is valid only for *Interpolated Position Mode* and is expressed in [65536 = 1inc/$T_{SYNC}$].

## IpPosDataConfig

Group of parameters that define the IpPosDataRecord configuration.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x60C4.00 | 6 | 6 | U8 | - | CST | - | - |

Number of parameters in this group.

### *IpPosDataConfigMaxBufferSize*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4592 | 0x60C4.01 | 1 | 1 | U32 | - | RW | - | - |

IpPosDataRecord configuration: maximum number of buffer points.

### *IpPosDataConfigActualBufferSize*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4594 | 0x60C4.02 | 1 | 1 | U32 | - | RW | - | - |

IpPosDataRecord configuration: actual number of buffer points.

### *IpPosDataConfigBufferOrganization*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4596 | 0x60C4.03 | 0 | 0 | U8 | - | RW | - | - |

IpPosDataRecord configuration: it specifies the buffer organization (FIFO type buffer).

### *IpPosDataConfigBufferPosition*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4597 | 0x60C4.04 | 0 | 0 | U16 | - | RW | - | - |

IpPosDataRecord configuration: index of IpPosDataRecord used for the interpolation (index of the first available point).

### *IpPosDataConfigSizeDataRecord*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4598 | 0x60C4.05 | Desc | - | U8 | - | WO | - | - |

IpPosDataRecord configuration: it expresses the dimension in bytes of IpPosDataRecord (dimension in byte of the single point). It can take the values 4 or 8, depending on the IpPosSubModeSelect parameter; if IpPosSubModeSelect=0 then IpPosDataConfigSizeDataRecord=4, else IpPosDataConfigSizeDataRecord=8

### *IpPosDataConfigBufferClear*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4599 | 0x60C4.06 | 1 | 1 | U8 | - | RW | - | - |

IpPosDataRecord configuration: it specifies that the IpPosDataRecord is enabled (it can't be disabled).

### ***MaxAcceleration***

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4600 | 0x60C5.00 | Desc | Desc | U32 | inc/s$^2$ | RW | *YES* | ES |

Maximum deceleration used to run a profile. The default value is 205887.3 rad/s$^2$ while the range of the accepted values goes from 0.32 to 205887.3 rad/s$^2$.

### ***MaxDeceleration***

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4602 | 0x60C6.00 | Desc | Desc | U32 | inc/s$^2$ | RW | *YES* | ES |

Maximum deceleration used to run a profile. The default value is 205887.3 rad/s$^2$ while the range of the accepted values goes from 0.32 to 205887.3 rad/s$^2$.

### ***TargetVelocity***

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4700 | 0x60FF.00 | Desc | 0 | S32 | inc/s | RW | *YES* | - |

Target velocity the motor has to reach in *Profile Velocity Mode (CiA402)*, in *Profile Velocity Mode (CUSTOM)* or in *Cyclic Synchronous Velocity Mode*. If the drive is programmed with the *Profile Velocity Mode (CiA402)* or the *Profile Velocity Mode (CUSTOM)* and its status is Operation enable, the writing of a velocity that has an absolute value greater than EndVelocity and StartVelocity starts the movement, while the writing of a velocity that has an absolute value lower or equal stops the movement. The sign of the speed written in this parameter defines the motion direction. The range of accepted values goes from -3216.9 to 3216.9 rad/s.

### *SupportedDriveModes*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 4800 | 0x6502.00 | 0x3ED | 0x3ED | U32 | - | RO | - | ES |

Seupported CiA-402 standard operative modes: csp (*Cyclic Synchronous Position Mode*), csv (*Cyclic Synchronous Velocity Mode*), cst (*Cyclic Synchronous Torque Mode*), ip (*Interpolated Position Mode*), hm (*Homing Mode*), tq (*Torque Mode*), pv (*Profile Velocity Mode (CiA402)*), pp (*Profile Position Mode*).

## 27.19. Brake (5000-5199)

### Brake

Group of parameter to manage the brake.

#### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x36D0.00 | 3 | 3 | U8 | - | CST | - | - |

Number of parameters in this group.

#### *BrakeReleaseTime*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 5000 | 0x36D0.01 | - | Desc | U16 | ms | RW | - | EM |

Brake release delay time.

#### *BrakeCloseTime*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 5001 | 0x36D0.02 | - | Desc | U16 | ms | RW | - | EM |

Brake response time.

#### *BrakeStatus*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 5002 | 0x36D0.03 | - | Desc | U16 | - | RW | - | - |

Brake status. This parameter is writable only when the motor is disabled.. The values that it can assume are: 0=Brake close (the brake is closed and ther motor is braked), 1=Brake release (the brake is released and ther motor is not braked).

# 27.20. Auxiliary position sensor (6500-6549)

## RealEncParam

Group of parameters for the real master sensor functionalities configuration.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x36C9.00 | 3 | 3 | U8 | - | CST | - | - |

Number of parameters in this group.

### *RealEncoderPosition*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6500 | 0x36C9.01 | - | 0 | S32 | cnt | RW | *YES* | - |

Position reached by the physical auxiliary position sensor. Through a writing operation it's possible to modify this value.

### *RealEncoderVelocity*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6502 | 0x36C9.03 | - | 0 | S32 | cnt/s | RO | *YES* | - |

Velocity reached by the physical auxiliary position sensor.

### *RealEncoderPolarity*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6504 | 0x36C9.02 | - | 0 | U16 | - | RW | - | ES |

Polarity of the physical sensor: Rotation direction where the position value increases: 0=Forward, 1=Reverse.

## VirtualEncParam

Group of parameters for the virtual master sensor functionalities configuration.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x36C8.00 | 3 | 3 | U8 | - | CST | - | - |

Number of parameters in this group.

### *VirtualEncoderPosition*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6510 | 0x36C8.01 | - | 0 | S32 | cnt | RW | *YES* | - |

Position reached by the virtual auxiliary position sensor.

### *VirtualEncoderVelocity*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6512 | 0x36C8.03 | - | 0 | S32 | Desc | RW | *YES* | - |

Velocity of virtual auxiliary sensor expressed in [65536 = 10000cnt/s] (e.g.. to obtain 2500 cnt/s the value to be inserted is 16384).

### *VirtualEncoderRunStop*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6514 | 0x36C8.02 | 0-1 | 0 | U8 | - | RW | - | - |

Run/stop command of the virtual sensor (0=stop, 1=run).

## AuxiliaryEncParam

Group of parameters for the selected auxiliary master sensor functionalities configuration.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x36CA.00 | 3 | 3 | U8 | - | CST | - | - |

Number of parameters in this group.

### *AuxiliaryEncoderPosition*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6520 | 0x36CA.01 | - | - | S32 | cnt | RO | *YES* | - |

Position reached by the selected auxiliary position sensor.

### *AuxiliaryEncoderSpeed*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6522 | 0x36CA.03 | - | - | S32 | cnt/s | RO | *YES* | - |

Velocity reached by the selected auxiliary position sensor.

### *AuxiliaryEncoderSelector*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6524 | 0x36CA.02 | 0-1 | 0 | U16 | - | RW | - | - |

Auxiliary sensor selection command: 0=real encoder, 1=virtual encoder.

# 27.21. Digital inputs and outputs (6550-6999)

## DigitalInputFunctStatus

Status of digital inputs if programmed as FC, Home, Enable.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x4054.00 | 3 | 3 | U8 | - | CST | - | - |

Number of parameters in this group.

### *FcStatus*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6558 | 0x4054.01 | - | - | U16 | - | RO | - | - |

Status of the programmed inputs with the function of positive and negative limit switch.

WARNING

**If the Polarity is of Reverse type, then the roles of Positive limit switch (FC +) and Negative limit switch (FC -) are reversed: Positive limit switch (FC +) behaves like Negative limit switch (FC -) and Negative limit switch (FC -) behaves like Positive limit switch (FC +). This is true both in the text of this manual and in the MotionDrive.**

| FcStatus | Description |
|---|---|
| Bit 0 | Status of the programmed input with the functionality Negative limit switch (FC -) |
| Bit 1 | Status of the programmed input with the functionality Positive limit switch (FC +) |

### *HomeStatus*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6559 | 0x4054.02 | - | - | U16 | - | RO | - | - |

Status of the programmes input with the function of Home

### *EnableInputStatus*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6560 | 0x4054.03 | - | - | U16 | - | RO | - | - |

Status of the programmed input with the function of Enable. If no input is associated to the Enable function, EnableInputStatus is always equal to 1.

### ***DisableOption***

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6561 | 0x406E.00 | - | 2 | S16 | - | RW | - | ES |

Disabling options through the digital input configured as Enable.

| DisableOption | Description |
|---|---|
| -1 | The motor is stopped with maximum deceleration by resetting to zero RequestedSpeed |
| 2 | The motor is stopped with deceleration equal to QuickStopDeceleration |

### *DisableOkOutput*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6570 | 0x406F.00 | - | 1041 | U32 | - | RW | - | ES |

Mask used to select the Faults disabling the digital output Drive Ok (Drv OK): 0 = the Fault, even if present, does not disable the output; 1 = the Fault disables the output. The relation between the mask bits and the Faults is shown in Table 24.1.

## PwmHwParam

Group of parameters for the PWM output configuration.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x403F.00 | 6 | 6 | U8 | - | CST | - | - |

Number of parameters in this group.

### *PwmHwFrequencyIO0*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6580 | 0x403F.01 | 1-50000 | 1000 | U16 | Hz | RW | - | - |

Frequency of the signal generated by the I/O 0 when programmed as output pwm.

### *PwmHwDutyCycleIO0*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6581 | 0x403F.02 | 0-100 | 0 | U16 | % | RW | - | - |

Duty Cycle of the signal generated by the I/O 0 when programmed as output pwm. The value 0 means the output is disabled while the value 100 means the output is enabled.

### *PwmHwFrequencyIO1*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6582 | 0x403F.03 | 1-50000 | 1000 | U16 | Hz | RW | - | - |

Frequency of the signal generated by the I/O 0 when programmed as output pwm.

### *PwmHwDutyCycleIO1*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6583 | 0x403F.04 | 0-100 | 0 | U16 | % | RW | - | - |

Duty Cycle of the signal generated by the I/O 0 when programmed as output pwm. The value 0 means the output is disabled while the value 100 means the output is enabled.

### *PwmHwFrequencyIO2*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6584 | 0x403F.05 | 1-50000 | 1000 | U16 | Hz | RW | - | - |

Frequency of the signal generated by the I/O 0 when programmed as output pwm.

### *PwmHwDutyCycleIO2*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6585 | 0x403F.06 | 0-100 | 0 | U16 | % | RW | - | - |

Duty Cycle of the signal generated by the I/O 0 when programmed as output pwm. The value 0 means the output is disabled while the value 100 means the output is enabled.

## *LogicalDigitalInputStatus*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6600 | 0x4051.01 | - | - | U32 | - | RO | *YES* | - |

Logical status of the digital inputs, after having applied PolarityInputValue.

| Bit | Resource |
|---|---|
| 0-15 | Reserved |
| 16 | IN/OUT 0 |
| 17 | IN/OUT 1 |
| 18 | IN/OUT 2 |
| 19 | IN/OUT 3 |
| 20 | IN 4 |
| 21 | IN 5 |
| 22 | IN 6 |
| 23 | IN 7 |
| 24 | IN 8 |
| 25 | IN 9 |
| 26 | /STO |
| 27-31 | Reserved |

***Table 27.17. Association of the bits of the parameter LogicalDigitalInputStatus of the I/O of iBMD.***

## *DigitalInputs*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6602 | 0x60FD.00 | - | - | U32 | - | RO | *YES* | - |

Physical status of the digital inputs, for the description of the single bits see Table 27.17.

### PhysicalOutputs

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6604 | 0x60FE.01 | - | Desc | U32 | - | RW | *YES* | - |

Status of the digital outputs. The default value of the digital outputs which have the Generic Output (I/O X - Out X) functionality is 0 (output off).

| Bit | Resource |
|---|---|
| 0-15 | Reserved |
| 16 | IN/OUT 0 |
| 17 | IN/OUT 1 |
| 18 | IN/OUT 2 |
| 19 | IN/OUT 3 |
| 20 | OUT 4 |
| 21 | OUT 5 |
| 22 | OUT 6 |
| 23-31 | Reserved |

***Table 27.18. Association of the bits of the parameter PhysicalOutputs of the I/O of iBMD.***

### DigitalOutputsBitMask

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6606 | 0x60FE.02 | - | 0XFFFFFFFF | U32 | - | RW | *YES* | - |

Enabling mask of the writing of the parameter PhysicalOutputs : 1=writing enabled, 0=writing not enabled. For a description of the single bits see Table 27.18

### DebounceTime

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6608 | 0x405F.00 | 250-65000 | 250 | U16 | µs | RW | - | ES |

Filtering time of the digital inputs. See Filtering of the digital inputs

### EnableDebounce

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6609 | 0x405E.00 | - | 0 | U16 | - | RW | - | ES |

Enabling mask of the filtering on the selected inputs. See Filtering of the digital inputs

### PolarityInputValue

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6610 | 0x405A.00 | - | 0 | U16 | - | RW | *YES* | ES |

This parameter is used to define which inputs must work by reversed logics. As default the input is on the logical status 1 when there is power on that. This setting has no effect if the input is programmed as limit switch. See Selection of the polarity of the digital inputs for further details.

| Bit | Resource |
|---|---|
| 0 | IN/OUT 0 |
| 1 | IN/OUT 1 |
| 2 | IN/OUT 2 |
| 3 | IN/OUT 3 |
| 4 | IN 4 |
| 5 | IN 5 |
| 6 | IN 6 |
| 7 | IN 7 |
| 8 | IN 8 |
| 9 | IN 9 |
| 7-15 | Reserved |

***Table 27.19. Association of the bits of the parameter PolarityInputValue of the I/O of i*BMD.**

## *TerminationResistance*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6611 | 0x405B.00 | - | 0 | U16 | - | RW | *YES* | ES |

This parameter is used to enable the termination resistances for the resources at differential logics. See Termination resistances for further details.

| Bit | Resource |
|---|---|
| 0 | IN/OUT 0 |
| 1 | IN/OUT 1 |
| 2 | IN/OUT 2 |
| 3-15 | Reserved |

***Table 27.20. Association of the bits of the parameter TerminationResistance of the I/O of i*BMD.**

## *IO_0_Function*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6700 | 0x4070.01 | - | 1 | U16 | - | RW | - | ES |

Functionality given to the I/O 0.
The codes of the functionalities are listed in Table 27.21. In the Table 16.6 you can find the functionalities which can be given to the single resources.

| Functionalities | Code |
|---|---|
| Generic Input (I/O X - In X) | 1 |
| Generic Output (I/O X - Out X) | 2 |
| Fault (Fault) | 3 |
| Home | 7 |
| Step | 8 |
| Dir | 9 |
| Positive limit switch (FC +) | 10 |
| Negative limit switch (FC -) | 11 |
| Enable | 13 |

| Functionalities | Code |
|---|---|
| Quadrature Input ChA (Ch A) | 14 |
| Quadrature Input ChB (Ch B) | 15 |
| Quadrature Input Index (Idx) | 16 |
| Pwm out (Pwm O) | 17 |
| Motor Fan (M. Fan) | 19 |
| Drive Fan (D. Fan) | 20 |
| Drive Ok (Drv OK) | 22 |
| Simulated 24V Out (S24V) | 23 |
| Simulated GND (SGND) | 24 |

***Table 27.21. Codes related to the functionalities which can be given to the I/O resources***

⚠ NOTICE

**Simulated GND (SGND) input is not protected against overcurrent.**

## IO_1_Function

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6701 | 0x4071.01 | - | 1 | U16 | - | RW | - | ES |

Functionality assigned to I/O 1.
The codes of the functionalities are listed in Table 27.21. In the Table 16.6 you can find the resources which can be given to the single functionalities.

## IO_2_Function

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6702 | 0x4072.01 | - | 1 | U16 | - | RW | - | ES |

Functionality assigned to I/O 2.
The codes of the functionalities are listed in Table 27.21. In the Table 16.6 you can find the resources which can be given to the single functionalities.

## IO_3_Function

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6703 | 0x4073.01 | - | 1 | U16 | - | RW | - | ES |

Functionality assigned to I/O 3.
The codes of the functionalities are listed in Table 27.21. In the Table 16.6 you can find the resources which can be given to the single functionalities.

## In_4_Function

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6724 | 0x4084.01 | - | 1 | U16 | - | RW | - | ES |

Functionality assigned to In 4.

The codes of the functionalities are listed in Table 27.21. In the Table 16.6 you can find the resources which can be given to the single functionalities.

### In_5_Function

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6725 | 0x4085.01 | - | 1 | U16 | - | RW | - | ES |

Functionality assigned to In 5.
The codes of the functionalities are listed in Table 27.21. In the Table 16.6 you can find the resources which can be given to the single functionalities.

### In_6_Function

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6726 | 0x4086.01 | - | 1 | U16 | - | RW | - | ES |

Functionality assigned to In 6.
The codes of the functionalities are listed in Table 27.21. In the Table 16.6 you can find the resources which can be given to the single functionalities.

### In_7_Function

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6727 | 0x4087.01 | - | 1 | U16 | - | RW | - | ES |

Functionality assigned to In 7.
The codes of the functionalities are listed in Table 27.21. In the Table 16.6 you can find the resources which can be given to the single functionalities.

### In_8_Function

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6728 | 0x4088.01 | - | 1 | U16 | - | RW | - | ES |

Functionality assigned to In 8.
The codes of the functionalities are listed in Table 27.21. In the Table 16.6 you can find the resources which can be given to the single functionalities.

### In_9_Function

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6729 | 0x4089.01 | - | 1 | U16 | - | RW | - | ES |

Functionality assigned to In 9.
The codes of the functionalities are listed in Table 27.21. In the Table 16.6 you can find the resources which can be given to the single functionalities.

### Out_4_Function

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6789 | 0x40C4.01 | - | 2 | U16 | - | RW | - | ES |

Functionality assigned to Out 4.
The codes of the functionalities are listed in Table 27.21. In the Table 16.6 you can find the resources which can be given to the single functionalities.

### *Out_5_Function*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6790 | 0x40C5.01 | - | 2 | U16 | - | RW | - | ES |

Functionality assigned to Out 5.
The codes of the functionalities are listed in Table 27.21. In the Table 16.6 you can find the resources which can be given to the single functionalities.

### *Out_6_Function*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 6791 | 0x40C6.01 | - | 2 | U16 | - | RW | - | ES |

Functionality assigned to Out 6.
The codes of the functionalities are listed in Table 27.21. In the Table 16.6 you can find the resources which can be given to the single functionalities.

## 27.22. Analog input (7000-7999)

### AI0CalibrationParameters

Calibration of the analog input 0.

#### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x4100.00 | - | 4 | U8 | - | CST | - | - |

Number of parameters in this group. For further details see Section 17.2, *Calibration*.

#### *AI0CalibrationStatus*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 7001 | 0x4100.01 | - | 4 | U16 | - | RO | - | ES |

Status of the calibration of the analog input 0.

| AI0CalibrationStatus | Description |
|---|---|
| 0 | No calibration run; the voltage values may not respect the specified tolerance |
| 1 | Calibration not completed (only offset); complete the calibration |
| 2 | Calibration not completed (only gain); complete the calibration |
| 3 | Calibration completed |
| 4 | Default calibration |

### *AI0CalibrationOffset*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 7002 | 0x4100.02 | - | Desc | S16 | - | RO | - | ES |

Calibration offset of the analog input 0. The default value is calculated with the calibration run by Bonfiglioli.

### *AI0CalibrationGain*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 7003 | 0x4100.03 | - | Desc | U16 | - | RO | - | ES |

Calibration gain of the analog input 0.

### *AI0CalibrationVoltage*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 7004 | 0x4100.04 | 4000-10000 | - | U16 | mV | RW | - | - |

Calibration voltage of the analog input 0. The default value is calculated with the calibration run by Bonfiglioli.

## AI0FilterParameters

Filter of the analog input 0.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x4110.00 | - | 3 | U8 | - | CST | - | - |

Number of parameters in this group. For further details see Chapter 28, *Digital filters*.

### *AI0FilterFrequency*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 7011 | 0x4110.01 | - | 100 | U16 | Hz | RW | - | ES |

Typical frequency of the filter on the analog input 0.

### *AI0FilterType*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 7012 | 0x4110.02 | - | 2 | U16 | - | RW | - | ES |

Filter type on the analog input 0.

| AI0FilterType | Description |
|---|---|
| 0 | All-stop filter |
| 1 | Low-pass filter of the first order |

| AI0FilterType | Description |
|---|---|
| 2 | Low-pass filter of the second order |
| 3 | Band-eliminating filter |
| 65535 | All-pass filter |

### AI0FilterQFactor

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 7013 | 0x4110.03 | - | 100 | U16 | 10 = 1 | RW | - | ES |

Quality Q factor of the filter on the analog input 0.

## AI0ConversionParameters

Settings for the conversions with the analog input 0.

### Number of entries

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x4120.00 | - | 8 | U8 | - | CST | - | - |

Number of parameters in this group. For further details see Section 17.3, *Conversion*.

### AI0VSettings

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 7051 | 0x4120.01 | - | 1 | U16 | - | RW | - | ES |

Setting the voltage for the conversion with the analog input 0 (0 = only positive; 1 = symmetric).

### AI0RSettings

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 7052 | 0x4120.02 | - | 1 | U16 | - | RW | - | ES |

Setting the reference for the conversion with the analog input 0 (0 = only positive; 1 = symmetric).

### AI0VPolarity

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 7053 | 0x4120.03 | - | 0 | U16 | - | RW | - | ES |

Voltage polarity for the conversion with the analog input 0 (0 = normal; 1 = reversed).

### AI0RPolarity

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 7054 | 0x4120.04 | - | 0 | U16 | - | RW | - | ES |

Reference polarity for the conversion with the analog input 0 (0 = normal; 1 = reversed).

### *AI0V0Zone*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 7055 | 0x4120.05 | - | 10 | U16 | mV | RW | - | ES |

Half amplitude of the dead zone of the analog input.

### *AI0VRefLevel*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 7056 | 0x4120.06 | - | 10000 | U16 | mV | RW | - | ES |

Voltage value to define the conversion.

### *AI0TRefValue*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 7057 | 0x4120.07 | 0 ÷ 32767 | 1200 | U16 | 10 = 1%IS | RW | - | ES |

Torque value to define the conversion.

### *AI0WRefValue*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 7058 | 0x4120.08 | - | MotorRatedSpeed | U32 | inc/s | RW | - | ES |

Speed value to define the conversion.

# 27.23. PDO managed by the port CANopen (10000-11999)

NOTE

**This vocabulary section can be found only in the drive version CAN**

## PdoRx1_CommunicationParameters

Communication parameters of the PDO RX 1.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x1400.00 | 2 | 2 | U8 | - | CST | - | - |

Number of parameters in this group.

### *PdoRx1_CobID*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 10000 | 0x1400.01 | Desc | Desc | U32 | - | RW | - | - |

PDO RX 1 COB-ID. In Table 27.22 the bit code interpretation is reported. The default value is 0x80000200 + Node-ID.

| Bit | Description |
|---|---|
| 0 - 10 | Field at 11 bits for the standard ID (CAN 2.0). |
| 11 - 28 | Field at 18 bits for the extended ID, not supported. They must be set at 0. |
| 29 | Format identifier. It must be equal to 0 (standard identifier). |
| 30 | It must always be set at 0 because the drives of the series *i*BMD do not support the PDOs of RTR type |
| 31 | If set to 0, it enables the PDO when the NMT state machine is in the OPERATIONAL status. |

***Table 27.22. Bit encoding of the parameters to set the COB-ID of the PDO***

### *PdoRx1_TransmissionType*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 10002 | 0x1400.02 | Desc | 255 | U8 | - | RW | - | - |

Type of transmission PDO RX 1. In Table 27.23 you can find the accepted values.

| PDO RX Transmission Type | Description |
|---|---|
| 0 - 240 | *Synchronous*. The received PDO is applied only to the receiving of the next SYNC. |
| 241 - 253 | Not supported |
| 254 - 255 | *On event*. The received PDO is immediately applied. |

***Table 27.23. Accepted values of the parameters to set the Transmission Type of the PDOs RX***

## PdoRx2_CommunicationParameters

Communication parameters of the PDO RX 2.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x1401.00 | 2 | 2 | U8 | - | CST | - | - |

Number of parameters in this group.

### *PdoRx2_CobID*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 10015 | 0x1401.01 | Desc | Desc | U32 | - | RW | - | - |

PDO RX 2 COB-ID. In Table 27.22 the bit code interpretation is reported. The default value is 0x80000300 + Node-ID.

### PdoRx2_TransmissionType

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 10017 | 0x1401.02 | Desc | 255 | U8 | - | RW | - | - |

Transmission type PDO RX 2. In Table 27.23 you can find the accepted values.

## PdoRx3_CommunicationParameters

Communication parameters of the PDO RX 3.

### Number of entries

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x1402.00 | 2 | 2 | U8 | - | CST | - | - |

Number of parameters in this group.

### PdoRx3_CobID

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 10030 | 0x1402.01 | Desc | Desc | U32 | - | RW | - | - |

PDO RX 3 COB-ID. In Table 27.22 the bit code interpretation is reported. The default value is 0x80000400 + Node-ID.

### PdoRx3_TransmissionType

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 10032 | 0x1402.02 | Desc | 255 | U8 | - | RW | - | - |

Transmission type PDO RX 3. In Table 27.23 you can find the accepted values.

## PdoRx4_CommunicationParameters

Communication parameters of the PDO RX 4.

### Number of entries

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x1403.00 | 2 | 2 | U8 | - | CST | - | - |

Number of parameters in this group.

### PdoRx4_CobID

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 10045 | 0x1403.01 | Desc | Desc | U32 | - | RW | - | - |

PDO RX 4 COB-ID. In Table 27.22 the bit code interpretation is reported. The default value is 0x80000500 + Node-ID.

*PdoRx4_TransmissionType*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 10047 | 0x1403.02 | Desc | 255 | U8 | - | RW | - | - |

Transmission type PDO RX 4. In Table 27.23 you can find the accepted values.

### PdoRx1_MappingParameters

Parameters for the PDO RX 1 mapping.

*Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 10150 | 0x1600.00 | 0 - 8 | 1 | U8 | - | RW | - | - |

Number of mapped objects.

*PdoRx1_Objects*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| Desc | Desc | Desc | Desc | U32 | - | RW | - | - |

Parameters used for mapping the mappable parameters in the PDOs.

| Modbus | CANopen | Name | Default |
|---|---|---|---|
| 10151 | 0x1600.01 | PdoRx1_Object1 | 0x60400010 (Controlword) |
| 10153 | 0x1600.02 | PdoRx1_Object2 | 0 |
| 10155 | 0x1600.03 | PdoRx1_Object3 | 0 |
| 10157 | 0x1600.04 | PdoRx1_Object4 | 0 |
| 10159 | 0x1600.05 | PdoRx1_Object5 | 0 |
| 10161 | 0x1600.06 | PdoRx1_Object6 | 0 |
| 10163 | 0x1600.07 | PdoRx1_Object7 | 0 |
| 10165 | 0x1600.08 | PdoRx1_Object8 | 0 |

### PdoRx2_MappingParameters

Parameters for the PDO RX 2 mapping.

*Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 10175 | 0x1601.00 | 0 - 8 | 2 | U8 | - | RW | - | - |

Number of mapped objects.

*PdoRx2_Objects*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| Desc | Desc | Desc | Desc | U32 | - | RW | - | - |

Parameters used for mapping the mappable parameters in the PDOs.

| Modbus | CANopen | Name | Default |
|---|---|---|---|
| 10176 | 0x1601.01 | PdoRx2_Object1 | 0x60400010 (Controlword) |
| 10178 | 0x1601.02 | PdoRx2_Object2 | 0x60600008 (ModesOfOperation) |
| 10180 | 0x1601.03 | PdoRx2_Object3 | 0 |
| 10182 | 0x1601.04 | PdoRx2_Object4 | 0 |
| 10184 | 0x1601.05 | PdoRx2_Object5 | 0 |
| 10186 | 0x1601.06 | PdoRx2_Object6 | 0 |
| 10188 | 0x1601.07 | PdoRx2_Object7 | 0 |
| 10190 | 0x1601.08 | PdoRx2_Object8 | 0 |

## PdoRx3_MappingParameters

Parameters for the PDO RX 3 mapping.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 10200 | 0x1602.00 | 0 - 8 | 2 | U8 | - | RW | - | - |

Number of mapped objects.

### *PdoRx3_Objects*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| Desc | Desc | Desc | Desc | U32 | - | RW | - | - |

Parameters used for mapping the mappable parameters in the PDOs.

| Modbus | CANopen | Name | Default |
|---|---|---|---|
| 10201 | 0x1602.01 | PdoRx3_Object1 | 0x60400010 (Controlword) |
| 10203 | 0x1602.02 | PdoRx3_Object2 | 0x607A0020 (TargetPosition) |
| 10205 | 0x1602.03 | PdoRx3_Object3 | 0 |
| 10207 | 0x1602.04 | PdoRx3_Object4 | 0 |
| 10209 | 0x1602.05 | PdoRx3_Object5 | 0 |
| 10211 | 0x1602.06 | PdoRx3_Object6 | 0 |
| 10213 | 0x1602.07 | PdoRx3_Object7 | 0 |
| 10215 | 0x1602.08 | PdoRx3_Object8 | 0 |

## PdoRx4_MappingParameters

Parameters for the PDO RX 4 mapping.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 10225 | 0x1603.00 | 0 - 8 | 2 | U8 | - | RW | - | - |

Number of mapped objects.

### *PdoRx4_Objects*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| Desc | Desc | Desc | Desc | U32 | - | RW | - | - |

Parameters used for mapping the mappable parameters in the PDOs.

| Modbus | CANopen | Name | Default |
|---|---|---|---|
| 10226 | 0x1603.01 | PdoRx4_Object1 | 0x60400010 (Controlword) |
| 10228 | 0x1603.02 | PdoRx4_Object2 | 0x60FF0020 (TargetVelocity) |
| 10230 | 0x1603.03 | PdoRx4_Object3 | 0 |
| 10232 | 0x1603.04 | PdoRx4_Object4 | 0 |
| 10234 | 0x1603.05 | PdoRx4_Object5 | 0 |
| 10236 | 0x1603.06 | PdoRx4_Object6 | 0 |
| 10238 | 0x1603.07 | PdoRx4_Object7 | 0 |
| 10240 | 0x1603.08 | PdoRx4_Object8 | 0 |

## PdoTx1_CommunicationParameters

Communication parameters of the PDO TX 1.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x1800.00 | 5 | 5 | U8 | - | CST | - | - |

Number of parameters in this group.

### *PdoTx1_CobID*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 10350 | 0x1800.01 | Desc | Desc | U32 | - | RW | - | - |

PDO TX 1 COB-ID. In Table 27.22 the bit code interpretation is reported. The default value is 0xC0000180 + Node-ID.

### *PdoTx1_TransmissionType*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 10352 | 0x1800.02 | Desc | 255 | U8 | - | RW | - | - |

Transmission type PDO TX 1. In Table 27.24 you can find the accepted values.

| PDO TX Transmission Type | Description |
|---|---|
| 0 | *Acyclic*. The PDO is transmitted together with the SYNC, but only when the internal logics of the drive asks for the sending. This value cannot be set if the Event timer of the PDO is different from 0. |
| 1 - 240 | *Cyclic*. The PDO is transmitted together with the SYNC, one for each *PDO TX Transmission Type* SYNC messages received. This values cannot be set if the Event timer of the PDO is different from 0. |
| 241 - 253 | Not supported |
| 254 - 255 | *On event*. The PDO is immediately transmitted the internal logics of the drive asks for the sending. |

***Table 27.24. Accepted values for setting the Transmission Type of the PDO TX***

### *PdoTx1_InhibitTime*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 10353 | 0x1800.03 | - | 1 | U16 | 10 = 1ms | RW | - | - |

PDO TX 1 inhibition time (see the notes reported in Section 9.5, *Process data object (PDO)*).

### *PdoTx1_EventTimer*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 10356 | 0x1800.05 | - | 0 | U16 | ms | RW | - | - |

Event timer of the PDO TX 1.

## PdoTx2_CommunicationParameters

Communication parameters of the PDO TX 2.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x1801.00 | 5 | 5 | U8 | - | CST | - | - |

Number of parameters in this group.

### *PdoTx2_CobID*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 10365 | 0x1801.01 | Desc | Desc | U32 | - | RW | - | - |

PDO TX 2 COB-ID. In Table 27.22 the bit code interpretation is reported. The default value is 0xC0000280 + Node-ID.

### *PdoTx2_TransmissionType*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 10367 | 0x1801.02 | Desc | 255 | U8 | - | RW | - | - |

Transmission type PDO TX 2. In Table 27.24 you can find the accepted values.

### *PdoTx2_InhibitTime*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 10368 | 0x1801.03 | - | 1 | U16 | 10 = 1ms | RW | - | - |

PDO TX 2 inhibition time (see the notes reported in Section 9.5, *Process data object (PDO)*).

### *PdoTx2_EventTimer*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 10371 | 0x1801.05 | - | 0 | U16 | ms | RW | - | - |

Event timer of the PDO TX 2.

## PdoTx3_CommunicationParameters

Communication parameters of the PDO TX 3.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x1802.00 | 5 | 5 | U8 | - | CST | - | - |

Number of parameters in this group.

### *PdoTx3_CobID*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 10380 | 0x1802.01 | Desc | Desc | U32 | - | RW | - | - |

PDO TX 3 COB-ID. In Table 27.22 the bit code interpretation is reported. The default value is 0xC0000380 + Node-ID.

### *PdoTx3_TransmissionType*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 10382 | 0x1802.02 | Desc | 255 | U8 | - | RW | - | - |

Transmission type PDO TX 3. In Table 27.24 you can find the accepted values.

### *PdoTx3_InhibitTime*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 10383 | 0x1802.03 | - | 1 | U16 | 10 = 1ms | RW | - | - |

PDO TX 3 inhibition time (see the notes reported in Section 9.5, *Process data object (PDO)*).

### *PdoTx3_EventTimer*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 10386 | 0x1802.05 | - | 0 | U16 | ms | RW | - | - |

Event timer of the PDO TX 3.

### PdoTx4_CommunicationParameters

Communication parameters of the PDO TX 4.

#### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x1803.00 | 5 | 5 | U8 | - | CST | - | - |

Number of parameters in this group.

#### *PdoTx4_CobID*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 10395 | 0x1803.01 | Desc | Desc | U32 | - | RW | - | - |

PDO TX 4 COB-ID. In Table 27.22 the bit code interpretation is reported. The default value is 0xC0000480 + Node-ID.

#### *PdoTx4_TransmissionType*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 10397 | 0x1803.02 | Desc | 255 | U8 | - | RW | - | - |

Transmission type PDO TX 4. In Table 27.24 you can find the accepted values.

#### *PdoTx4_InhibitTime*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 10398 | 0x1803.03 | - | 1 | U16 | 10 = 1ms | RW | - | - |

PDO TX 4 inhibition time (see the notes reported in Section 9.5, *Process data object (PDO)*).

#### *PdoTx4_EventTimer*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 10401 | 0x1803.05 | - | 0 | U16 | ms | RW | - | - |

Event timer of the PDO TX 4.

### PdoTx1_MappingParameters

Parameters for the PDO TX 1 mapping.

#### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 10500 | 0x1A00.00 | 0 - 8 | 1 | U8 | - | RW | - | - |

Number of mapped objects.

### PdoTx1_Objects

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| Desc | Desc | Desc | Desc | U32 | - | RW | - | - |

Parameters used for mapping the mappable parameters in the PDOs.

| Modbus | CANopen | Name | Default |
|---|---|---|---|
| 10501 | 0x1A00.01 | PdoTx1_Object1 | 0x60410010 (Statusword) |
| 10503 | 0x1A00.02 | PdoTx1_Object2 | 0 |
| 10505 | 0x1A00.03 | PdoTx1_Object3 | 0 |
| 10507 | 0x1A00.04 | PdoTx1_Object4 | 0 |
| 10509 | 0x1A00.05 | PdoTx1_Object5 | 0 |
| 10511 | 0x1A00.06 | PdoTx1_Object6 | 0 |
| 10513 | 0x1A00.07 | PdoTx1_Object7 | 0 |
| 10515 | 0x1A00.08 | PdoTx1_Object8 | 0 |

## PdoTx2_MappingParameters

Parameters for the PDO TX 2 mapping.

### Number of entries

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 10525 | 0x1A01.00 | 0 - 8 | 2 | U8 | - | RW | - | - |

Number of mapped objects.

### PdoTx2_Objects

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| Desc | Desc | Desc | Desc | U32 | - | RW | - | - |

Parameters used for mapping the mappable parameters in the PDOs.

| Modbus | CANopen | Name | Default |
|---|---|---|---|
| 10526 | 0x1A01.01 | PdoTx2_Object1 | 0x60410010 (Statusword) |
| 10528 | 0x1A01.02 | PdoTx2_Object2 | 0x60610008 (ModesOfOperationDisplay) |
| 10530 | 0x1A01.03 | PdoTx2_Object3 | 0 |
| 10532 | 0x1A01.04 | PdoTx2_Object4 | 0 |
| 10534 | 0x1A01.05 | PdoTx2_Object5 | 0 |
| 10536 | 0x1A01.06 | PdoTx2_Object6 | 0 |
| 10538 | 0x1A01.07 | PdoTx2_Object7 | 0 |
| 10540 | 0x1A01.08 | PdoTx2_Object8 | 0 |

## PdoTx3_MappingParameters

Parameters for the PDO TX 3 mapping.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 10550 | 0x1A02.00 | 0 - 8 | 2 | U8 | - | RW | - | - |

Number of mapped objects.

### *PdoTx3_Objects*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| Desc | Desc | Desc | Desc | U32 | - | RW | - | - |

Parameters used for mapping the mappable parameters in the PDOs.

| Modbus | CANopen | Name | Default |
|---|---|---|---|
| 10551 | 0x1A02.01 | PdoTx3_Object1 | 0x60410010 (Statusword) |
| 10553 | 0x1A02.02 | PdoTx3_Object2 | 0x60640020 (PositionActualValue) |
| 10555 | 0x1A02.03 | PdoTx3_Object3 | 0 |
| 10557 | 0x1A02.04 | PdoTx3_Object4 | 0 |
| 10559 | 0x1A02.05 | PdoTx3_Object5 | 0 |
| 10561 | 0x1A02.06 | PdoTx3_Object6 | 0 |
| 10563 | 0x1A02.07 | PdoTx3_Object7 | 0 |
| 10565 | 0x1A02.08 | PdoTx3_Object8 | 0 |

## PdoTx4_MappingParameters

Parameters for the PDO TX 4 mapping.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 10575 | 0x1A03.00 | 0 - 8 | 2 | U8 | - | RW | - | - |

Number of mapped objects.

### *PdoTx4_Objects*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| Desc | Desc | Desc | Desc | U32 | - | RW | - | - |

Parameters used for mapping the mappable parameters in the PDOs.

| Modbus | CANopen | Name | Default |
|---|---|---|---|
| 10576 | 0x1A03.01 | PdoTx4_Object1 | 0x60410010 (Statusword) |
| 10578 | 0x1A03.02 | PdoTx4_Object2 | 0x606C0020 (VelocityActualValue) |

| Modbus | CANopen | Name | Default |
|---|---|---|---|
| 10580 | 0x1A03.03 | PdoTx4_Object3 | 0 |
| 10582 | 0x1A03.04 | PdoTx4_Object4 | 0 |
| 10584 | 0x1A03.05 | PdoTx4_Object5 | 0 |
| 10586 | 0x1A03.06 | PdoTx4_Object6 | 0 |
| 10588 | 0x1A03.07 | PdoTx4_Object7 | 0 |
| 10590 | 0x1A03.08 | PdoTx4_Object8 | 0 |

# 27.24. Sync manager and PDOs managed by the port EtherCAT

NOTE

**This vocabulary section can be found only in the drive versions ETC**

### PdoRx1_MappingParameters

Parameters for the PDO RX 1 mapping.

*Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x1600.00 | 0 - 8 | 2 | U8 | - | RW | - | - |

Number of mapped objects.

*PdoRx1_Objects*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| Desc | Desc | Desc | Desc | U32 | - | RW | - | - |

Parameters used for mapping the mappable parameters in the PDOs.

| Modbus | CANopen | Name | Default |
|---|---|---|---|
| - | 0x1600.01 | PdoRx1_Object1 | 0x60400010 (Controlword) |
| - | 0x1600.02 | PdoRx1_Object2 | 0x60C10120 (IpPosFirstParameter) |
| - | 0x1600.03 | PdoRx1_Object3 | 0 |
| - | 0x1600.04 | PdoRx1_Object4 | 0 |
| - | 0x1600.05 | PdoRx1_Object5 | 0 |
| - | 0x1600.06 | PdoRx1_Object6 | 0 |
| - | 0x1600.07 | PdoRx1_Object7 | 0 |
| - | 0x1600.08 | PdoRx1_Object8 | 0 |

### PdoRx2_MappingParameters

Parameters for the PDO RX 2 mapping.

### Number of entries

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x1601.00 | 0 - 8 | 6 | U8 | - | RW | - | - |

Number of mapped objects.

### PdoRx2_Objects

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| Desc | Desc | Desc | Desc | U32 | - | RW | - | - |

Parameters used for mapping the mappable parameters in the PDOs.

| Modbus | CANopen | Name | Default |
|---|---|---|---|
| - | 0x1601.01 | PdoRx2_Object1 | 0x60400010 (Controlword) |
| - | 0x1601.02 | PdoRx2_Object2 | 0x607A0020 (TargetPosition) |
| - | 0x1601.03 | PdoRx2_Object3 | 0x60FF0020 (TargetVelocity) |
| - | 0x1601.04 | PdoRx2_Object4 | 0x60710010 (TargetTorque) |
| - | 0x1601.05 | PdoRx2_Object5 | 0x60720010 (MaxTorque) |
| - | 0x1601.06 | PdoRx2_Object6 | 0x60600008 (ModesOfOperation) |
| - | 0x1601.07 | PdoRx2_Object7 | 0 |
| - | 0x1601.08 | PdoRx2_Object8 | 0 |

## PdoRx3_MappingParameters

Parameters for the PDO RX 3 mapping.

### Number of entries

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x1602.00 | 0 - 8 | 2 | U8 | - | RW | - | - |

Number of mapped objects.

### PdoRx3_Objects

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| Desc | Desc | Desc | Desc | U32 | - | RW | - | - |

Parameters used for mapping the mappable parameters in the PDOs.

| Modbus | CANopen | Name | Default |
|---|---|---|---|
| - | 0x1602.01 | PdoRx3_Object1 | 0x60400010 (Controlword) |

| Modbus | CANopen | Name | Default |
|---|---|---|---|
| - | 0x1602.02 | PdoRx3_Object2 | 0x607A0020 (TargetPosition) |
| - | 0x1602.03 | PdoRx3_Object3 | 0 |
| - | 0x1602.04 | PdoRx3_Object4 | 0 |
| - | 0x1602.05 | PdoRx3_Object5 | 0 |
| - | 0x1602.06 | PdoRx3_Object6 | 0 |
| - | 0x1602.07 | PdoRx3_Object7 | 0 |
| - | 0x1602.08 | PdoRx3_Object8 | 0 |

## PdoRx4_MappingParameters

Parameters for the PDO RX 4 mapping.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x1603.00 | 0 - 8 | 2 | U8 | - | RW | - | - |

Number of mapped objects.

### *PdoRx4_Objects*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| Desc | Desc | Desc | Desc | U32 | - | RW | - | - |

Parameters used for mapping the mappable parameters in the PDOs.

| Modbus | CANopen | Name | Default |
|---|---|---|---|
| - | 0x1603.01 | PdoRx4_Object1 | 0x60400010 (Controlword) |
| - | 0x1603.02 | PdoRx4_Object2 | 0x60FF0020 (TargetVelocity) |
| - | 0x1603.03 | PdoRx4_Object3 | 0 |
| - | 0x1603.04 | PdoRx4_Object4 | 0 |
| - | 0x1603.05 | PdoRx4_Object5 | 0 |
| - | 0x1603.06 | PdoRx4_Object6 | 0 |
| - | 0x1603.07 | PdoRx4_Object7 | 0 |
| - | 0x1603.08 | PdoRx4_Object8 | 0 |

## PdoTx1_MappingParameters

Parameters for the PDO TX 1 mapping.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x1A00.00 | 0 - 8 | 2 | U8 | - | RW | - | - |

Number of mapped objects.

### *PdoTx1_Objects*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| Desc | Desc | Desc | Desc | U32 | - | RW | - | - |

Parameters used for mapping the mappable parameters in the PDOs.

| Modbus | CANopen | Name | Default |
|---|---|---|---|
| - | 0x1A00.01 | PdoTx1_Object1 | 0x60410010 (Statusword) |
| - | 0x1A00.02 | PdoTx1_Object2 | 0x60640020 (PositionActualValue) |
| - | 0x1A00.03 | PdoTx1_Object3 | 0 |
| - | 0x1A00.04 | PdoTx1_Object4 | 0 |
| - | 0x1A00.05 | PdoTx1_Object5 | 0 |
| - | 0x1A00.06 | PdoTx1_Object6 | 0 |
| - | 0x1A00.07 | PdoTx1_Object7 | 0 |
| - | 0x1A00.08 | PdoTx1_Object8 | 0 |

## PdoTx2_MappingParameters

Parameters for the PDO TX 2 mapping.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x1A01.00 | 0 - 8 | 6 | U8 | - | RW | - | - |

Number of mapped objects.

### *PdoTx2_Objects*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| Desc | Desc | Desc | Desc | U32 | - | RW | - | - |

Parameters used for mapping the mappable parameters in the PDOs.

| Modbus | CANopen | Name | Default |
|---|---|---|---|
| - | 0x1A01.01 | PdoTx2_Object1 | 0x60410010 (Statusword) |
| - | 0x1A01.02 | PdoTx2_Object2 | 0x60640020 (PositionActualValue) |
| - | 0x1A01.03 | PdoTx2_Object3 | 0x60F40020 (PositionFollowingError) |
| - | 0x1A01.04 | PdoTx2_Object4 | 0x60770010 (ActualTorque) |
| - | 0x1A01.05 | PdoTx2_Object5 | 0x60610008 (ModesOfOperationDisplay) |
| - | 0x1A01.06 | PdoTx2_Object6 | 0x606C0020 |
| - | 0x1A01.07 | PdoTx2_Object7 | 0 |

| Modbus | CANopen | Name | Default |
|---|---|---|---|
| - | 0x1A01.08 | PdoTx2_Object8 | 0 |

## PdoTx3_MappingParameters

Parameters for the PDO TX 3 mapping.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x1A02.00 | 0 - 8 | 2 | U8 | - | RW | - | - |

Number of mapped objects.

### *PdoTx3_Objects*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| Desc | Desc | Desc | Desc | U32 | - | RW | - | - |

Parameters used for mapping the mappable parameters in the PDOs.

| Modbus | CANopen | Name | Default |
|---|---|---|---|
| - | 0x1A02.01 | PdoTx3_Object1 | 0x60410010 (Statusword) |
| - | 0x1A02.02 | PdoTx3_Object2 | 0x60640020 (PositionActualValue) |
| - | 0x1A02.03 | PdoTx3_Object3 | 0 |
| - | 0x1A02.04 | PdoTx3_Object4 | 0 |
| - | 0x1A02.05 | PdoTx3_Object5 | 0 |
| - | 0x1A02.06 | PdoTx3_Object6 | 0 |
| - | 0x1A02.07 | PdoTx3_Object7 | 0 |
| - | 0x1A02.08 | PdoTx3_Object8 | 0 |

## PdoTx4_MappingParameters

Parameters for the PDO TX 4 mapping.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x1A03.00 | 0 - 8 | 3 | U8 | - | RW | - | - |

Number of mapped objects.

### *PdoTx4_Objects*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| Desc | Desc | Desc | Desc | U32 | - | RW | - | - |

Parameters used for mapping the mappable parameters in the PDOs.

| Modbus | CANopen | Name | Default |
|---|---|---|---|
| - | 0x1A03.01 | PdoTx4_Object1 | 0x60410010 (Statusword) |
| - | 0x1A03.02 | PdoTx4_Object2 | 0x60640020 (PositionActualValue) |
| - | 0x1A03.03 | PdoTx4_Object3 | 0x606C0020 (VelocityActualValue) |
| - | 0x1A03.04 | PdoTx4_Object4 | 0 |
| - | 0x1A03.05 | PdoTx4_Object5 | 0 |
| - | 0x1A03.06 | PdoTx4_Object6 | 0 |
| - | 0x1A03.07 | PdoTx4_Object7 | 0 |
| - | 0x1A03.08 | PdoTx4_Object8 | 0 |

## SM_CommunicationType

Communication type set in the Sync Manager.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x1C00.00 | 4 | 4 | U8 | - | CST | - | - |

Number of parameters in this group.

### *SM0_CommunicationType*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x1C00.01 | 1 | 1 | U8 | - | RO | - | ES |

Communication type of the Sync manager 0. In Table 27.25 you can find the accepted values.

| Value | Communication type of the Sync Manager |
|---|---|
| 1 | Mailbox RX |
| 2 | Mailbox TX |
| 3 | Process data RX |
| 4 | Process data TX |

***Table 27.25. Sync Manager Communication Type***

### *SM1_CommunicationType*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x1C00.02 | 2 | 2 | U8 | - | RO | - | ES |

Communication type of the Sync manager 1. In Table 27.25 you can find the accepted values.

### *SM2_CommunicationType*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x1C00.03 | 3 | 3 | U8 | - | RO | - | ES |

Communication type of the Sync manager 2. In Table 27.25 you can find the accepted values.

### *SM3_CommunicationType*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x1C00.04 | 4 | 4 | U8 | - | RO | - | ES |

Communication type of the Sync manager 3. In Table 27.25 you can find the accepted values.

## SM0_PdoAssignment

Parameters for the assignment of the PDOs to the Sync Manager 0.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x1C10.00 | 0 | 0 | U8 | - | RO | - | - |

Number of PDOs given to the Sync Manager 0. For this Sync Manager it is not possible to assign any PDO.

### *SM0_PdoMapping*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| Desc | Desc | Desc | Desc | U16 | - | RO | - | - |

Parameters for the assignment of the PDOs to the Sync Manager 0. For this Sync Manager it is not possible to assign any PDO.

| Modbus | CANopen | Name | Default |
|---|---|---|---|
| - | 0x1C10.01 | SM0_PdoMapping0 | 0 |
| - | 0x1C10.02 | SM0_PdoMapping1 | 0 |
| - | 0x1C10.03 | SM0_PdoMapping2 | 0 |
| - | 0x1C10.04 | SM0_PdoMapping3 | 0 |

## SM1_PdoAssignment

Parameters for the assignment of the PDOs to the Sync Manager 1.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x1C11.00 | 0 | 0 | U8 | - | RO | - | - |

Number of PDOs given to the Sync Manager 1. For this Sync Manager it is not possible to assign any PDO.

### *SM1_PdoMapping*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| Desc | Desc | Desc | Desc | U16 | - | RO | - | - |

Parameters for the assignment of the PDOs to the Sync Manager 1. For this Sync Manager it is not possible to assign any PDO.

| Modbus | CANopen | Name | Default |
|---|---|---|---|
| - | 0x1C11.01 | SM1_PdoMapping0 | 0 |
| - | 0x1C11.02 | SM1_PdoMapping1 | 0 |
| - | 0x1C11.03 | SM1_PdoMapping2 | 0 |
| - | 0x1C11.04 | SM1_PdoMapping3 | 0 |

## SM2_PdoAssignment

Parameters for the assignment of the PDOs to the Sync Manager 2.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x1C12.00 | 0-4 | 1 | U8 | - | RW | - | - |

Number of PDOs given to the Sync Manager 2.

### *SM2_PdoMapping*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| Desc | Desc | Desc | Desc | U16 | - | RW | - | - |

Parameters for the assignment of the PDOs to the Sync Manager 2, for this Sync Manager it is possible to assign only PDO RX. The assignment takes place by writing the index CANopen (without subindex) of the *PDO RX Mapping parameters* that you want to assign.

| Modbus | CANopen | Name | Default |
|---|---|---|---|
| - | 0x1C12.01 | SM2_PdoMapping0 | 0x1600 (PdoRx1_MappingParameters) |
| - | 0x1C12.02 | SM2_PdoMapping1 | 0 |
| - | 0x1C12.03 | SM2_PdoMapping2 | 0 |
| - | 0x1C12.04 | SM2_PdoMapping3 | 0 |

## SM3_PdoAssignment

Parameters for the assignment of the PDOs to the Sync Manager 3.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x1C13.00 | 0-4 | 1 | U8 | - | RW | - | - |

Number of PDOs assigned to the Sync Manager 3.

### *SM3_PdoMapping*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| Desc | Desc | Desc | Desc | U16 | - | RW | - | - |

Parameters for the assignment of the PDOs to the Sync Manager 3, for this Sync Manager it is possible to assign only PDO TX. The assignment takes place by writing the index CANopen (without subindex) of the *PDO TX Mapping parameters* that you want to assign.

| Modbus | CANopen | Name | Default |
|---|---|---|---|
| - | 0x1C13.01 | SM3_PdoMapping0 | 0x1A00 (PdoTx1_MappingParameters) |
| - | 0x1C13.02 | SM3_PdoMapping1 | 0 |
| - | 0x1C13.03 | SM3_PdoMapping2 | 0 |
| - | 0x1C13.04 | SM3_PdoMapping3 | 0 |

## SM0_Synchronization

Parameters used for the management of the messages synchronization of the sync 0.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x1C30.00 | 3 | 3 | U8 | - | CST | - | - |

Number of parameters in this group.

### *SM0_SynchronizationType*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x1C30.01 | 0 | 0 | U16 | - | RO | - | - |

Synchronization type of the messages managed by the Sync Manager 0.

### *SM0_CycleTime*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x1C30.02 | 0 | 0 | U32 | ns | RO | - | - |

Time period of the messages managed by the Sync Manager 0.

### *SM0_ShiftTime*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x1C30.03 | 0 | 0 | U32 | ns | RO | - | - |

Time interval between the receiving/sending of the messages managed by the Sync Manager 0 and their application.

## SM1_Synchronization

Parameters used for the management of the messages synchronization of the sync 1.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x1C31.00 | 3 | 3 | U8 | - | CST | - | - |

Number of parameters in this group.

### *SM1_SynchronizationType*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x1C31.01 | 0 | 0 | U16 | - | RO | - | - |

Synchronization type of the messages managed by the Sync Manager 1.

### *SM1_CycleTime*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x1C31.02 | 0 | 0 | U32 | ns | RO | - | - |

Time period of the messages managed by the Sync Manager 1.

### *SM1_ShiftTime*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x1C31.03 | 0 | 0 | U32 | ns | RO | - | - |

Time interval between the receiving/sending of the messages managed by the Sync Manager 1 and their application.

## SM2_Synchronization

Parameters used for the management of the messages synchronization of the sync 2.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x1C32.00 | 3 | 3 | U8 | - | CST | - | - |

Number of parameters in this group.

### *SM2_SynchronizationType*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 11200 | 0x1C32.01 | 0 - 2 | 1 | U16 | - | RW | - | - |

Synchronization type of the messages managed by the Sync Manager 2. In Table 27.26 you can find the meaning of the values.

| Value | Synchronization type of the Sync Manager |
|---|---|
| 0 | *Free run*. No synchronization. |
| 1 | *Synchronized on this Sync Manager*. Synchronization of the event started by the Sync Manager on which this setting is applied. |
| 2 | *Synchronized on Sync Signal 0*. Synchronization on the signal Sync 0 managed by the Distributed clocks. |
| 34 | *Synchronized on SM 2*. Synchronization of the event started by the Sync Manager 2. |

***Table 27.26. Sync Manager Synchronization Type***

### *SM2_CycleTime*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 11201 | 0x1C32.02 | 0 | 0 | U32 | ns | RW | - | - |

Time period of the messages managed by the Sync Manager 2. Only times that are multiples of 1 µs are admitted.

### *SM2_ShiftTime*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 11203 | 0x1C32.03 | 0 | 0 | U32 | ns | RO | - | - |

Time interval between the receiving/sending of the messages managed by the Sync Manager 2 and their application.

## SM3_Synchronization

Parameters used for the management of the messages synchronization of the sync 3.

### *Number of entries*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| - | 0x1C33.00 | 3 | 3 | U8 | - | CST | - | - |

Number of parameters in this group.

### *SM3_SynchronizationType*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 11300 | 0x1C33.01 | 0; 2; 34 | 34 | U16 | - | RW | - | - |

Synchronization type of the messages managed by the Sync Manager 3. In Table 27.26 you can find the meaning of the values.

### *SM3_CycleTime*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 11301 | 0x1C33.02 | 0 | 0 | U32 | ns | RW | - | - |

Time period of the messages managed by the Sync Manager 3. Only times that are multiples of 1 µs are admitted.

### *SM3_ShiftTime*

| Modbus | CANopen | Range | Default | Type | Units | Acc | PDO | Mem |
|---|---|---|---|---|---|---|---|---|
| 11303 | 0x1C33.03 | 0 | 0 | U32 | ns | RO | - | - |

Time interval between the receiving/sending of the messages managed by the Sync Manager 3 and their application.

# Chapter 28
# Digital filters

The drives of the *i*BMD series provide a library of programmable digital filters. The available filters are:

- All-pass filter
- Low-pass filter of the first order
- Low-pass filter of the second order
- Band-eliminating filter
- All-stop filter.

The parameters of every filter can be modified in any moment, even during the working of the filter. The output of the filter adjusts in real time to the new settings.

ADVICE

**Do not set the frequencies of the filters at 0 Hz or at values which are one third higher than the frequency of the filter sampling.**

**All-pass filter**
The all-pass filter or transparent filter does not apply any filtering action. The input signal of the filter is brought back to the output with unity gain and without phase delay.

**Low-pass filter of the first order**

The low-pass filter of the first order is implemented in the *i*BMD drive series as a Butterworth low-pass digital filter of the first order with pre-warping compensation. Here follow the Bode diagrams of the function of filter transfer as the frequency changes, normalized at the value of the critical frequency. You can notice that for frequencies higher than the critical frequency, the module slope is -20dB/decade and the maximum phase delay is 90°.

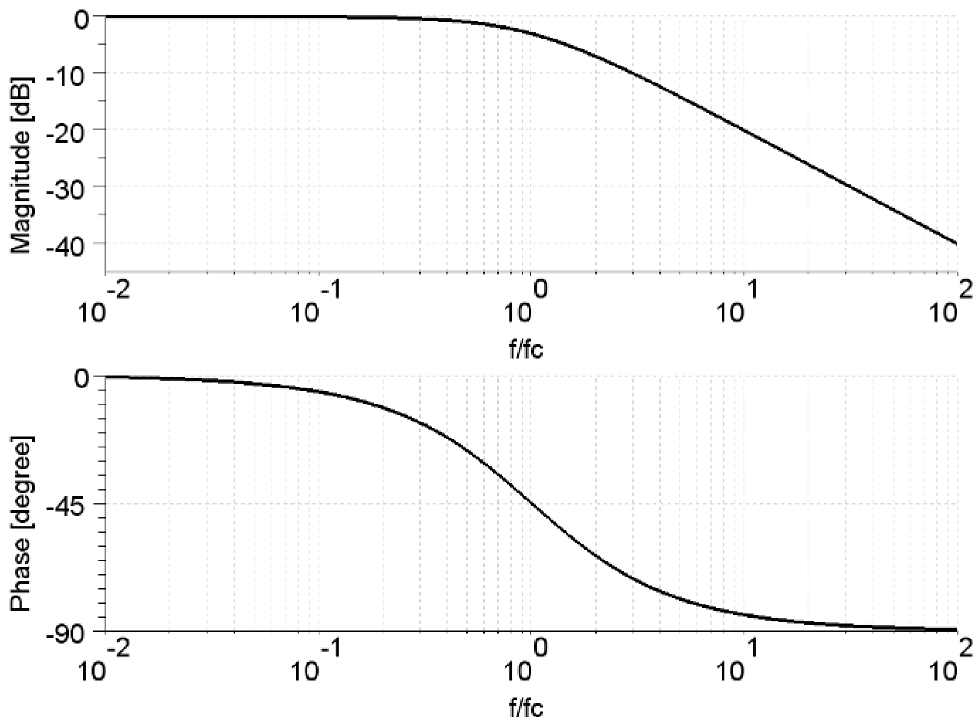



***Figure 28.1. Bode diagrams of the transfer function of the low-pass filter of the first Butterworth order.***

| NOTE |
| --- |
| **Setting the frequency at 0 means having a filter with an infinite attenuation band. This condition cannot be accepted because the filter loses its low-pass property. It is not recommended to use the filter under these conditions.** |

**Low-pass filter of the second order**

The low-pass filter of the second order is implemented in the *i*BMD drive series as a Butterworth low-pass digital filter of the second order with pre-warping compensation. Here follow the Bode diagrams of the function of filter transfer as the frequency changes, normalized at the value of the critical frequency. You can notice that for frequencies higher than the critical frequency, the module slope is -40dB/decade and the maximum phase delay is 180°.

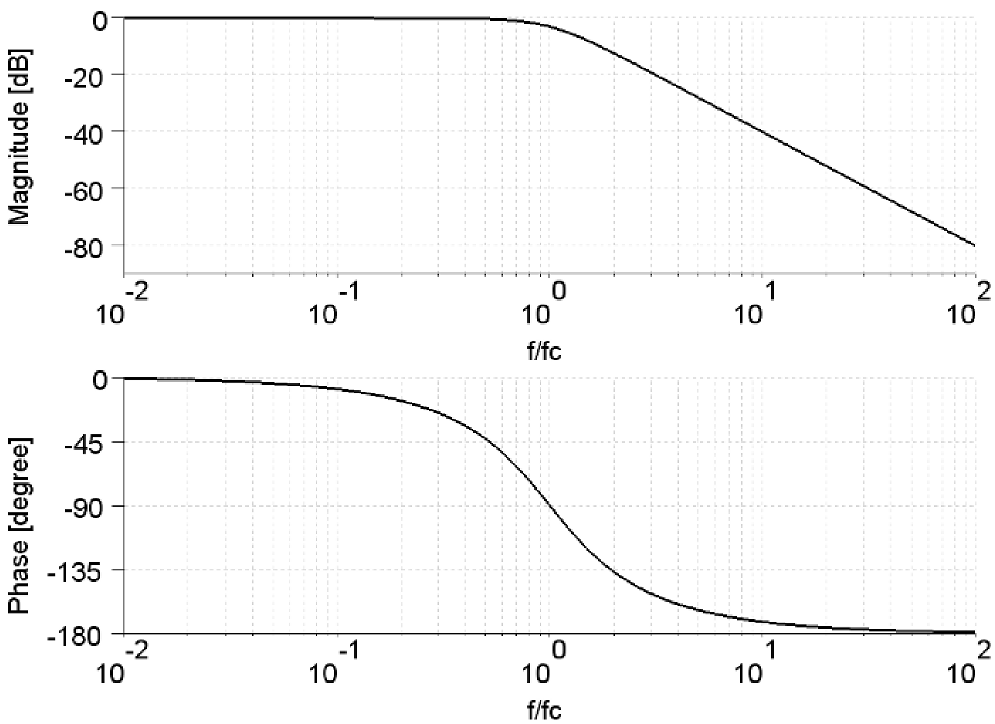



*Figure 28.2. Bode diagrams of the transfer function of the low-pass filter of the second Butterworth order.*

NOTE

**Setting the frequency at 0 means having a filter with an infinite attenuation band. This condition cannot be accepted because the filter loses its low-pass property. It is not recommended to use the filter under these conditions.**

### All-stop filter

The all-stop filter stops the passing of every signal by resetting at zero its output.

### Band-eliminating filter

The band-eliminating filter is implemented in the *i*BMD drive series as a Notch filter of the second order with pre-warping compensation. Here follow the Bode diagrams of the function of filter transfer as the frequency changes, normalized at the value of the central frequency, for different values of the Q quality factor.

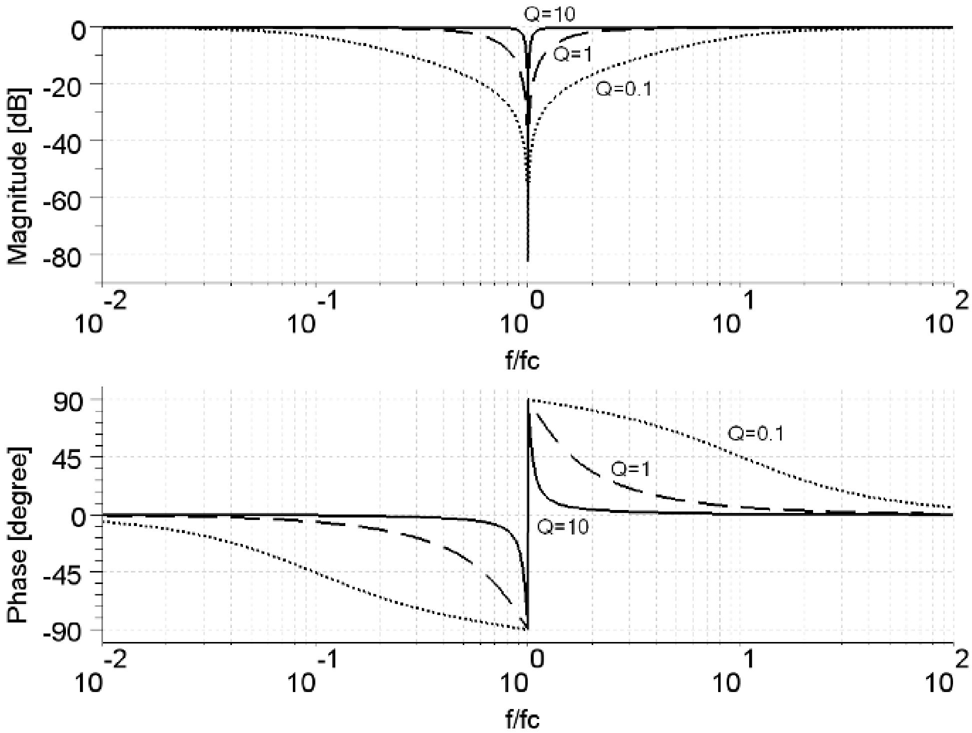



*Figure 28.3. Bode diagrams of the transfer function of the Notch filter of the second order.*

NOTE

**Setting a Q quality factor at 0 means having a filter with an infinite attenuation band. This condition cannot be accepted because the filter loses its band-eliminating property. Furthermore when the Q quality factor increases and the central frequency decreases, the filter convergence time increases. It is not recommended to use the filter under these conditions.**

# Chapter 29
# Order codes

## 29.1. OrderCode

The *i*BMD series drives are available in different models with various functionalities. The ManufacturerDeviceName, that's composed by the order coded preceded by the product name (e.g. *i*BMD), defines the features and the functionalities, and identifies the drive model.

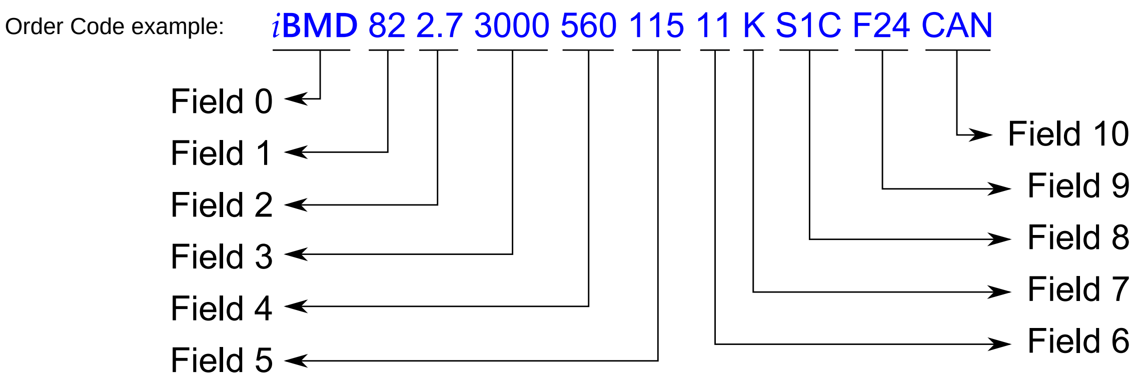



*Figure 29.1. Position of the fields in the ManufacturerDeviceName.*

| ManufacturerDeviceName | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Field** | **Description** | **Value** | | | | | | | | | | |
| Series | Integrated brushless drive | *i*BMD | | | | | | | | | | |
| Motor size | Flange 82mm; 2.7Nm, 3.8Nm | | 82 | | | | | | | | | |
| | Flange 102mm; 5.1Nm, 6.2Nm | | 102 | | | | | | | | | |
| | Flange 145mm; 14.5Nm, 18.5Nm | | 145 | | | | | | | | | |
| | Flange 170mm; 29Nm, 36Nm | | 170 | | | | | | | | | |
| Motor stall torque | Flange 82mm; 2.7Nm | | | 2.7 | | | | | | | | |
| | Flange 82mm; 3.8Nm | | | 3.8 | | | | | | | | |
| | Flange 102mm; 5.1Nm | | | 5.1 | | | | | | | | |
| | Flange 102mm; 6.2Nm | | | 6.2 | | | | | | | | |
| | Flange 145mm; 14.5Nm | | | 14.5 | | | | | | | | |
| | Flange 145mm; 18.5Nm | | | 18.5 | | | | | | | | |
| | Flange 170mm; 29Nm | | | 29 | | | | | | | | |
| | Flange 170mm; 36Nm | | | 36 | | | | | | | | |
| Motor rated speed | All drives | | | | 3000 | | | | | | | |
| Nominal DC voltage | All drives | | | | | 560 | | | | | | |
| Mechanical interface | Size 82, 102 | | | | | | 100 | | | | | |
| | Size 82, 102 | | | | | | 115 | | | | | |
| | Size 145, 170 | | | | | | 165 | | | | | |
| Shaft diameter | Size 82 | | | | | | | 11 | | | | |
| | Size 82 | | | | | | | 14 | | | | |
| | Size 82, 102, 145 | | | | | | | 19 | | | | |
| | Size 102, 145, 170 | | | | | | | 24 | | | | |
| | Size 145, 170 | | | | | | | 28 | | | | |
| | Size 170 | | | | | | | 32 | | | | |
| Shaft keway | With key | | | | | | | | K | | | |
| | Without key | | | | | | | | NK | | | |
| Absolute encoders | Single turn (16 sin/cos) | | | | | | | | | S1C | | |
| | Single turn (128 sin/cos) | | | | | | | | | S1O | | |
| | Multi turn (16 sin/cos - 4096 turns) | | | | | | | | | M1C | | |
| | Multi turn (128 sin/cos - 4096 turns) | | | | | | | | | M1O | | |
| Brake or Flywheel | Brake 24Vdc | | | | | | | | | | F24 | |
| | Flywheel | | | | | | | | | | F1 | |
| Communication interface | Fieldbus | | | | | | | | | | | CAN |
| | | | | | | | | | | | | ETC |

***Table 29.1. Fields that make up the ManufacturerDeviceName.***

NOTE

**In regards to the Table 29.1 not all the combinations are possible. Please contact the sales office to verify the order possibilities.**

Bonfiglioli has been designing and developing innovative and reliable power transmission and control solutions for industry, mobile machinery and renewable energy applications since 1956.

**Bonfiglioli Riduttori S.p.A.**
Via Giovanni XXIII, 7/A
40012 Lippo di Calderara di Reno
Bologna, Italy

Tel: +39 051 647 3111
Fax: +39 051 647 3126
bonfiglioli@bonfiglioli.com
www.bonfiglioli.com

Ed. 2.1 - English