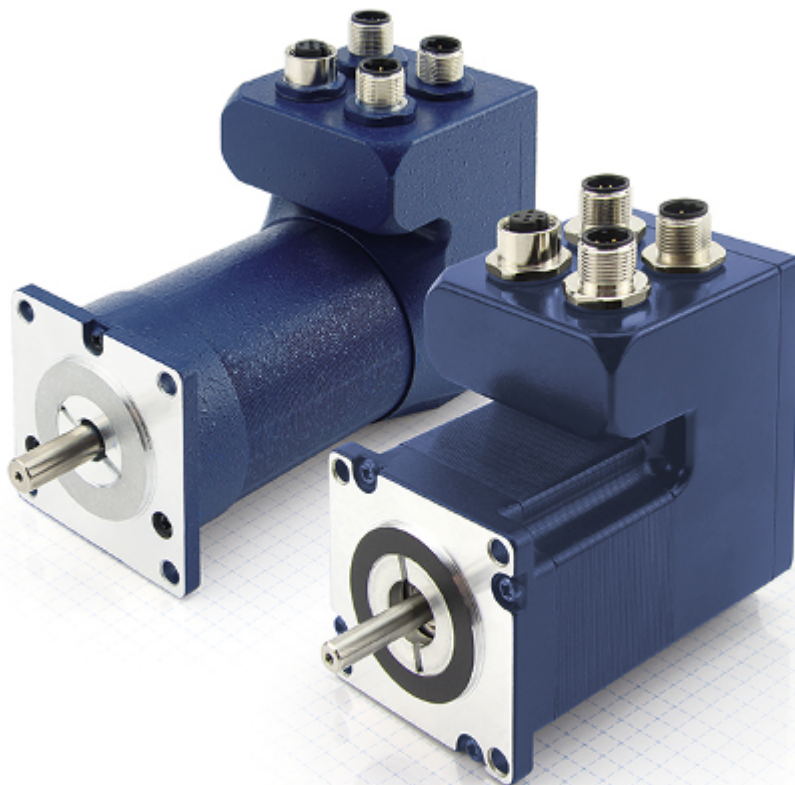


Technical Manual PD4-E

Fieldbus: EtherCAT

For use with the following devices:

PD4-E591L42-E-65-1, PD4-E601L42-E-65-1, PD4-EB59CD-E-65-1



Valid with firmware version FIR-v1748
and since hardware version W002b

Technical Manual Version: 1.0.1

Contents

| | |
|---|-----------|
| 1 Introduction..... | 7 |
| 1.1 Version information..... | 7 |
| 1.2 Copyright, marking and contact..... | 7 |
| 1.3 Intended use..... | 7 |
| 1.4 Warranty and disclaimer..... | 7 |
| 1.5 Specialist staff..... | 8 |
| 1.6 Other applicable regulations..... | 8 |
| 1.7 EU directives for product safety..... | 8 |
| 1.8 Used icons..... | 8 |
| 1.9 Emphasis in the text..... | 8 |
| 1.10 Numerical values..... | 9 |
| 1.11 Bits..... | 9 |
| 1.12 Counting direction (arrows)..... | 9 |
| 2 Safety and warning notices..... | 10 |
| 3 Technical details and pin assignment..... | 11 |
| 3.1 Environmental conditions..... | 11 |
| 3.2 Dimensioned drawings..... | 11 |
| 3.3 Electrical properties and technical data..... | 12 |
| 3.4 Overtemperature protection..... | 13 |
| 3.5 LED signaling..... | 15 |
| 3.6 Pin assignment..... | 18 |
| 4 Commissioning..... | 23 |
| 4.1 Establishing communication..... | 24 |
| 4.2 Auto setup..... | 24 |
| 4.3 Test run..... | 27 |
| 4.4 Special drive modes (clock-direction and analog speed)..... | 28 |
| 5 General concepts..... | 31 |
| 5.1 Control modes..... | 31 |
| 5.2 CiA 402 Power State Machine..... | 35 |
| 5.3 User-defined units..... | 40 |
| 5.4 Limitation of the range of motion..... | 45 |
| 5.5 Cycle times..... | 45 |
| 6 Operating modes..... | 47 |
| 6.1 Profile Position..... | 47 |
| 6.2 Velocity..... | 55 |
| 6.3 Profile Velocity..... | 56 |
| 6.4 Profile Torque..... | 59 |
| 6.5 Homing..... | 61 |
| 6.6 Interpolated Position Mode..... | 68 |
| 6.7 Cyclic Synchronous Position..... | 69 |
| 6.8 Cyclic Synchronous Velocity..... | 71 |
| 6.9 Cyclic Synchronous Torque..... | 72 |

| | |
|---|------------|
| 6.10 Clock-direction mode..... | 73 |
| 6.11 Auto setup..... | 75 |
| 7 Special functions..... | 77 |
| 7.1 Digital inputs and outputs..... | 77 |
| 7.2 I ² t Motor overload protection..... | 86 |
| 7.3 Saving objects..... | 87 |
| 8 EtherCAT..... | 92 |
| 9 Programming with NanoJ..... | 93 |
| 9.1 NanoJ program..... | 93 |
| 9.2 Mapping in the NanoJ program..... | 96 |
| 9.3 System calls in a NanoJ program..... | 98 |
| 10 Description of the object dictionary..... | 100 |
| 10.1 Overview..... | 100 |
| 10.2 Structure of the object description..... | 100 |
| 10.3 Object description..... | 100 |
| 10.4 Value description..... | 101 |
| 10.5 Description..... | 102 |
| 1000h Device Type..... | 103 |
| 1001h Error Register..... | 104 |
| 1003h Pre-defined Error Field..... | 105 |
| 1008h Manufacturer Device Name..... | 109 |
| 1009h Manufacturer Hardware Version..... | 109 |
| 100Ah Manufacturer Software Version..... | 110 |
| 1010h Store Parameters..... | 110 |
| 1011h Restore Default Parameters..... | 113 |
| 1018h Identity Object..... | 117 |
| 1020h Verify Configuration..... | 118 |
| 1600h Receive PDO 1 Mapping Parameter..... | 120 |
| 1601h Receive PDO 2 Mapping Parameter..... | 122 |
| 1602h Receive PDO 3 Mapping Parameter..... | 124 |
| 1603h Receive PDO 4 Mapping Parameter..... | 127 |
| 1A00h Transmit PDO 1 Mapping Parameter..... | 129 |
| 1A01h Transmit PDO 2 Mapping Parameter..... | 131 |
| 1A02h Transmit PDO 3 Mapping Parameter..... | 134 |
| 1A03h Transmit PDO 4 Mapping Parameter..... | 136 |
| 1C00h Sync Manager Communication Type..... | 139 |
| 1C12h Sync Manager PDO Assignment..... | 140 |
| 1C13h Sync Manager PDO Assignment..... | 142 |
| 1C32h Output Sync Manager Synchronization..... | 143 |
| 1C33h Input Sync Manager Synchronization..... | 144 |
| 1F50h Program Data..... | 146 |
| 1F51h Program Control..... | 147 |
| 1F57h Program Status..... | 148 |
| 2030h Pole Pair Count..... | 149 |
| 2031h Maximum Current..... | 150 |
| 2034h Upper Voltage Warning Level..... | 151 |
| 2035h Lower Voltage Warning Level..... | 151 |
| 2036h Open Loop Current Reduction Idle Time..... | 152 |
| 2037h Open Loop Current Reduction Value/factor..... | 152 |
| 2038h Brake Controller Timing..... | 153 |
| 2039h Motor Currents..... | 155 |

| | |
|--|-----|
| 203Ah Homing On Block Configuration..... | 156 |
| 203Bh I2t Parameters..... | 158 |
| 203Dh Torque Window..... | 160 |
| 203Eh Torque Window Time Out..... | 161 |
| 203Fh Max Slippage Time Out..... | 161 |
| 2056h Limit Switch Tolerance Band..... | 162 |
| 2057h Clock Direction Multiplier..... | 163 |
| 2058h Clock Direction Divider..... | 163 |
| 2059h Encoder Configuration..... | 163 |
| 205Ah Absolute Sensor Boot Value (in User Units)..... | 164 |
| 205Bh Clock Direction Or Clockwise/Counter Clockwise Mode..... | 165 |
| 2084h Bootup Delay..... | 165 |
| 2101h Fieldbus Module Availability..... | 166 |
| 2102h Fieldbus Module Control..... | 167 |
| 2103h Fieldbus Module Status..... | 168 |
| 2110h EtherCAT Slave Status..... | 170 |
| 2300h NanoJ Control..... | 171 |
| 2301h NanoJ Status..... | 171 |
| 2302h NanoJ Error Code..... | 172 |
| 230Fh Uptime Seconds..... | 174 |
| 2310h NanoJ Input Data Selection..... | 174 |
| 2320h NanoJ Output Data Selection..... | 175 |
| 2330h NanoJ In/output Data Selection..... | 177 |
| 2400h NanoJ Inputs..... | 178 |
| 2410h NanoJ Init Parameters..... | 179 |
| 2500h NanoJ Outputs..... | 180 |
| 2600h NanoJ Debug Output..... | 181 |
| 2701h Customer Storage Area..... | 182 |
| 2800h Bootloader And Reboot Settings..... | 183 |
| 3202h Motor Drive Submode Select..... | 184 |
| 3203h Feedback Selection..... | 185 |
| 3204h Feedback Mapping..... | 187 |
| 3210h Motor Drive Parameter Set..... | 188 |
| 3212h Motor Drive Flags..... | 192 |
| 3220h Analog Inputs..... | 194 |
| 3221h Analogue Inputs Control..... | 195 |
| 3240h Digital Inputs Control..... | 196 |
| 3241h Digital Input Capture..... | 199 |
| 3242h Digital Input Routing..... | 200 |
| 3243h Digital Input Homing Capture..... | 202 |
| 3250h Digital Outputs Control..... | 204 |
| 3252h Digital Output Routing..... | 207 |
| 3320h Read Analogue Input..... | 207 |
| 3321h Analogue Input Offset..... | 209 |
| 3322h Analogue Input Pre-scaling..... | 210 |
| 33A0h Feedback Incremental A/B/I 1..... | 211 |
| 3700h Deviation Error Option Code..... | 212 |
| 4012h HW Information..... | 213 |
| 4013h HW Configuration..... | 214 |
| 4014h Operating Conditions..... | 215 |
| 4015h Special Drive Modes..... | 217 |
| 4016h Factory Settings..... | 218 |
| 4040h Drive Serial Number..... | 219 |
| 4041h Device Id..... | 219 |
| 603Fh Error Code..... | 220 |
| 6040h Controlword..... | 220 |
| 6041h Statusword..... | 221 |
| 6042h VI Target Velocity..... | 222 |
| 6043h VI Velocity Demand..... | 223 |

| | |
|---|-----|
| 6044h VI Velocity Actual Value..... | 223 |
| 6046h VI Velocity Min Max Amount..... | 224 |
| 6048h VI Velocity Acceleration..... | 225 |
| 6049h VI Velocity Deceleration..... | 226 |
| 604Ah VI Velocity Quick Stop..... | 227 |
| 604Ch VI Dimension Factor..... | 228 |
| 605Ah Quick Stop Option Code..... | 229 |
| 605Bh Shutdown Option Code..... | 230 |
| 605Ch Disable Option Code..... | 231 |
| 605Dh Halt Option Code..... | 231 |
| 605Eh Fault Option Code..... | 232 |
| 6060h Modes Of Operation..... | 233 |
| 6061h Modes Of Operation Display..... | 233 |
| 6062h Position Demand Value..... | 234 |
| 6063h Position Actual Internal Value..... | 234 |
| 6064h Position Actual Value..... | 235 |
| 6065h Following Error Window..... | 235 |
| 6066h Following Error Time Out..... | 236 |
| 6067h Position Window..... | 236 |
| 6068h Position Window Time..... | 237 |
| 606Bh Velocity Demand Value..... | 238 |
| 606Ch Velocity Actual Value..... | 238 |
| 606Dh Velocity Window..... | 239 |
| 606Eh Velocity Window Time..... | 239 |
| 6071h Target Torque..... | 240 |
| 6072h Max Torque..... | 240 |
| 6074h Torque Demand..... | 241 |
| 6075h Motor Rated Current..... | 241 |
| 6077h Torque Actual Value..... | 242 |
| 607Ah Target Position..... | 242 |
| 607Bh Position Range Limit..... | 243 |
| 607Ch Home Offset..... | 244 |
| 607Dh Software Position Limit..... | 244 |
| 607Eh Polarity..... | 245 |
| 607Fh Max Profile Velocity..... | 246 |
| 6080h Max Motor Speed..... | 247 |
| 6081h Profile Velocity..... | 248 |
| 6082h End Velocity..... | 248 |
| 6083h Profile Acceleration..... | 248 |
| 6084h Profile Deceleration..... | 249 |
| 6085h Quick Stop Deceleration..... | 249 |
| 6086h Motion Profile Type..... | 250 |
| 6087h Torque Slope..... | 250 |
| 608Fh Position Encoder Resolution..... | 251 |
| 6090h Velocity Encoder Resolution..... | 252 |
| 6091h Gear Ratio..... | 253 |
| 6092h Feed Constant..... | 254 |
| 6096h Velocity Factor..... | 255 |
| 6097h Acceleration Factor..... | 256 |
| 6098h Homing Method..... | 257 |
| 6099h Homing Speed..... | 258 |
| 609Ah Homing Acceleration..... | 259 |
| 60A2h Jerk Factor..... | 259 |
| 60A4h Profile Jerk..... | 261 |
| 60A8h SI Unit Position..... | 262 |
| 60A9h SI Unit Velocity..... | 263 |
| 60B0h Position Offset..... | 263 |
| 60B1h Velocity Offset..... | 264 |
| 60B2h Torque Offset..... | 264 |

| | |
|--|-----|
| 60C1h Interpolation Data Record..... | 265 |
| 60C2h Interpolation Time Period..... | 266 |
| 60C4h Interpolation Data Configuration..... | 267 |
| 60C5h Max Acceleration..... | 269 |
| 60C6h Max Deceleration..... | 269 |
| 60E4h Additional Position Actual Value..... | 270 |
| 60E5h Additional Velocity Actual Value..... | 271 |
| 60E6h Additional Position Encoder Resolution - Encoder Increments..... | 272 |
| 60E8h Additional Gear Ratio - Motor Shaft Revolutions..... | 273 |
| 60E9h Additional Feed Constant - Feed..... | 274 |
| 60EBh Additional Position Encoder Resolution - Motor Revolutions..... | 275 |
| 60EDh Additional Gear Ratio - Driving Shaft Revolutions..... | 276 |
| 60EEh Additional Feed Constant - Driving Shaft Revolutions..... | 277 |
| 60F2h Positioning Option Code..... | 278 |
| 60F4h Following Error Actual Value..... | 280 |
| 60F8h Max Slippage..... | 280 |
| 60FAh Control Effort..... | 281 |
| 60FCh Position Demand Internal Value..... | 282 |
| 60FDh Digital Inputs..... | 282 |
| 60FEh Digital Outputs..... | 283 |
| 60FFh Target Velocity..... | 284 |
| 6502h Supported Drive Modes..... | 284 |
| 6503h Drive Catalogue Number..... | 285 |
| 6505h Http Drive Catalogue Address..... | 286 |

11 Copyrights..... 287

| | |
|--------------------------------------|-----|
| 11.1 Introduction..... | 287 |
| 11.2 AES..... | 287 |
| 11.3 MD5..... | 287 |
| 11.4 uIP..... | 288 |
| 11.5 DHCP..... | 288 |
| 11.6 CMSIS DSP Software Library..... | 288 |
| 11.7 FatFs..... | 289 |
| 11.8 Protothreads..... | 289 |
| 11.9 lwIP..... | 289 |

1 Introduction

The *PD4-E* is a brushless motor with integrated controller in protection class IP65. The integrated absolute encoder makes immediate operation possible in *closed loop* mode without homing.

This manual describes the functions of the controller and the available operating modes. It also shows how you can address and program the controller via the communication interface.

You can find further information on the product on the Nanotec website us.nanotec.com

1.1 Version information

| Manual version | Date | Changes | Firmware version | Hardware version |
|----------------|---------|---------------------------------|------------------|------------------|
| 1.0.0 | 11/2017 | First edition | FIR-v1738 | W002b |
| 1.0.1 | 02/2018 | Additions and error corrections | FIR-v1748 | W002b |

1.2 Copyright, marking and contact

Copyright © 2013 – 2018 Nanotec® Electronic GmbH & Co. KG. All rights reserved.



Nanotec® Electronic GmbH & Co. KG
Kapellenstraße 6
D-85622 Feldkirchen/Munich

Phone: +49 (0)89-900 686-0

Fax: +49 (0)89-900 686-50

Internet: us.nanotec.com

TwinCAT™ is a registered trademark of the Beckhoff Automation GmbH™.

1.3 Intended use

The *PD4-E* is designed for use under the approved **Environmental conditions**.

Any other use is considered unintended use.



Note

Changes or modifications to the product are not permitted.

1.4 Warranty and disclaimer

Nanotec produces component parts that are used in a wide range of industrial applications. The selection and use of Nanotec products is the responsibility of the system engineer and end user. Nanotec accepts no responsibility for the integration of the products in the end system.

Under no circumstances may a Nanotec product be integrated as a safety controller in a product or construction. All products containing a component part manufactured by Nanotec must, upon delivery to the end user, be provided with corresponding warning notices and instructions for safe use and safe operation. All warning notices provided by Nanotec must be passed on directly to the end user.

Our general terms and conditions apply: en.nanotec.com/service/general-terms-and-conditions/.

1.5 Specialist staff

Only specialists may install, program and commission the device:

- Persons who have appropriate training and experience in work with motors and their control.
- Persons who are familiar with and understand the content of this technical manual.
- Persons who know the applicable regulations.

1.6 Other applicable regulations

In addition to this technical manual, the following regulations are to be observed:

- Accident-prevention regulations
- Local regulations on occupational safety

1.7 EU directives for product safety

The following EU directives were observed:

- RoHS directive (2011/65/EU, 2015/863/EU)
- EMC directive (2014/30/EU)

1.8 Used icons

All notices are in the same format. The degree of the hazard is divided into the following classes.



CAUTION

- The CAUTION notice indicates a possibly dangerous situation.
- Failure to observe the notice **may** result in moderately severe injuries.
- Describes how you can avoid the danger.



Note

- Indicates an error source or likelihood of confusion.
- Failure to observe the notice **may** result in damage to this or other devices.
- Describes how device damage can be avoided.



Tip

Shows a tip for the application or task.

1.9 Emphasis in the text

The following conventions are used in the document:

Text set in **bold** indicates cross references and hyperlinks:

- The following bits in object **6041_h** (statusword) have a special function:

- A list of available system calls can be found in chapter **System calls in a NanoJ program**.

Text set in *italics* marks named objects:

- Read the *installation manual*.
- Use the *Plug & Drive Studio* software to perform the auto setup.
- For software: You can find the corresponding information in the *Operation* tab.
- For hardware: Use the *ON/OFF* switch to switch the device on.

A text set in *Courier* marks a code section or programming command:

- The line with the `od_write(0x6040, 0x00, 5);` command has no effect.
- The NMT message is structured as follows: `000 | 81 2A`

A text in "quotation marks" marks user input:

- Start the NanoJ program by writing object `2300h`, bit 0 = "1".
- If a holding torque is already needed in this state, the value "1" must be written in `3212h:01h`.

1.10 Numerical values

Numerical values are generally specified in decimal notation. The use of hexadecimal notation is indicated by a subscript *h* at the end of the number.

The objects in the object dictionary are written with index and subindex as follows:

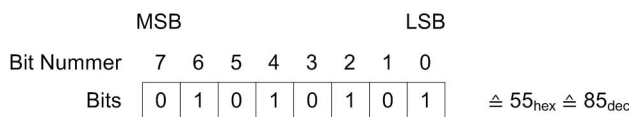
<Index>:<Subindex>

Both the index as well as the subindex are specified in hexadecimal notation. If no subindex is listed, the subindex is `00h`.

Example: Subindex 5 of object `1003h` is addressed with `1003h:05h`, subindex 00 of object `6040h` with `6040h`.

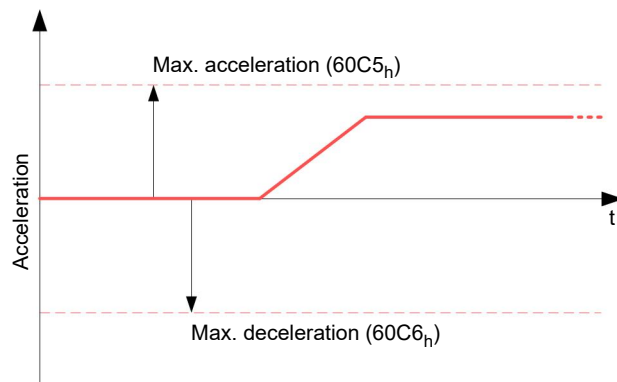
1.11 Bits

The numbering of individual bits in an object always begins with the LSB (bit number 0). See the following figure, which uses data type *UNSIGNED8* as an example.



1.12 Counting direction (arrows)

In figures, the counting direction is always in the direction of an arrow. Objects `60C5h` and `60C6h` depicted as examples in the following figure are both specified as positive.



2 Safety and warning notices



Note

- Damage to the controller.
- Changing the wiring during operation may damage the controller.
- Only change the wiring in a de-energized state. After switching off, wait until the capacitors have discharged.



Note

- Fault of the controller due to excitation voltage of the motor.
- Voltage peaks during operation may damage the controller.
- Install suitable circuits (e.g., charging capacitor) that reduce voltage peaks.



Note

- There is no polarity reversal protection.
- Polarity reversal results in a short-circuit between supply voltage and GND (earth) via the power diode.
- Install a line protection device (fuse) in the supply line.



Note

- The device contains components that are sensitive to electrostatic discharge.
- Improper handling can damage the device.
- Observe the basic principles of ESD protection when handling the device.

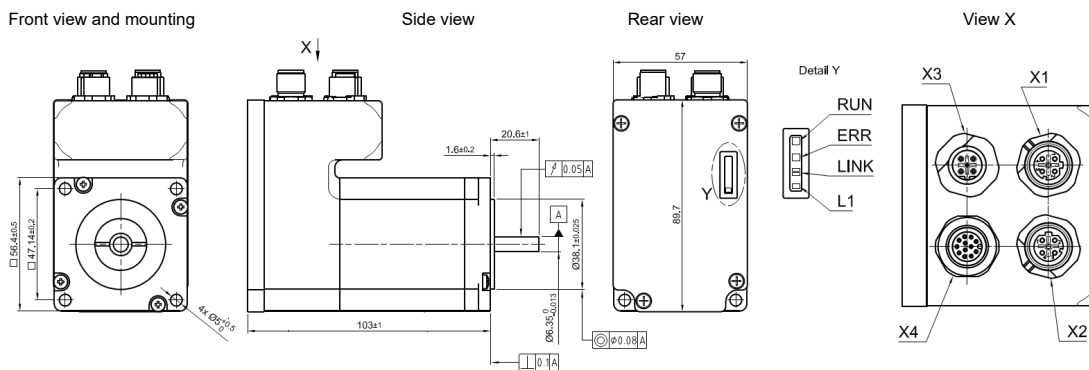
3 Technical details and pin assignment

3.1 Environmental conditions

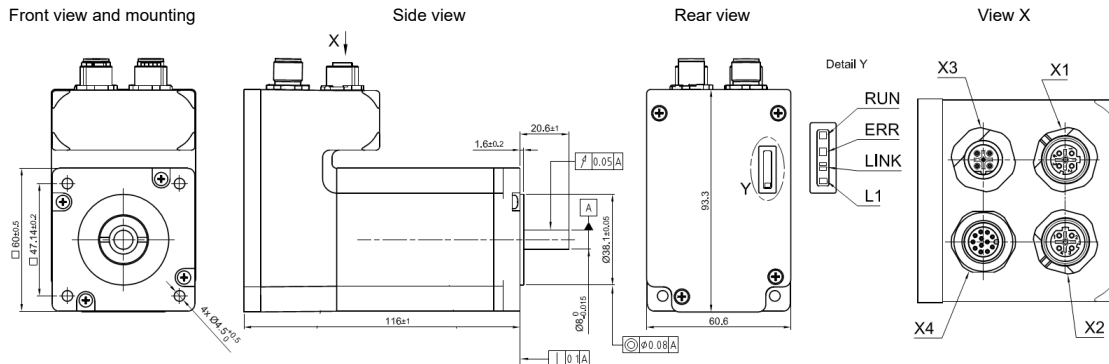
| Environmental condition | Value |
|---|--------------------------------|
| Protection class | IP65 (except for shaft output) |
| Ambient temperature (operation) | -10 ... +40°C |
| Air humidity (non-condensing) | 0 ... 85% |
| Altitude of site above <i>sea level</i> (without drop in performance) | 1500 m |
| Ambient temperature (storage) | -25 ... +85°C |

3.2 Dimensioned drawings

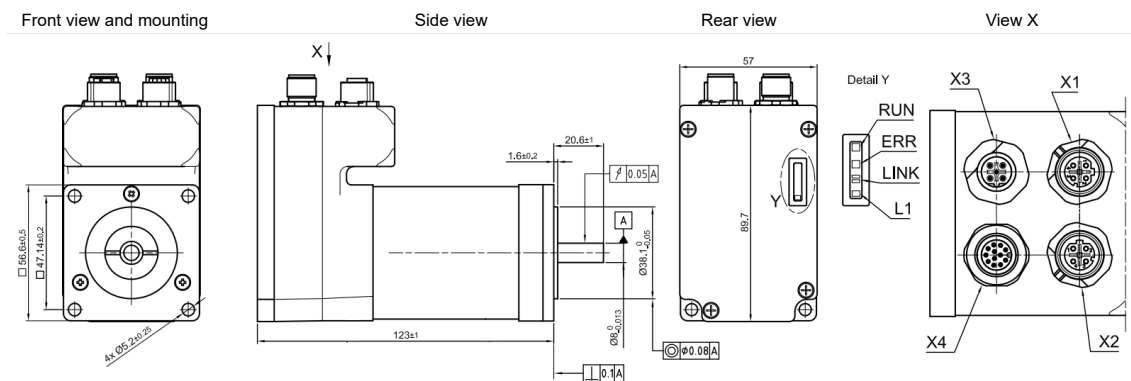
3.2.1 PD4-E591L42-E-65-1



3.2.2 PD4-E601L42-E-65-1



3.2.3 PD4-EB59CD-E-65-1



3.3 Electrical properties and technical data

3.3.1 Technical data – motor

| | PD4-E | PD4-EB |
|--------------------------|------------------------------------|--------------------------|
| Type | High-pole DC servo (stepper motor) | Low-pole DC servo (BLDC) |
| Operating voltage | 12 - 48 V DC ±5% | 12 - 48 V DC ±5% |
| Phase current rms | 4.2 A | 6 A |
| Peak current rms for 1 s | N/A | max. 18 A |

3.3.2 Motor performance data

| Type | Holding torque Ncm | Rated / peak torque Ncm | Rated speed (rpm) |
|-------------|--------------------|-------------------------|-------------------|
| PD4-E591L42 | 187 | N/A | N/A |
| PD4-E601L42 | 354 | N/A | N/A |

| Type | Holding torque Ncm | Rated / peak torque Ncm | Rated speed (rpm) |
|------------|--------------------|-------------------------|-------------------|
| PD4-EB59CD | N/A | 60 / 180 | 3500 |

3.3.3 Technical data

| | |
|---------------------------------|--|
| Operating modes | <i>Profile Position Mode, Profile Velocity Mode, Profile Torque Mode, Velocity Mode, Homing Mode, Interpolated Position Mode, Cyclic Sync Position Mode, Cyclic Sync Velocity Mode, Cyclic Synchronous Torque Mode, Clock-Direction Mode</i> |
| Set value setting / programming | <i>EtherCAT, analog input, NanoJ program</i> |
| Inputs | 6 digital inputs (+5 V/+24 V DC), individually switchable by means of software, factory settings: 5 V 1 analog input 0–10 V or 0–20 mA (switchable by means of software) |
| Outputs | 2 outputs, <i>Open Drain</i> , max. 100 mA |
| Integrated encoder | Magnetic, single-turn absolute encoder, 1024 pulses/revolution |
| Protection circuit | Overvoltage and undervoltage protection Overtemperature protection (> 68° Celsius on rear cover) Polarity reversal protection: In the event a polarity reversal, a short-circuit will occur between supply voltage and GND over a power diode; a fuse is therefore necessary in the supply line. The values of the fuse are dependent on the application and must be dimensioned <ul style="list-style-type: none"> • greater than the maximum current consumption of the controller • less than the maximum current of the voltage supply. If the fuse value is very close to the maximum current consumption of the controller, a medium / slow tripping characteristics should be used. |

3.4 Overtemperature protection

Above a temperature of approx. 75°C on the power board (corresponds to 65–72°C outside on the back cover), the power part of the controller switches off and the error bit is set (see objects **1001_n** and **1003_n**). After cooling down and confirming the error (see **table for the controlword**, "Fault reset"), the controller again functions normally.

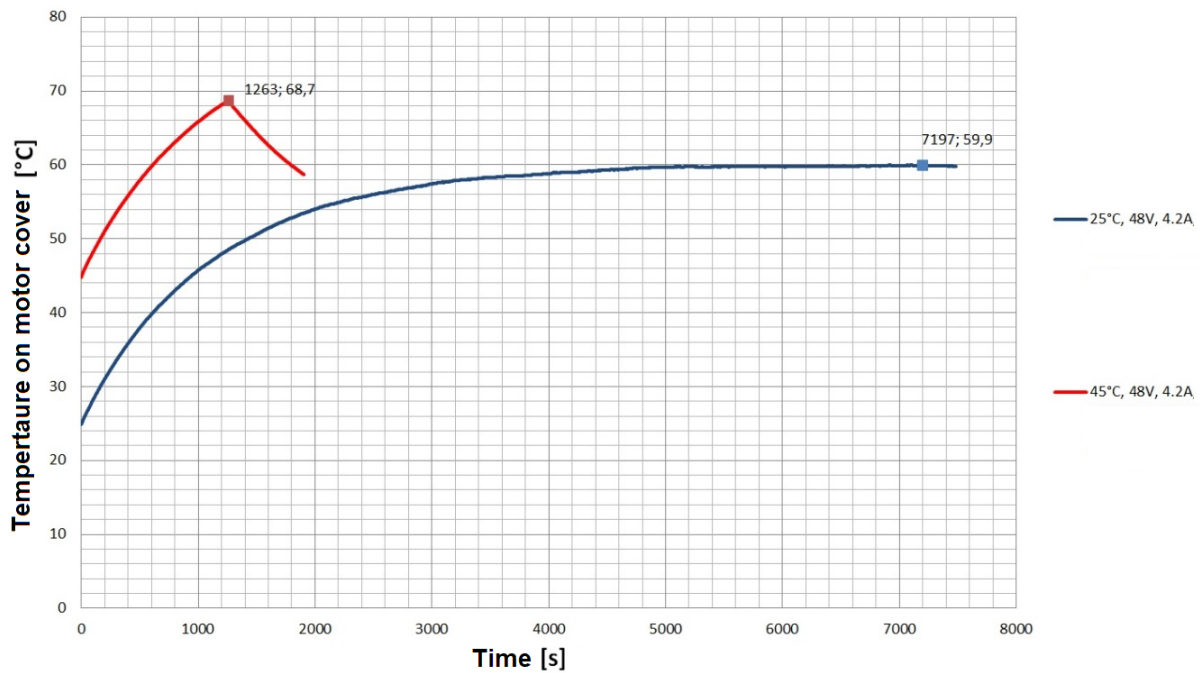
The following temperature test results provide information on the temperature behavior of this controller.

Temperature tests are performed under the following conditions:

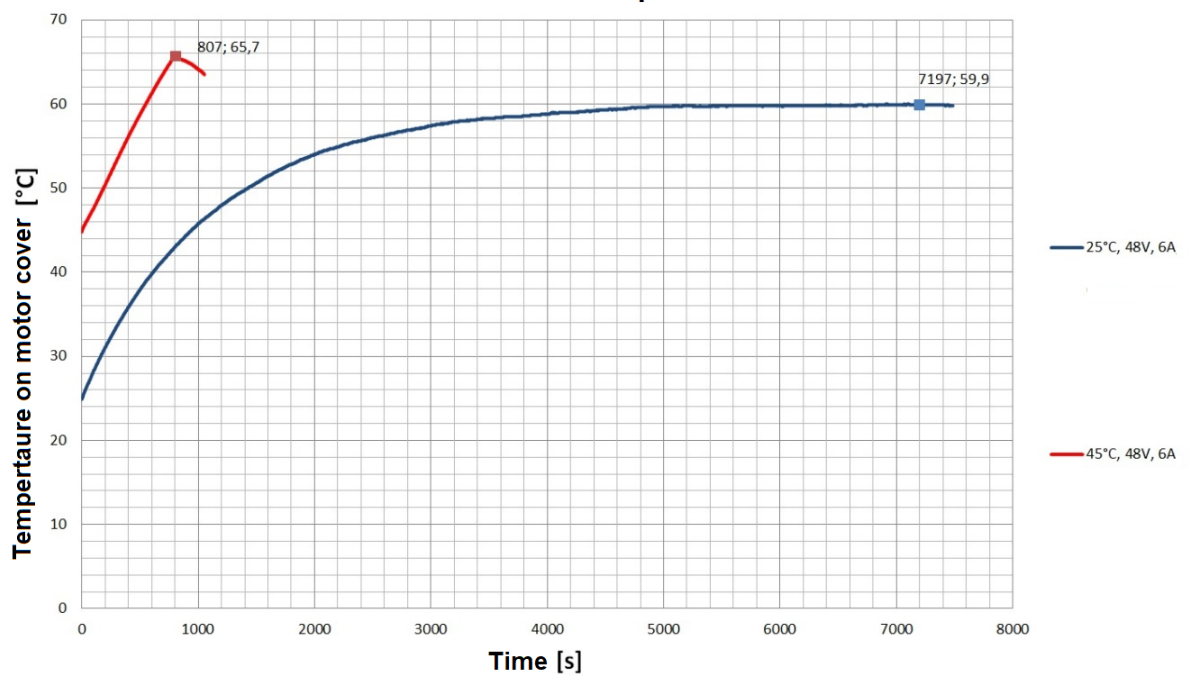
- Operating voltage: 48 V DC
- Motor current: 4.2 A (stepper motor) / 6 A (BLDC motor) rms
- Operation mode: Velocity Mode, full step, 30 rpm
- Ambient temperature: 25 °C / 45 °C
- Altitude of site: 500 m above seal level
- No external cooling in the climatic chamber, e.g., via fan

The following graphics show the results of the temperature tests:

PD4-E Temperature



PD4-EB59CD Temperature



Summary:

At 25°C (+48 V, 4.2/6 A rms, Velocity Mode 30 rpm), the controller was in operation for longer than 2 hours without having been switched off. The temperature on the cover was stable at approx. 60°C.

At 45°C (+48 V, 4.2/6 A rms, Velocity Mode 30 rpm), temperature protection switched off the controller after approx. 21 (stepper motor) or 14 (BLDC motor) minutes.

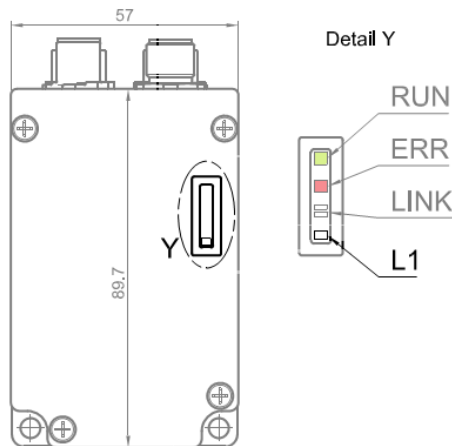


Note

Aside from the motor, the exact temperature behavior is also dependent on the flange connection and the heat transfer there as well as on the convection in the application. For this reason, we recommend always performing an endurance test in the actual environment for applications in which current level and ambient temperature pose a problem.

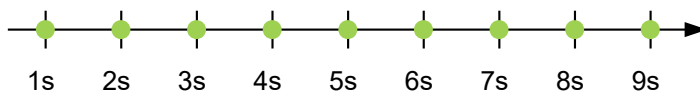
3.5 LED signaling

3.5.1 Power LED



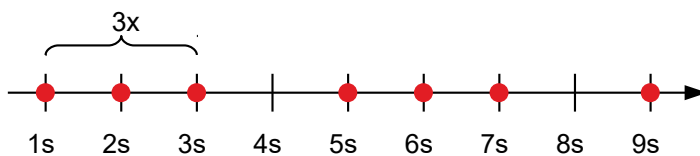
Normal operation

In normal operation, the green power LED L1 flashes briefly once per second.



Case of an error

If an error has occurred, the LED turns red and signals an error number. In the following figure, the error number 3 is signaled.



The following table shows the meaning of the error numbers.

| Flash rate | Error |
|------------|-------------|
| 1 | General |
| 2 | Voltage |
| 3 | Temperature |

| Flash rate | Error |
|------------|----------------|
| 4 | Overcurrent |
| 5 | Controller |
| 6 | Watchdog-Reset |



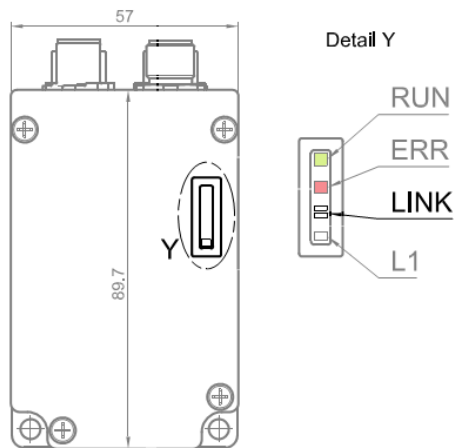
Note

For each error that occurs, a more precise error code is stored in object **1003_h**.

3.5.2 EtherCAT LEDs

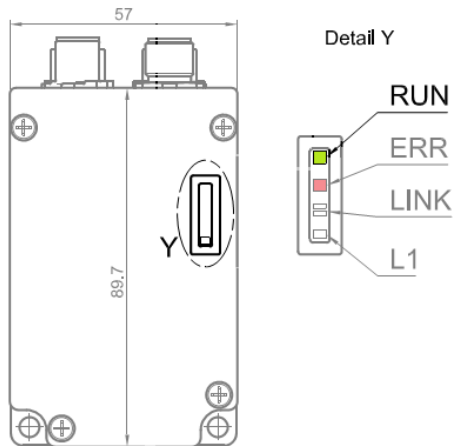
The *PD4-E* is equipped with three LEDs (*LINK*, *RUN*, *ERR*) which indicate the status of the controller on the EtherCAT bus.

LINK



The green *LINK* LED is on if the EtherCAT cable is connected and flashes during data transfer.

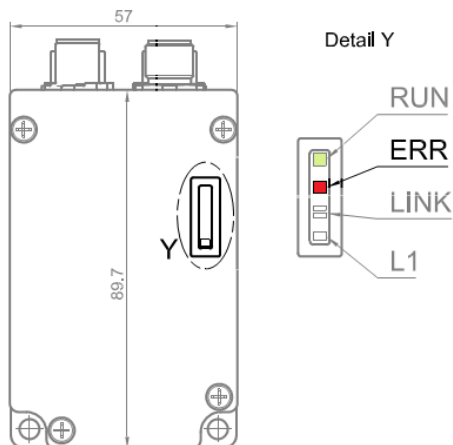
RUN



The green *RUN* LED indicates one of the following controller states (*EtherCAT slave*):

| LED behavior | Status | Prerequisite |
|--------------------|-----------------------------|---|
| Permanently off | INIT | The <i>slave</i> is in the initialisation state. |
| Flashing | PRE-OPERATIONAL | The <i>slave</i> is in the <i>PRE-OPERATIONAL</i> state. |
| Single flash | SAFE-OPERATIONAL | The <i>slave</i> is in the <i>SAFE-OPERATIONAL</i> state. |
| Flickering | INITIALISATION or BOOTSTRAP | The <i>slave</i> is in the initialisation phase and has not yet achieved the <i>INIT</i> status or the <i>slave</i> is in <i>boot loader mode</i> and firmware is being loaded. |
| continuously green | OPERATIONAL | The <i>slave</i> is in the <i>OPERATIONAL</i> state. |

ERR



The red *ERR* LED indicates one of the following controller states (*EtherCAT slave*):

| LED behavior | Status | Prerequisite |
|------------------|--|---|
| Permanently off | No error | Not an error |
| Flashing | Invalid Configuration | General configuration error |
| Single flash | Local Error | The <i>slave</i> automatically changed its <i>EtherCAT status</i> due to a local error. |
| Double flash | Process Data Watchdog Timeout/ EtherCAT Watchdog Timeout | <i>Watchdog</i> timeout |
| Triple flash | Application Error | Error in the <i>power state machine</i> of the <i>slave</i> |
| Flickering | Booting Error | Error when loading the firmware |
| continuously red | Application controller failure | The <i>slave</i> no longer responds. |

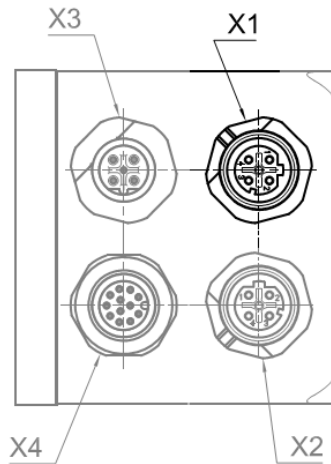
3.6 Pin assignment

3.6.1 Overview

| Connector | Function |
|-----------|--|
| X1 | EtherCAT IN |
| X2 | EtherCAT OUT |
| X3 | Voltage supply |
| X4 | Inputs/outputs and external logic supply |
| L1 | Power LED, see chapter Power LED |
| RUN | EtherCAT RUN LED, see chapter EtherCAT LEDs |
| ERR | EtherCAT ERROR LED, see chapter EtherCAT LEDs |
| LINK | EtherCAT LINK LED, see chapter EtherCAT LEDs |

3.6.2 Connector X1 - EtherCAT IN

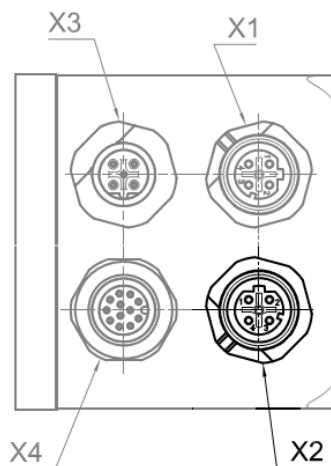
Connection for EtherCAT. Connector type: M12, 4-pin, D-coded, female



| Pin | Function | Note |
|-----|----------|---------------------|
| 1 | TD+ | Transmission Data + |
| 2 | RD+ | Receiver Data + |
| 3 | TD- | Transmission Data - |
| 4 | RD- | Receiver Data - |

3.6.3 Connector X2 - EtherCAT OUT

Connection for EtherCAT. Connector type: M12, 4-pin, D-coded, female

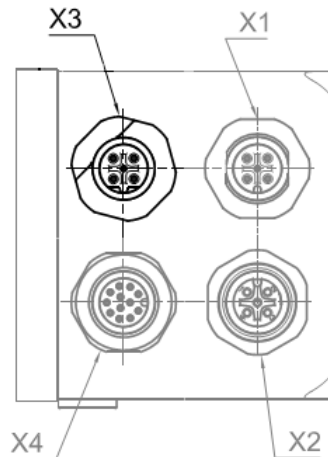


| Pin | Function | Note |
|-----|----------|---------------------|
| 1 | TD+ | Transmission Data + |
| 2 | RD+ | Receiver Data + |

| Pin | Function | Note |
|-----|----------|---------------------|
| 3 | TD- | Transmission Data - |
| 4 | RD- | Receiver Data - |

3.6.4 Connector X3 - voltage supply

Connection for the main supply. Connector type: M12, 5-pin, B-coded, male



Voltage source

The operating or supply voltage supplies a battery, a transformer with rectification and filtering, or a switching power supply.



Note

- EMC: For a DC power supply line longer than 30 m or when using the motor on a DC bus, additional interference-suppression and protection measures are necessary.
- An EMI filter is to be inserted in the DC supply line as close as possible to the controller/motor.
- Long data or supply lines are to be routed through ferrites.

Pin assignment

| Pin | Function | Note |
|-----|----------|------------------------|
| 1 | +Ub | 12 - 48 V DC $\pm 5\%$ |
| 2 | +Ub | 12 - 48 V DC $\pm 5\%$ |
| 3 | GND | |
| 4 | GND | |
| 5 | n.c. | Not used |

Permissible operating voltage

The maximum operating voltage is 51.5 V DC. If the input voltage of the controller exceeds this threshold value, the motor is switched off and an error triggered. Above 50.5 V, the integrated ballast circuit (3 W power) is activated.

The minimum operating voltage is 10 V DC. If the input voltage of the controller falls below this threshold value, the motor is switched off and an error triggered.

A charging capacitor of at least 4700 µF / 50 V (approx. 1000 µF per ampere rated current) must be connected to the supply voltage to avoid exceeding the permissible operating voltage (e.g., during braking).

3.6.5 Connector X4 - inputs/outputs and external logic supply

Connection for the digital and analog inputs/outputs and the external logic supply. Connector type: M12, 12-pin, A-coded, male

| Pin | Function | Note |
|-----|------------------|---|
| 1 | GND | |
| 2 | Digital input 1 | 5 V / 24 V signal, switchable by means of software with object 3240_h , max. 1 MHz |
| 3 | Digital input 2 | 5 V / 24 V signal, switchable by means of software with object 3240_h , max. 1 MHz |
| 4 | Digital input 3 | 5 V / 24 V signal, switchable by means of software with object 3240_h , max. 1 MHz |
| 5 | Digital input 4 | 5 V / 24 V signal, switchable by means of software with object 3240_h , max. 1 MHz |
| 6 | Digital input 5 | 5 V / 24 V signal, switchable by means of software with object 3240_h , max. 1 MHz |
| 7 | Digital input 6 | 5 V / 24 V signal, switchable by means of software with object 3240_h , max. 1 MHz |
| 8 | Analog input | 10 bit, 0 V...+10 V or 0...20 mA, switchable by means of software with object 3221_h |
| 9 | Digital output 1 | Digital output, open drain, max. 24 V / 100 mA |
| 10 | Digital output 2 | Digital output, open drain, max. 24 V / 100 mA |
| 11 | Voltage output | +5 V, max. 100 mA |
| 12 | +UB Logic | +24 V DC, input voltage for the logic supply, current consumption: approx. 39 mA |



Note

The windings of the motor are not supplied by the logic supply.

In **4015_h**, you can activate the alternative function of the digital inputs, which is used for the *special drive modes*. See chapter **Special drive modes (clock-direction and analog speed)**.

If you set **3240_h:07_h** to the value "1", three differential inputs are available instead of six single-ended inputs.

The following table shows all possible combinations:

| Pin | Basic function | | Alternative function | |
|-----|---|---|----------------------|--------------|
| | Single-ended | Differential | Single-ended | Differential |
| 2 | Input 1 | - Input 1 | Release | -Release |
| 3 | Input 2 / direction input in clock-direction mode | + Input 1 | Direction | Release |
| 4 | Input 3 / clock input in clock-direction mode | - Input 2 / - Direction input in clock-direction mode | Clock | -Direction |
| 5 | Input 4 | + Input 2 / + Direction input in clock-direction mode | Digital input 4 | Direction |
| 6 | Input 5 | - Input 3 / - Clock input in clock-direction mode | Digital input 5 | -Clock |
| 7 | Input 6 | + Input 3 / + Clock input in clock-direction mode | Digital input 6 | Clock |

The following switching thresholds apply for inputs 1 to 6:

| Max. Voltage | Switching thresholds | |
|--------------|----------------------|-----------------|
| | Safe switch on | Safe switch off |
| 5 V | > 4.09 V | < 0.95 V |
| 24 V | > 14.74 V | < 3.78 V |

4 Commissioning

Described in this chapter is how you establish communication with the controller and set the necessary parameters to make the motor ready for operation.

The *Plug & Drive Studio* software offers you an option for performing the configuration and adapting the motor parameters to your application. You can find further information in document *Plug & Drive Studio: Quick Start Guide* at us.nanotec.com.

The controller also offers you the possibility to switch *special drive modes* on/off via object **4015_n**. You can thereby control the motor directly via the inputs (analog input/clock-direction). See chapter **Special drive modes (clock-direction and analog speed)** for details.

Observe the following notes:



CAUTION

- Moving parts can cause hand injuries.
- If you touch moving parts during running operation, hand injuries may result.
- Do not reach for moving parts during operation. After switching off, wait until all movements have ended.



CAUTION

- In free-standing operation, motor movements are uncontrolled and can cause injuries.
- If the motor is unsecured, it can, e.g., fall down. Foot injuries or damage to the motor could occur.
- If you operate the motor free-standing, observe the motor, switch it off immediately in the event of danger and make certain that the motor cannot fall down.



CAUTION

- Moving parts can catch hair and loose clothing.
- During running operation, moving parts can catch hair or loose clothing, which may lead to injuries.
- If you have long hair, wear a hairnet or take other suitable protective measures when near moving parts. Do not work with loose clothing or ties near moving parts.



CAUTION

- Risk of overheating or fire if there is insufficient cooling.
- If cooling is insufficient or if the ambient temperature is too high, there is a risk of overheating or fire.
- During use, make certain that the cooling and ambient temperature conditions are ensured.



Note

- EMC: Current-carrying cables – particularly around supply and motor cables – produce electromagnetic alternating fields.
- These can interfere with the motor and other devices. Nanotec recommends the following measures:
- Use shielded cables and earth the cable shielding on both ends over a short distance.
- Use cables with cores in twisted pairs.
- Keep power supply and motor cables as short as possible.
- Earth motor housing with large contact area over a short distance.
- Lay supply, motor and control cables separately.

4.1 Establishing communication

4.1.1 EtherCAT

Before commissioning, we recommend reading chapter **Technical details and pin assignment**.

Software connection



Tip

The following description assumes that an EtherCAT master from Beckhoff with the *TwinCAT* software is used.

1. Connect the EtherCAT master to the controller, see **Technical details and pin assignment**.
2. Supply the controller with voltage.
3. Obtain the *ESI file* that corresponds exactly to the used **firmware version** from the following sources:
 - a. From the Nanotec website **www.nanotec.de**. The current version of the firmware and the *ESI file* can be found in the *Plug & Drive Studio* download folder.
 - b. From Nanotec support.
4. Close the *TwinCAT* system manager if it is open.
5. Then copy the *ESI file* to the *TwinCAT* subfolder:
 - If you use *TwinCAT* version 2, use folder `<TWINCAT INSTALL DIR>/Io/EtherCAT`
 - If you use *TwinCAT* version 3, use folder `<TWINCAT INSTALL DIR>/3.1/Config/IO/EtherCAT`

Example

Example: If *TwinCAT* 2 is installed on your computer under path `C:\TwinCAT\`, copy the *ESI file* to path `C:\TwinCAT\Io\EtherCAT\`.

6. Open the *ESI file* with an editor. Find the *AddInfo* parameter. Enter:
 - the value "2" if you would like to integrate the controller as *Box* (factory settings)
 - the value "0" if you would like to integrate the controller as *NC-Axis*Save and close the file.
7. Now restart the *TwinCAT* system manager. The *ESI files* are read in again following a restart.



Note

The cycle time of the sync signal must always be set to 1 ms. You can set the bus cycle time (and, consequently, the interpolation time in **60C2_n**) to integer multiples of 1 ms.

4.2 Auto setup

To determine a number of parameters related to the motor and the connected sensors (encoders/Hall sensors), an auto setup is performed. **Closed Loop** operation requires a successfully completed auto setup.



Note

- Note the following prerequisites for performing the auto setup:
- The motor must be load-free.
- The motor must not be touched.
- The motor must be able to turn freely in any direction.
- No NanoJ programs may be running (object 2300_h:00_h bit 0 = "0", see **2300h NanoJ Control**).



Tip

Execution of the auto setup requires a relatively large amount of processor computing power. During the auto setup, this may result in fieldbuses not being operated in a timely manner.



Note

The limit switches and, thus, the tolerance bands are active in this mode. For further information on the limit switches, see **Limitation of the range of motion**.



Tip

As long as the motor connected to the controller or the sensors for feedback (encoders/Hall sensors) are not changed, auto setup is only to be performed once during initial commissioning.



Note

With the Plug & Drive motors, it is not necessary to perform an auto setup, as this was already performed at the factory.

4.2.1 Parameter determination

Auto setup determines various parameters of the connected motor and of the present sensors by means of multiple test runs and measurement runs. To a certain extent, the type and number of parameters are dependent on the respective motor configuration.

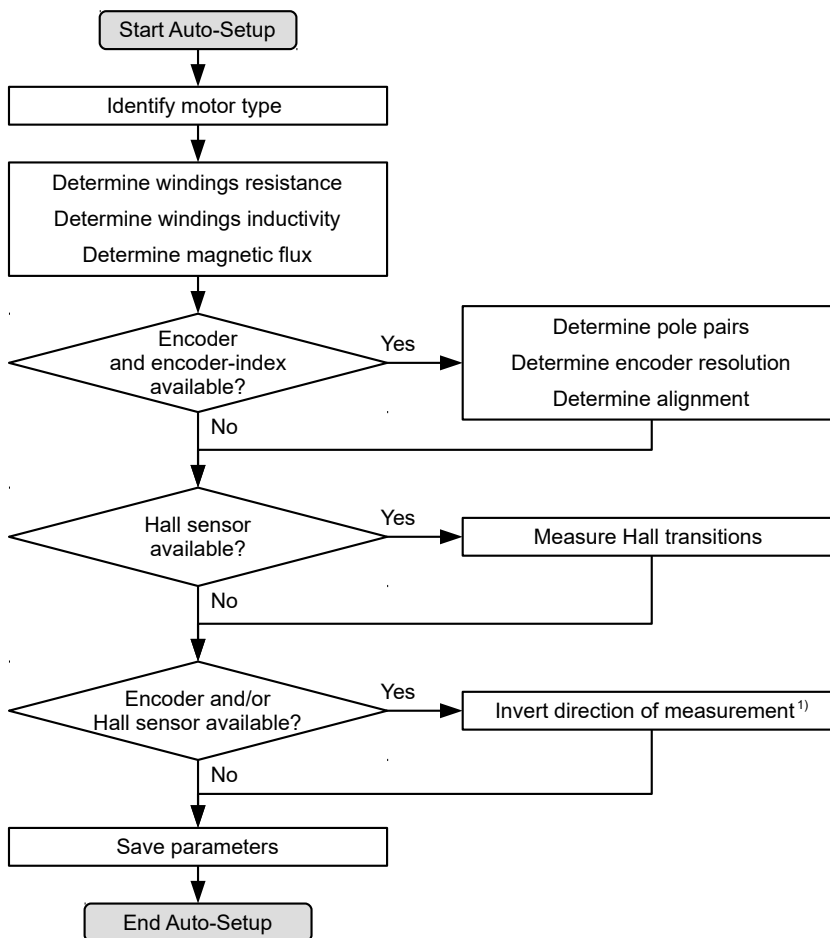
| Parameter | All motors independent of the configuration |
|--|---|
| Motor type (stepper motor or BLDC motor) | X |
| Winding resistance | X |
| Winding inductance | X |
| Interlinking flux | X |

| Parameter | Motor without encoder | Motor with encoder and index | Motor with encoder without index |
|---|-----------------------|------------------------------|----------------------------------|
| Encoder resolution | - | X | --- |
| Alignment (shifting of the electrical zero to the index.) | - | X | --- |

| Parameter | Motor without Hall sensor | Motor with Hall sensor |
|------------------|---------------------------|------------------------|
| Hall transitions | - | X |

4.2.2 Execution

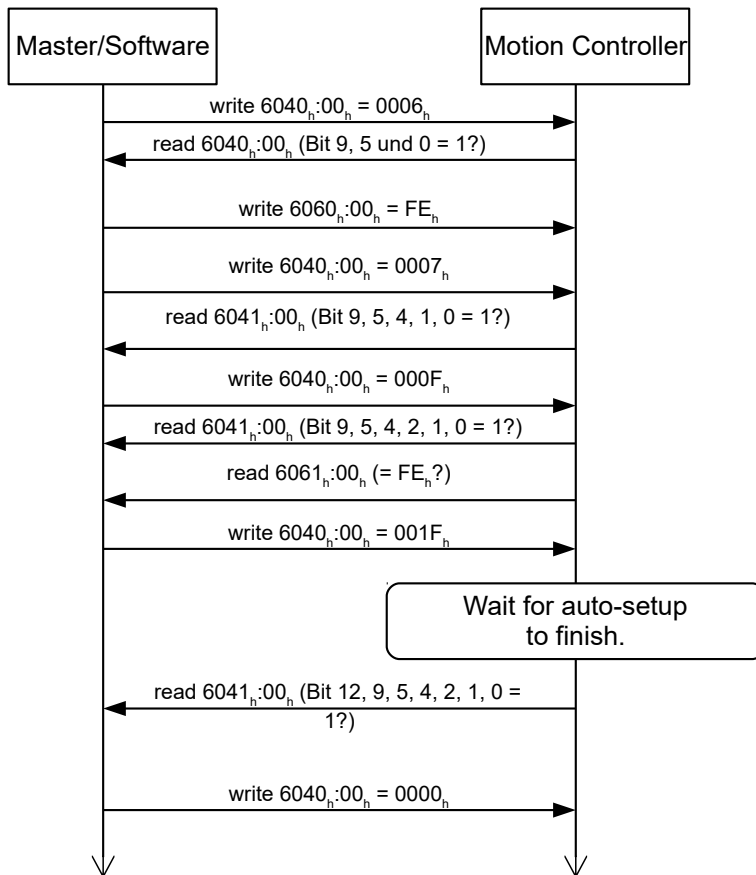
1. To preselect the *auto setup* operating mode, enter the value "-2" ("FE_h") in object 6060_h:00_h. The *power state machine* must now switch to the *Operation enabled* state, see **CiA 402 Power State Machine**.
2. Start *auto setup* by setting bit 4 *OMS* in object 6040_h:00_h (controlword).



While the auto setup is running, the following tests and measurements are performed in succession:

- 1) To determine the values, the direction of the measurement method is reversed and edge detection re-evaluated.

Value 1 in bit 12 *OMS* in object 6041_h:00_h (statusword) indicates that the auto setup was completely executed and ended. In addition, bit 10 *TARG* in object 6041_h:00_h can be used to query whether (= "1") or not (= "0") an encoder index was found.



4.2.3 Parameter memory

After a successful *auto setup*, the determined parameter values are automatically taken over into the corresponding objects and stored with the storage mechanism, see **Saving objects** and **1010h Store Parameters**. Categories *Drive* 1010h:05h and *Tuning* 1010h:06h are used.



CAUTION

- After executing auto setup mode, the internal coordinate system is no longer valid.
- *Homing* alone does not suffice! If the controller is not restarted, unexpected reactions may result.
- Restart the device after an auto setup!

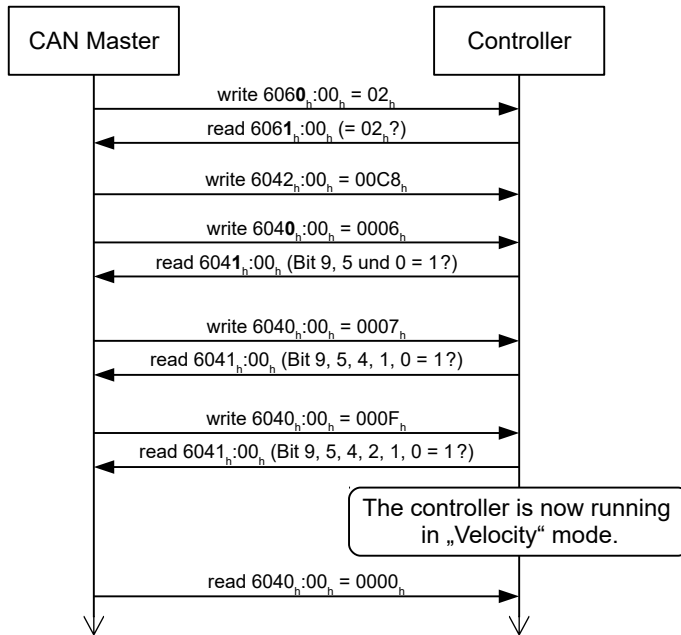
4.3 Test run

After configuring and the auto setup, a test run can be performed. As an example, the **Velocity** operating mode is used.

The values are transferred from your *EtherCAT master* to the controller. After every transfer, the *master* should use the status objects of the controller to ensure successful parameterization.

1. Select the *Velocity* mode by setting object **6060h** (Modes Of Operation) to the value "2".
2. Write the desired speed in **6042h**.
3. Switch the *power state machine* to the *Operation enabled* state, see **CiA 402 Power State Machine**.

The following sequence starts *Velocity* mode; the motor turns at 200 rpm.



4. To stop the motor, set controlword (**6040_h**) to "0".

4.4 Special drive modes (clock-direction and analog speed)

You have the possibility to control the motor directly via the clock and direction input or the analog input by activating the *special drive modes*. These include:

- **Clock-direction**
- **Analog speed**
- **Test run with 30 rpm**

You can also determine the **control mode** – *open loop* or *closed loop*.

Digital input 1 serves here as an enable (see **Connector X4 - inputs/outputs and external logic supply**).



Note

After activating the *special drive modes*, the state of the **CiA 402 Power State Machine** is controlled only via a digital input (enable). State changes that are requested in object **6040_h** (controlword) have no effect.

4.4.1 Activation

To activate the *special drive modes*, you must enter the value "2" in **4015_h:01_h**. In **4015_h:02_h**, set the mode by writing a value between "00"_h and "0F"_h.

The following table lists all possible modes and their value for **4015:02_h**:

| Value | Mode | | | |
|-----------------|----------------------------|----------------------|---------------------------------|-----------|
| 00 _h | Clock-direction | - | - | Open Loop |
| 01 _h | Clock-direction | - | - | Open Loop |
| 02 _h | Clock-direction (test run) | Test run with 30 rpm | Clockwise direction of rotation | Open Loop |

| Value | Mode | | | |
|-----------------|----------------------------|---------------------------------|--|-------------|
| 03 _h | Clock-direction (test run) | Test run with 30 rpm | Counterclockwise direction of rotation | Open Loop |
| 04 _h | Analog speed | Direction via "Direction" input | Maximum speed 1000 rpm | Open Loop |
| 05 _h | Analog speed | Direction via "Direction" input | Maximum speed 100 rpm | Open Loop |
| 06 _h | Analog speed | Offset 5 V (joystick mode) | Maximum speed 1000 rpm | Open Loop |
| 07 _h | Analog speed | Offset 5 V (joystick mode) | Maximum speed 100 rpm | Open Loop |
| 08 _h | Clock-direction | - | - | Closed Loop |
| 09 _h | Clock-direction | - | - | Closed Loop |
| 0A _h | Clock-direction (test run) | Test run with 30 rpm | Clockwise direction of rotation | Closed Loop |
| 0B _h | Clock-direction (test run) | Test run with 30 rpm | Counterclockwise direction of rotation | Closed Loop |
| 0C _h | Analog speed | Direction via "Direction" input | Maximum speed 1000 rpm | Closed Loop |
| 0D _h | Analog speed | Direction via "Direction" input | Maximum speed 100 rpm | Closed Loop |
| 0E _h | Analog speed | Offset 5 V (joystick mode) | Maximum speed 1000 rpm | Closed Loop |
| 0F _h | Analog speed | Offset 5 V (joystick mode) | Maximum speed 100 rpm | Closed Loop |

You must save object **4015_h** (*application category*) (see chapter **Saving objects**); the changes do not take effect until after the controller is restarted.

4.4.2 Clock-direction

The controller internally sets the operating mode to **clock-direction**. You must connect the *enable*, *clock* and *direction* inputs (see chapter **Connector X4 - inputs/outputs and external logic supply**).

4.4.3 Analog speed

The controller internally sets the operating mode to **Velocity**. To preset the speed, the voltage on the analog input is used and the corresponding target speed is written in **6042_h**.

Maximum speed

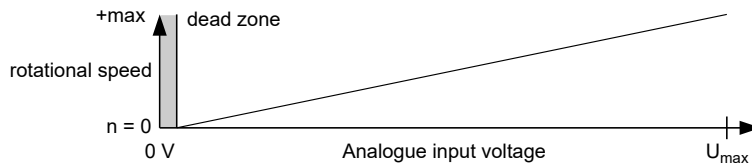
The maximum speed can be changed between 100 rpm and 1000 rpm. If a different speed is necessary, it can be set using the scaling factor (object **604C_h** subindices 01_h and 02_h).

Computation of the analog voltage

There are two modes for calculating the analog input voltage.

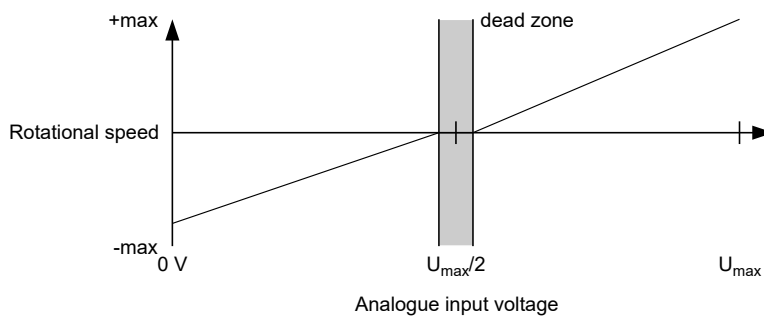
Normal mode

You must connect the *enable*, *direction* and *analog inputs* (see chapter **Connector X4 - inputs/outputs and external logic supply**). The maximum analog voltage corresponds to the maximum speed. The direction is preset here via the direction input. There is a dead zone from 0 V to 20 mV in which the motor does not move.



Joystick mode

You must connect the *enable input* and the *analog input* (see chapter **Connector X4 - inputs/ outputs and external logic supply**). Half of the maximum analog voltage corresponds to the speed 0. If the voltage drops below half, the speed increases in the negative direction. If the speed rises above half, the speed increases likewise in the positive direction. The dead zone here extends from $U_{max}/2 \pm 20 \text{ mV}$.



4.4.4 Test run with 30 rpm

The motor rotates at 30 rpm if the *enable input* is set.

5 General concepts

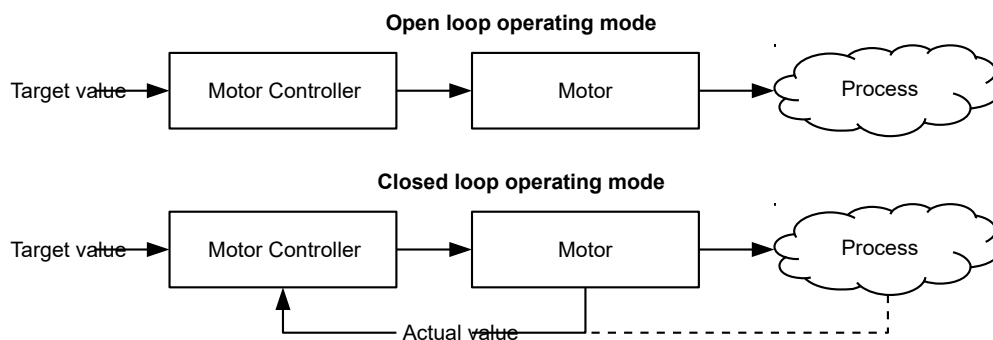
5.1 Control modes

5.1.1 General

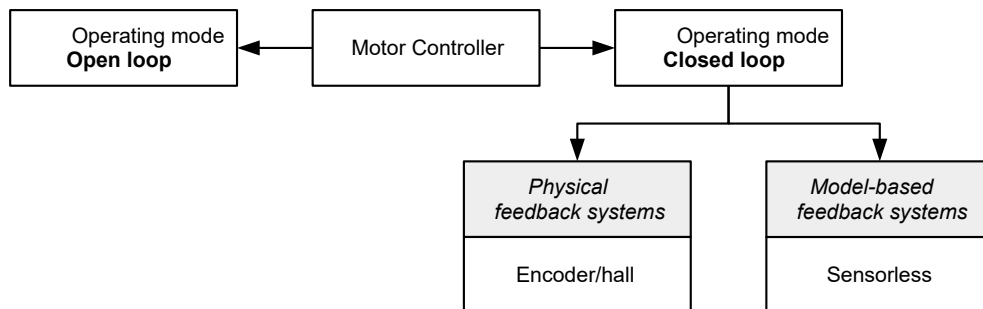
The control mode of systems without feedback is called *open loop*, the mode with feedback is called *closed loop*. In the *closed loop* control mode, it is initially irrelevant whether the fed back signals come from the motor itself or from the influenced process.

For controllers with feedback, the measured control variable (actual value) is constantly compared with a set point (set value). In the event of deviations between these values, the controller readjusts according to the specified control parameters.

Pure controllers, on the other hand, have no feedback for the value that is to be regulated. The set point (set value) is only specified.



In addition to the physical feedback systems (e.g., via encoders or Hall sensors), model-based feedback systems, collectively referred to as sensorless systems, are also used. Both feedback systems can also be used in combination to further improve the control quality.



Summarized in the following are all possible combinations of control modes and feedback systems with respect to the motor technology. Support of the respective control mode and feedback is controller-specific and is described in chapters *Pin assignment* and **Operating modes**.

| Control mode | Stepper motor | BLDC motor |
|--------------|---------------|------------|
| Open Loop | yes | no |
| Closed Loop | yes | yes |

| Feedback | Stepper motor | BLDC motor |
|----------|---------------|------------|
| Hall | no | yes |
| Encoder | yes | yes |

| Feedback | Stepper motor | BLDC motor |
|------------|---------------|------------|
| Sensorless | yes | yes |

Various operating modes can be used depending on the control mode. The following list contains all the types of operation that are possible in the various control modes.

| Operating mode | Control mode | |
|-----------------------------|-------------------|-------------|
| | Open Loop | Closed Loop |
| Profile Position | yes | yes |
| Velocity | yes | yes |
| Profile Velocity | yes | yes |
| Profile Torque | no ¹⁾ | yes |
| Homing | yes ²⁾ | yes |
| Interpolated Position Mode | yes ³⁾ | yes |
| Cyclic Synchronous Position | yes ³⁾ | yes |
| Cyclic Synchronous Velocity | yes ³⁾ | yes |
| Cyclic Synchronous Torque | no ¹⁾ | yes |
| Clock-direction | yes | yes |

1) The **Profile Torque** and **Cyclic Synchronous Torque** torque operating modes are not possible in the *open loop* control mode due to a lack of feedback.

2) Exception: Homing on block is not possible due to a lack of feedback.

3) Because ramps and speeds in operating modes **Cyclic Synchronous Position** and **Cyclic Synchronous Velocity** follow from the specified points of the master, it is not normally possible to preselect these parameters and to ascertain whether a step loss can be excluded. It is therefore not advisable to use these operating modes in combination with *open loop* control mode.

5.1.2 Open Loop

Introduction

Open loop mode is only used with stepper motors and is, by definition, a control mode without feedback. The field rotation in the stator is specified by the controller. The rotor directly follows the magnetic field rotation without step losses as long as no limit parameters, such as the maximum possible torque, are exceeded. Compared to *closed loop*, no complex internal control processes are needed in the controller. As a result, the requirements on the controller hardware and the controller logic are very low. *Open loop* mode is used primarily with price-sensitive applications and simple movement tasks.

Because, unlike *closed loop*, there is no feedback for the current rotor position, no conclusion can be drawn on the counter torque being applied to the output side of the motor shaft. To compensate for any torque fluctuations that arise on the output shaft of the motor, in *open loop* mode, the controller always supplies the maximum possible (e.g., specified by parameters) set current to the stator windings over the entire speed range. The high magnetic field strength thereby produced forces the rotor to assume the new steady state in a very short time. This torque is, however, opposite that of rotor's inertia. Under certain operating conditions, this combination is prone to resonances, comparable to a spring-mass system.

Commissioning

To use *open loop* mode, the following settings are necessary:

- In object **2030_h** (Pole Pair Count), enter the number of pole pairs (see motor data sheet: for a stepper motor with 2 phases, a step angle of 1.8° corresponds to 50 pole pairs and 0.9° corresponds to 100 pole pairs).
- In object **2031_h** (Max Current), enter the maximum current in mA (see motor data sheet).
- In object **3202_h** (Motor Drive Submode Select), set bit 0 (CL/OL) to the value "0".
- If the clock-direction mode is to be used, then observe chapter **Clock-direction mode**.

If necessary, current reduction on motor standstill should be activated to reduce the power loss and heat build-up. To activate current reduction, the following settings are necessary:

- In object **3202_h** (Motor Drive Submode Select), set bit 3 (CurRed) to "1".
- In object **2036_h** (Open Loop Current Reduction Idle Time), the time in milliseconds is specified that the motor must be at a standstill before current reduction is activated.
- In object **2037_h** (Open Loop Current Reduction Value/factor), the root mean square is specified to which the rated current is to be reduced if current reduction is activated in *open loop* and the motor is at a standstill.

Optimizations

Depending on the system, resonances may occur in *open loop* mode; susceptibility to resonances is particularly high at low loads. Practical experience has shown that, depending on the application, various measures are effective for largely reducing resonances:

- Reduce or increase current, see object **2031_h** (Max Current). Excessive torque reserve promotes resonances.
- Reduce or increase the operating voltage, taking into account the product-specific ranges (with sufficient torque reserve). The permissible operating voltage range can be found in the product data sheet.
- Optimize the control parameters of the current controller via objects **3210_h:09_h** (I_P) and **3210_h:0A_h** (I_L).
- Adjustments to the acceleration, deceleration and/or target speed depending on the selected control mode:

Profile Position operating mode

Objects **6083_h** (Profile Acceleration), **6084_h** (Profile Deceleration) and **6081_h** (Profile Velocity).

Velocity operating mode

Objects **6048_h** (Velocity Acceleration), **6049_h** (Velocity Deceleration) and **6042_h** (Target Velocity).

Profile Velocity operating mode

Objects **6083_h** (Profile Acceleration), **6084_h** (Profile Deceleration) and **6081_h** (Profile Velocity).

Homing operating mode

Objects **609A_h** (Homing Acceleration), **6099_h:01_h** (Speed During Search For Switch) and **6099_h:02_h** (Speed During Search For Zero).

Interpolated Position Mode operating mode

The acceleration and deceleration ramps can be influenced with the higher-level controller.

Cycle Synchronous Position operating mode

The acceleration and deceleration ramps can be influenced via the external "position specification / time unit" targets.

Cycle Synchronous Velocity operating mode

The acceleration and deceleration ramps can be influenced via the external "position specification / time unit" targets.

Clock-Direction operating mode

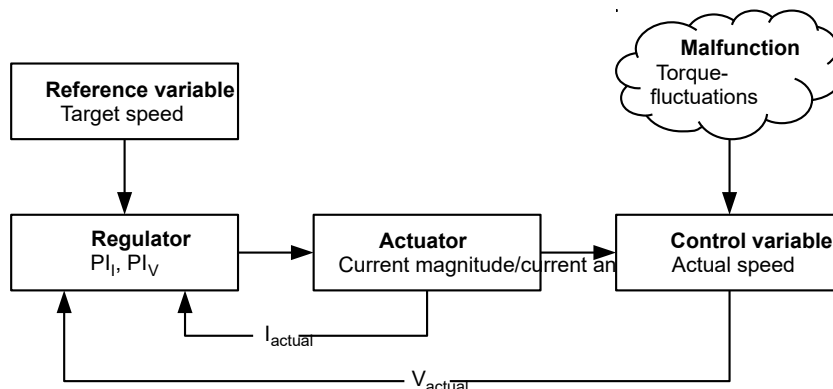
Change of the step resolution via objects **2057_h** (Clock Direction Multiplier) and **2058_h** (Clock Direction Divider). Optimize acceleration / deceleration ramps by adjusting the pulse frequency to pass through the resonance range as quickly as possible.

5.1.3 Closed Loop

Introduction

The *closed loop* theory is based on the idea of a control loop. A disturbance acting on a system should be compensated for quickly and without lasting deviation to adjust the control variable back to the set point.

Closed loop using a speed control as an example:



- PI_I = Proportional-integral current control loop
- PI_V = Proportional-integral velocity control loop
- I_{actual} = Actual current
- V_{actual} = Actual speed

The *closed loop* method is also referred to as "sine commutation via an encoder with field-oriented control". At the heart of *closed loop* technology is the performance-adjusted current control as well as the feedback of the actual values of the process. Using the encoder signals, the rotor orientation is recorded and sinusoidal phase currents generated in the motor windings. Vector control of the magnetic field ensures that the magnetic field of the stator is always perpendicular to that of the rotor and that the field strength corresponds precisely to the desired torque. The current thereby controlled in the windings provides a uniform motor force and results in an especially smooth-running motor that can be precisely regulated.

The feedback of the control variables necessary for *closed loop* mode can be realized with various technologies. In addition to the physical feedback with encoders or Hall sensors, it is also possible to virtually record the motor parameters through software-based model calculation. Physical variables, such as speed or back-EMF, can be reconstructed with the help of a so-called "observer" from the data of the current controller. With this sensorless technology, one has a "virtual rotary encoder", which – above a certain minimum speed – supplies the position and speed information with the same precision as a real optical or magnetic encoder.

All controllers from Nanotec that support *closed loop* mode implement a field oriented control with sine commutated current control. Thus, the stepper motors and BLDC motor are controlled in the same way as a servo motor. With *closed loop* mode, step angle errors can be compensated for during travel and load angle errors corrected within one full step.

Commissioning

An auto setup must be performed before using *closed loop* mode. The auto setup operating mode automatically determines the necessary parameters (e.g., motor data, feedback systems) that are necessary for optimum operation of the field oriented control. All information necessary for performing the auto setup can be found in chapter **Auto setup**.

With the Plug & Drive motors, it is not necessary to perform the *auto setup*, as this was already performed at the factory

Bit 0 in **3202_h** must be set .

5.2 CiA 402 Power State Machine

5.2.1 State machine

CiA 402

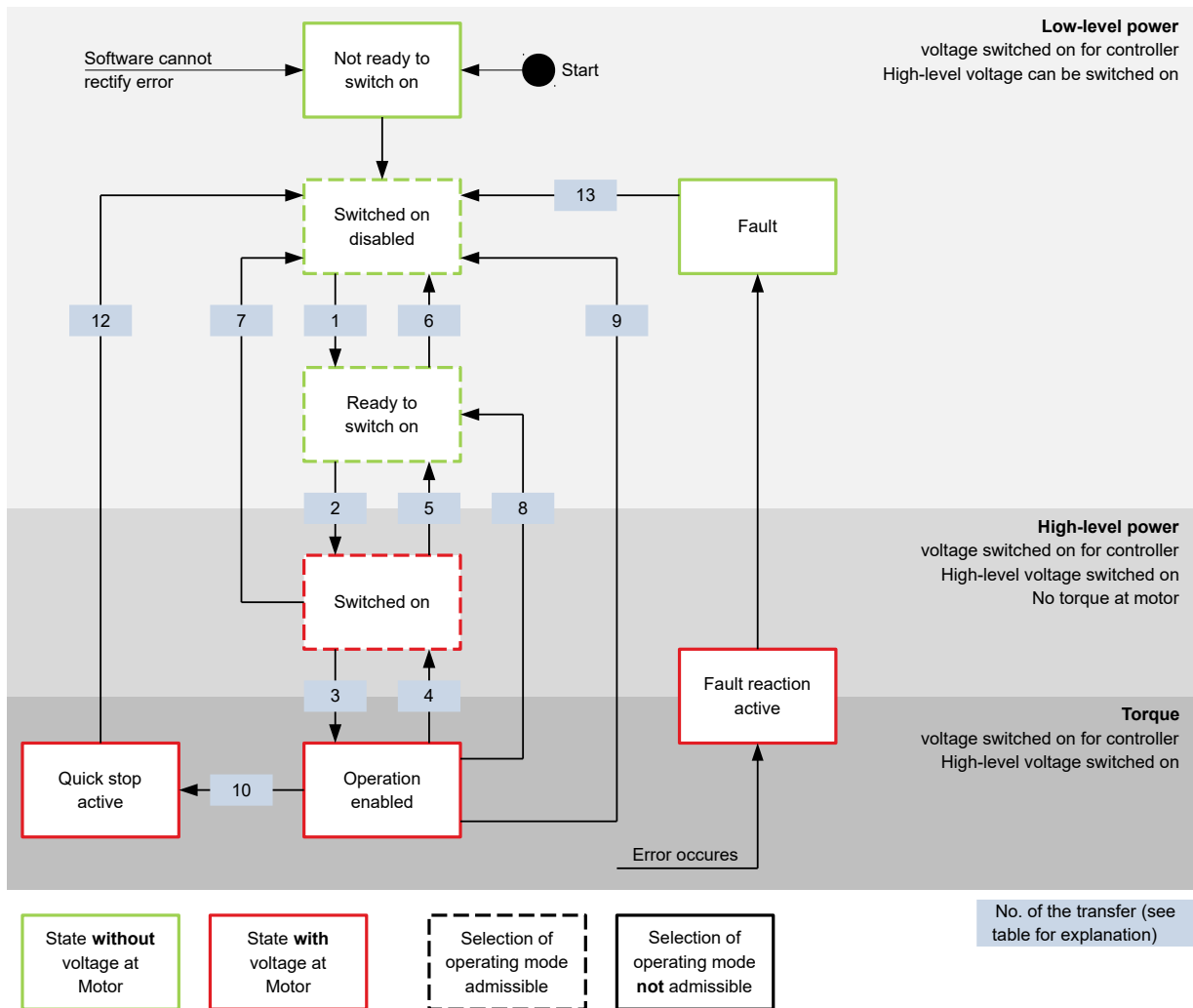
To switch the controller to the ready state, it is necessary to run through a *state machine*. This is defined in *CANopen standard 402*. State changes are requested in object **6040_h** (controlword). The actual state of the state machine can be found in object **6041_h** (statusword).

Controlword

State changes are requested via object **6040_h** (controlword).

State transitions

The diagram shows the possible state transitions.



Listed in the following table are the bit combinations for the controlword that result in the corresponding state transitions. An X here corresponds to a bit state that requires no further consideration. The only exception is the resetting of the error (fault reset): the transition is only requested by the rising edge of the bit.

| Command | Bit in object 6040 _h | | | | | Transition |
|-------------------|---------------------------------|-------|-------|-------|-------|-------------|
| | Bit 7 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
| Shutdown | 0 | X | 1 | 1 | 0 | 1, 5, 8 |
| Switch on | 0 | 0 | 1 | 1 | 1 | 2 |
| Disable voltage | 0 | X | X | 0 | X | 6, 7, 9, 12 |
| Quick stop | 0 | X | 0 | 1 | X | 10 |
| Disable operation | 0 | 0 | 1 | 1 | 1 | 4 |
| Enable operation | 0 | 1 | 1 | 1 | 1 | 3 |
| Fault reset | | X | X | X | X | 13 |

Statusword

Listed in the following table are the bit masks that break down the state of the controller.

| Statusword (6041 _h) | State |
|---------------------------------|------------------------|
| xxxx xxxx x0xx 0000 | Not ready to switch on |
| xxxx xxxx x1xx 0000 | Switch on disabled |
| xxxx xxxx x01x 0001 | Ready to switch on |
| xxxx xxxx x01x 0011 | Switched on |
| xxxx xxxx x01x 0111 | Operation enabled |
| xxxx xxxx x00x 0111 | Quick stop active |
| xxxx xxxx x0xx 1111 | Fault reaction active |
| xxxx xxxx x0xx 1000 | Fault |

After switching on and successfully completing the self-test, the controller reaches the *Switch on disabled* state.



Note

If an unrecoverable error occurs, the controller changes to the *Not ready to switch on* state and remains there.

Operating mode

The operating mode is set in object **6060_h**. The actually active operating mode is displayed in **6061_h**.

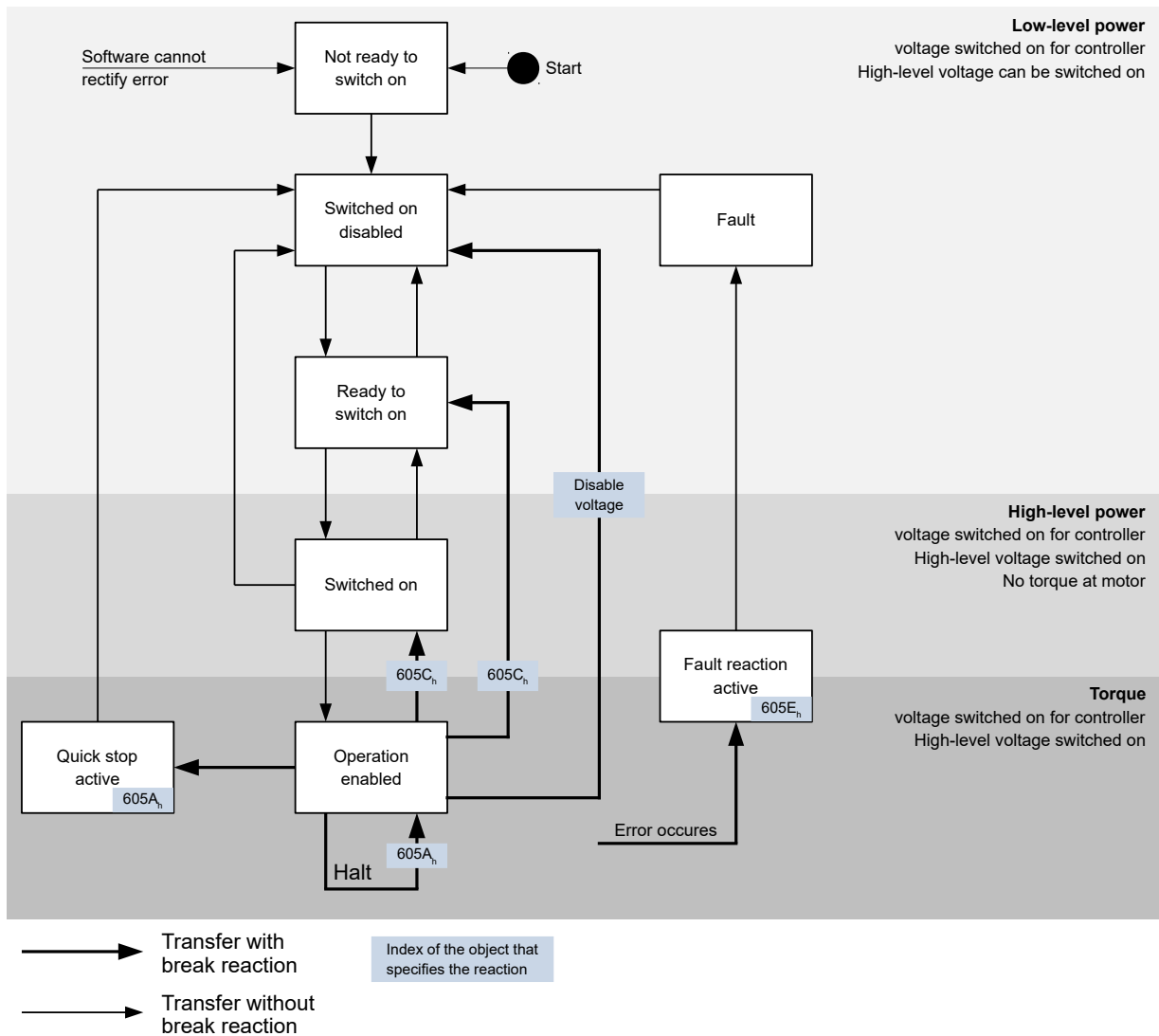
The operating mode can be set or changed at any time.

5.2.2 Behavior upon exiting the *Operation enabled* state

Halt motion reactions

Various halt motion reactions can be programmed upon exiting the *Operation enabled* state.

The following graphic shows an overview of the halt motion reactions.



Quick stop active

Transition to the *Quick stop active* state (quick stop option):

In this case, the action stored in object **605A_n** is executed (see following table).

| Value in object 605A _n | Description |
|-----------------------------------|--|
| -32768 ... -1 | Reserved |
| 0 | Immediate stop |
| 1 | Braking with <i>slow down ramp</i> (braking deceleration depending on operating mode) and subsequent state change to <i>Switch on disabled</i> |
| 2 | Braking with <i>quick stop ramp</i> and subsequent state change to <i>Switch on disabled</i> |
| 3 ... 32767 | Reserved |

Ready to switch on

Transition to the *Ready to switch on* state (shutdown option):

In this case, the action stored in object **605B_h** is executed (see following table).

| Value in object 605B _h | Description |
|-----------------------------------|--|
| -32768 ... -1 | Reserved |
| 0 | Immediate stop |
| 1 | Braking with <i>slow down ramp</i> (braking deceleration depending on operating mode) and subsequent state change to <i>Switch on disabled</i> |
| 2 ... 32767 | Reserved |

Switched on

Transition to the *Switched on* state (disable operation option):

In this case, the action stored in object **605C_h** is executed (see following table).

| Value in object 605C _h | Description |
|-----------------------------------|--|
| -32768 ... -1 | Reserved |
| 0 | Immediate stop |
| 1 | Braking with <i>slow down ramp</i> (braking deceleration depending on operating mode) and subsequent state change to <i>Switch on disabled</i> |
| 2 ... 32767 | Reserved |

Halt

The bit is valid in the following modes:

- **Profile Position**
- **Velocity**
- **Profile Velocity**
- **Profile Torque**
- **Interpolated Position Mode**

When setting bit 8 in object **6040_h** (controlword), the reaction stored in **605D_h** is executed (see following table):

| Value in object 605D _h | Description |
|-----------------------------------|--|
| -32768 ... 0 | Reserved |
| 1 | Braking with <i>slow down ramp</i> (braking deceleration depending on operating mode) |
| 2 | Braking with <i>quick stop ramp</i> (braking deceleration depending on operating mode) |
| 3 ... 32767 | Reserved |

Fault

Case of an error (fault):

If an error occurs, the motor will brake according to the value stored in object **605E_h**.

| Value in object 605E _h | Description |
|-----------------------------------|--|
| -32768 ... -1 | Reserved |
| 0 | Immediate stop |
| 1 | Braking with <i>slow down ramp</i> (braking deceleration depending on operating mode) |
| 2 | Braking with <i>quick stop ramp</i> (braking deceleration depending on operating mode) |
| 3 ... 32767 | Reserved |

Following/slippage error

If a following or slippage error occurs, the motor is braked according to the value stored in object **3700_h**.

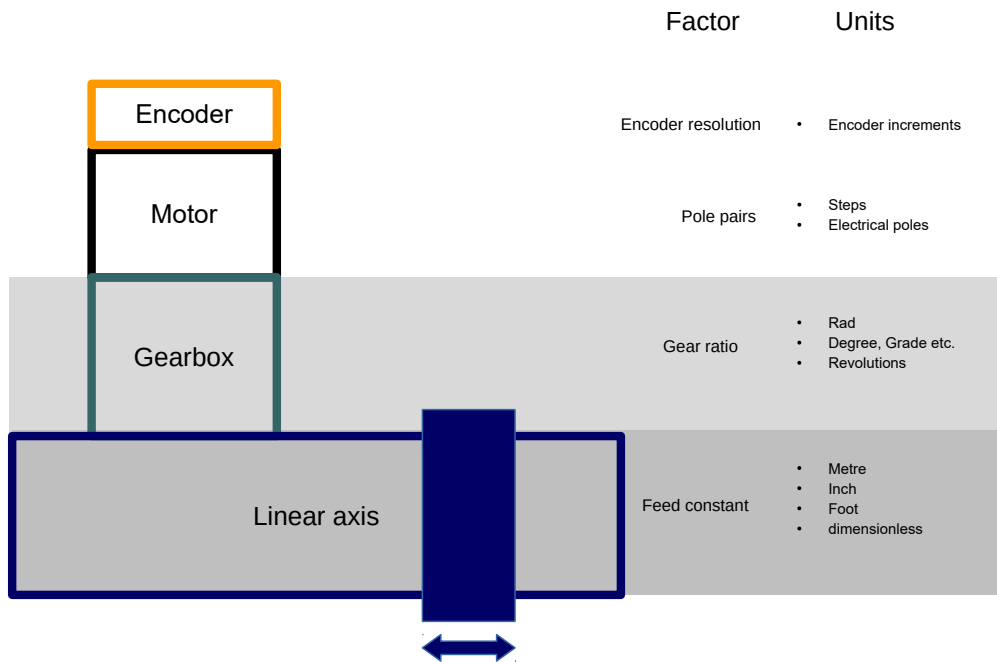
| Value | Description |
|---------------|--|
| -32768 ... -1 | Reserved |
| 0 | Immediate stop |
| 1 | Braking with <i>slow down ramp</i> (braking deceleration depending on operating mode) |
| 2 | Braking with <i>quick stop ramp</i> (braking deceleration depending on operating mode) |
| 3 ... 32767 | Reserved |

You can deactivate error monitoring by setting object **6065_h** to the value "-1" (FFFFFFFF_h) or object **60F8_h** to the value "7FFFFFFF_h".

5.3 User-defined units

The controller offers you the possibility to set user-defined units. It is thereby possible to set and read out the corresponding parameters, e.g., directly in degrees [°], [mm], etc.

Depending on the mechanical circumstances, you can also define a **Gear ratio** and/or a **Feed constant**.



Note

Value changes of all objects that are described in this chapter are not immediately applied in the *Operation enabled* state of the **CiA 402 Power State Machine**. For this to happen, the *Operation enabled* state must be exited.

5.3.1 Units

Units of the international unit system (*SI*) as well as a number of specific units are supported. It is also possible to specify a power of ten as a factor.

Listed in the following table are all supported units for the position and their values for **60A8_h** (**Position unit**) or **60A9_h** (**Speed unit**). Depending on the unit that is used, **Feed constant (6092_h)** and/or **Gear ratio (6091_h)** are/is taken into account.

| Name | Unit symbol | Value | 6091 _h | 6092 _h | Description |
|-----------------------|-------------|-----------------|-------------------|-------------------|---|
| meter | m | 01 _h | yes | yes | <i>Meter</i> |
| inch | in | C1 _h | yes | yes | <i>Inch</i> (=0.0254 m) |
| foot | ft | C2 _h | yes | yes | <i>Foot</i> (=0.3048 m) |
| grade | g | 40 _h | yes | no | <i>Gradian</i> (unit of angle, 400 corresponds to 360°) |
| radian | rad | 10 _h | yes | no | <i>Radian</i> |
| degree | ° | 41 _h | yes | no | <i>Degrees</i> |
| arcminute | ' | 42 _h | yes | no | <i>Arcminute</i> (60'=1°) |
| arcsecond | " | 43 _h | yes | no | <i>Arcsecond</i> (60"=1') |
| mechanical revolution | | B4 _h | yes | no | <i>Revolution</i> |

| Name | Unit symbol | Value | 6091 _h | 6092 _h | Description |
|-------------------|-------------|-----------------|-------------------|-------------------|--|
| encoder increment | | B5 _h | no | no | <i>Encoder increments</i> |
| step | | AC _h | no | no | <i>Steps.</i> With 2-phase stepper motors, the number of pole pairs (2030_h) multiplied by 4 is equivalent to one revolution. With 3-phase BLDC motors, the number of pole pairs (2030_h) multiplied by 6 is equivalent to one revolution. |
| electrical pole | | C0 _h | no | no | <i>Electric poles.</i> With a stepper motor that has, e.g., 50 pole pairs (2030_h), the unit corresponds to 1/50 of a revolution. |
| dimensionless | | 00 _h | yes | yes | <i>Dimensionless length unit</i> |

Listed in the following table are all supported units for the time and their values for **60A9_h** (**Speed unit**):

| Name | Unit symbol | Value | Description |
|--------|-------------|-----------------|----------------------------|
| second | s | 03 _h | <i>Second</i> |
| minute | min | 47 _h | <i>Minute</i> |
| hour | h | 48 _h | <i>Hour</i> |
| day | d | 49 _h | <i>Day</i> |
| year | a | 4A _h | <i>Year (=365.25 days)</i> |

Listed in the following table are the possible exponents and their values for **60A8_h** (**Position unit**) and **60A9_h** (**Speed unit**):

| Factor | Exponent | Value |
|------------------|----------|-----------------|
| 10 ⁶ | 6 | 06 _h |
| 10 ⁵ | 5 | 05 _h |
| ... | ... | ... |
| 10 ¹ | 1 | 01 _h |
| 10 ⁰ | 0 | 00 _h |
| 10 ⁻¹ | -1 | FF _h |
| ... | .. | ... |
| 10 ⁻⁵ | -5 | FB _h |
| 10 ⁻⁶ | -6 | FA _h |

5.3.2 Encoder resolution

The physical resolution of the used encoder/sensor is calculated from the encoder increments (**608F_h:1_h** (Encoder Increments)) per motor revolutions (**608F_h:2_h** (Motor Revolutions)):

$$\text{Position encoder resolution} = \frac{\text{Encoder Increments (608F}_{h}:01)}{\text{Motor Revolutions (608F}_{h}:02)}$$

5.3.3 Gear ratio

The gear ratio is calculated from motor revolutions (**6091_h:1** (Motor Revolutions)) per axis rotation (**6091_h:2** (Shaft Revolutions)) as follows:

$$\text{Gear ratio} = \frac{\text{Motor revolution (6091}_{h:1})}{\text{Shaft revolution (6091}_{h:2})}$$

5.3.4 Feed constant

The feed constant is calculated from the feed (**6092_h:1** (Feed)) per revolution of the output shaft (**6092_h:2** (Shaft Revolutions)) as follows:

$$\text{Feed Constant} = \frac{\text{Feed (6092}_{h:1})}{\text{Shaft Revolutions (6092}_{h:2})}$$

The feed constant is useful for specifying the lead screw pitch for a linear axis and is used if the unit is based on length dimensions or if it is dimensionless.

5.3.5 Calculation formulas for user units

Position unit

Object **60A8_h** contains:

- Bits 16 to 23: The position unit (see chapter **Units**)
- Bits 24 to 31: The exponent of a power of ten (see chapter **Units**)

| | | | | | | | | | | | | | | | |
|----------------|----|----|----|----|----|----|----|----------------|----|----|----|----|----|----|----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Factor | | | | | | | | Unit | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| reserved (00h) | | | | | | | | reserved (00h) | | | | | | | |

Example

If **60A8_h** is written with the value "FF410000_h" (bits 16-23=41_h and bits 24-31=FF_h), the unit is set to *tenths of degree* (factory setting).

With a relative target position (**607A_h**) of 3600, the motor moves exactly one mechanical revolution, if **Gear ratio** is 1:1. The **Feed constant** plays no role in this case.

Example

If **60A8_h** is written with the value "FD010000_h" (bits 16-23=01_h and bits 24-31=FD_h(=-3)), the unit is set to *millimeter*.

With a relative target position (**607A_h**) of 1, the motor moves exactly one mechanical revolution, if **Feed constant** and **Gear ratio** are 1:1.

If the **Feed constant** is set according to the lead screw pitch of a linear axis, the motor turns far enough that a feed of 1 mm is achieved.

Speed unit

Object **60A9_h** contains:

- Bits 8 to 15: The time unit (see chapter **Units**)
- Bits 16 to 23: The position unit (see chapter **Units**)
- Bits 24 to 31: The exponent of a power of ten (see chapter **Units**)

| | | | | | | | | | | | | | | | |
|--------------------|----|----|----|----|----|----|----|----------------------|----|----|----|----|----|----|----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Factor | | | | | | | | Nominator (Position) | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Denominator (Time) | | | | | | | | reserved (00h) | | | | | | | |

Example

If **60A9_h** is written with the value "00B44700_h" (bits 8-15=00_h, bits 16-23=B4_h and bits 24-31=47_h), the unit is set to *revolutions per minute* (factory setting).

Example

If **60A9_h** is written with the value "FD010300_h" (bits 8-15=FD_h(=-3), bits 16-23=01_h and bits 24-31=03_h), the unit is set to *millimeters per second*.



Note

The speed unit in **Velocity** mode is preset to *revolutions per minute*. You can only set the unit via the **604Ch VI Dimension Factor**.

Conversion factor for the speed unit

You can set an additional factor for the speed unit. Thus, a unit of, e.g., 1/3 revolutions/minute is possible. The factor n is calculated from the factor for numerator (**6096_h:01_h**) divided by the factor for denominator (**6096_h:02_h**).

$$n_{\text{velocity}} = \frac{6096_{\text{h}}:01}{6096_{\text{h}}:02}$$

Acceleration unit

The acceleration unit is **speed unit** per second.

Conversion factor for the acceleration unit

The factor n for the acceleration unit is calculated from the numerator (**6097_h:01_h**) divided by the denominator (**6097_h:02_h**).

$$n_{\text{acceleration}} = \frac{6097_{\text{h}}:01}{6097_{\text{h}}:02}$$

Jerk unit

The jerk unit is **Acceleration unit** per second.

Conversion factor for jerk

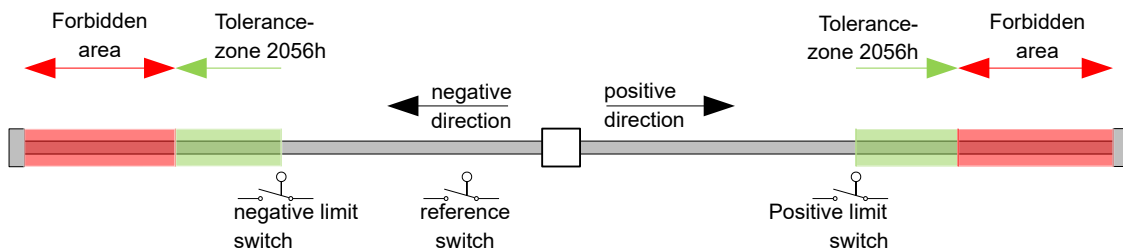
The factor n for the jerk is calculated from the numerator (**60A2_h:01_n**) divided by the denominator (**60A2_h:02_n**).

$$n_{\text{jerk}} = \frac{60A2_h:01}{60A2_h:02}$$

5.4 Limitation of the range of motion

The digital inputs can be used as limit switches, as is described in chapter **Digital inputs**, if you activate this function for the inputs. The controller also supports software limit switches.

5.4.1 Tolerance bands of the limit switches



The previous figure shows the breakdown of the tolerance bands next to the limit switches:

- The tolerance zone begins immediately after the limit switch. Free movement is possible in this zone. The length of the zone can be set in object **2056_h**.
- If the motor moves into the forbidden range, the controller triggers an immediate stop and it switches to the *fault* state, see also **State transitions**.

5.4.2 Software limit switches

The controller takes into account software limit switches (**607D_h** (Software Position Limit)). Target positions (**607A_h**) are limited by **607D_h**; the demand position (**6062_h**) may not be larger than the limits in **607D_h**. If the motor is located outside of the permissible range when setting up the limit switches, only travel commands in the direction of the permissible range are accepted.

5.5 Cycle times

The controller operates with a cycle time of 1 ms. This means that data are processed every 1 ms; multiple changes to a value (e.g., value of an object or level at a digital input) within one ms cannot be detected.

The following table includes an overview of the cycle times of the various processes.

| Task | Cycle time |
|--------------------|-------------------|
| Application | 1 ms |
| NanoJ application | 1 ms |
| Current controller | 31.25 μs (32 kHz) |

| Task | Cycle time |
|---------------------|---------------------|
| Speed controller | 250 μ s (4 kHz) |
| Position controller | 1 ms |

6 Operating modes

6.1 Profile Position

6.1.1 Overview

Description

Profile Position Mode is used to move to positions relative to the last target position or to an absolute position (last reference position). During the movement, the limit values for the speed, starting acceleration/braking deceleration and jerks are taken into account.



Note

The limit switches and, thus, the tolerance bands are active in this mode. For further information on the limit switches, see **Limitation of the range of motion**.

Activation

To activate the mode, the value "1" must be set in object **6060_h** (Modes Of Operation) (see "**CiA 402 Power State Machine**").

Controlword

The following bits in object **6040_h** (controlword) have a special function:

- Bit 4 starts a travel command. This is carried out on a transition from "0" to "1". An exception occurs if changing from another operating mode to *Profile Position*: If bit 4 is already set, it does not need to be set to "0" and then back to "1" in order to start the travel command.
- Bit 5: If this bit is set to "1", a travel command triggered by bit 4 is immediately executed. If it is set to "0", the just executed travel command is completed and only then is the next travel command started.
- Bit 6: With "0", the target position (**607A_h**) is absolute and with "1" the target position is relative. The reference position is dependent on bits 0 and 1 of object **60F2_h**.
- Bit 8 (Halt): If this bit is set to "1", the motor stops. On a transition from "1" to "0", the motor accelerates with the set start ramp to the target speed. On a transition from "0" to "1", the motor brakes and comes to a standstill. The braking deceleration is dependent here on the setting of the "Halt Option Code" in object **605D_h**.
- Bit 9 (Change on setpoint): If this bit is set, the speed is not changed until the first target position is reached. This means that, before the first target is reached, no braking is performed, as the motor should not come to a standstill at this position.

| Controlword 6040 _h | | |
|-------------------------------|-------|--|
| Bit 9 | Bit 5 | Definition |
| X | 1 | The new target position is moved to immediately. |
| 0 | 0 | Positioning is completed before moving to the next target position with the new limits. |
| 1 | 0 | The current target position is only passed through; afterwards, the new target position is moved to with the new values. |

For further information, see figure in "**Setting travel commands**".



Note

Bit 9 in the controlword is ignored if the ramp speed is not met at the target point. In this case, the controller would need to reset and take a run-up to reach the preset.

Statusword

The following bits in object **6041_h** (statusword) have a special function:

- Bit 10 (Target Reached): This bit is set to "1" if the last target was reached and the motor remains within a tolerance window (**6067_h**) for a preset time (**6068_h**).
- Bit 12 (Set-point acknowledge): This bit confirms receipt of a new and valid set point. It is set and reset in sync with the "New set-point" bit in the controlword.

There is an exception in the event that a new movement is started before another one has completed and the next movement is not to occur until after the first one has finished. In this case, the bit is reset if the command was accepted and the controller is ready to execute new travel commands. If a new travel command is sent even though this bit is still set, the newest travel command is ignored.

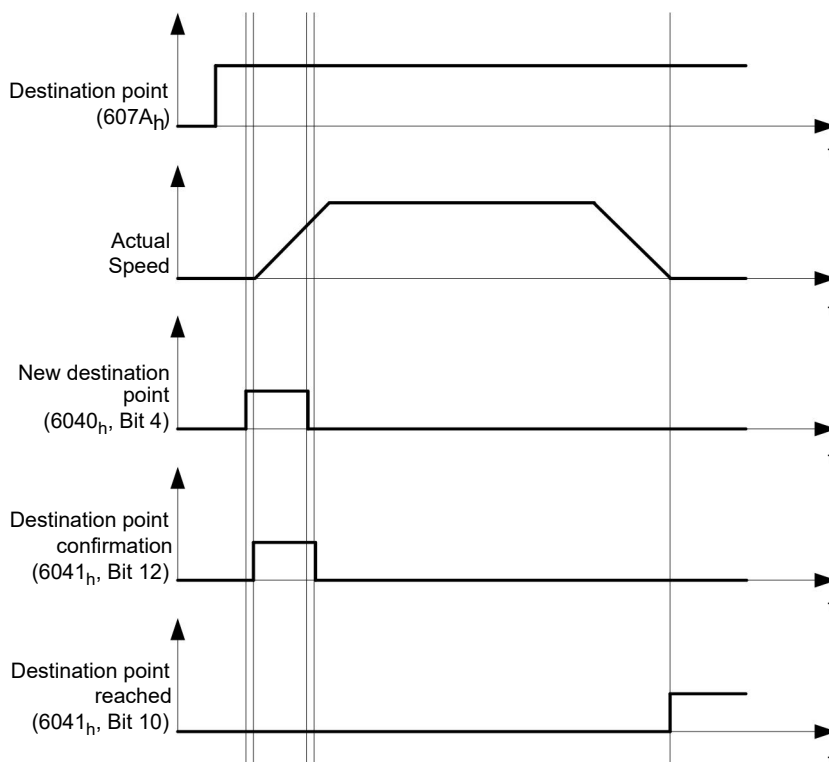
The bit is not set if one of the following conditions is met:

- The new target position can no longer be reached while adhering to all boundary conditions.
- A target position was already traveled to and a target position was already specified. A new target position can only be specified after the current positioning has been concluded.
- Bit 13 (Following Error): This bit is set in *closed loop* mode if the following error is greater than the set limits (**6065_h** (Following Error Window) and **6066_h** (Following Error Time Out)).

6.1.2 Setting travel commands

Travel command

In object **607A_h** (Target Position), the new target position is specified in user units (see **User-defined units**). The travel command is then triggered by setting bit 4 in object **6040_h** (controlword). If the target position is valid, the controller responds with bit 12 in object **6041_h** (statusword) and begins the positioning move. As soon as the position is reached, bit 10 in the statusword is set to "1".



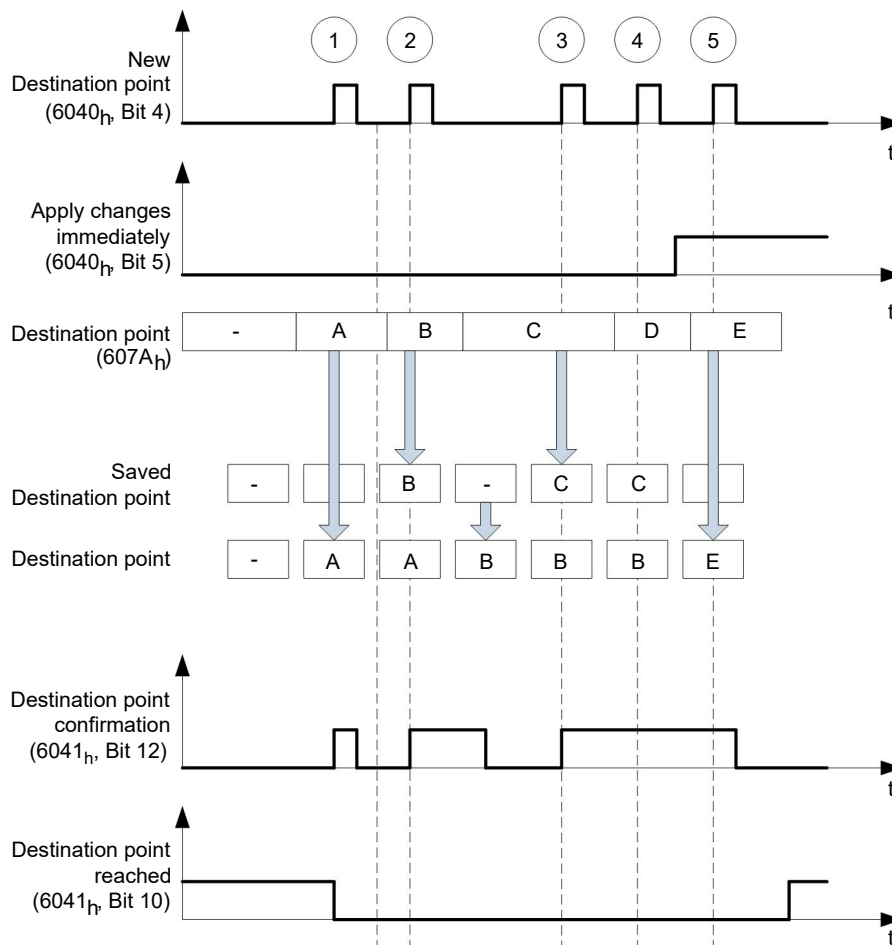
The controller can also reset bit 4 in object **6040_h** (controlword) on its own. This is set with bits 4 and 5 of object **60F2_h**.

Other travel commands

Bit 12 in object **6041_h** (statusword, set-point acknowledge) changes to "0" if another travel command can be buffered (see time 1 in the following figure). As long as a target position is being moved to, a second target position can be passed to the controller in preparation. All parameters – such as speed, acceleration, braking deceleration, etc. – can thereby be reset (time 2). If the buffer is empty, the next time can be queued up (time 3).

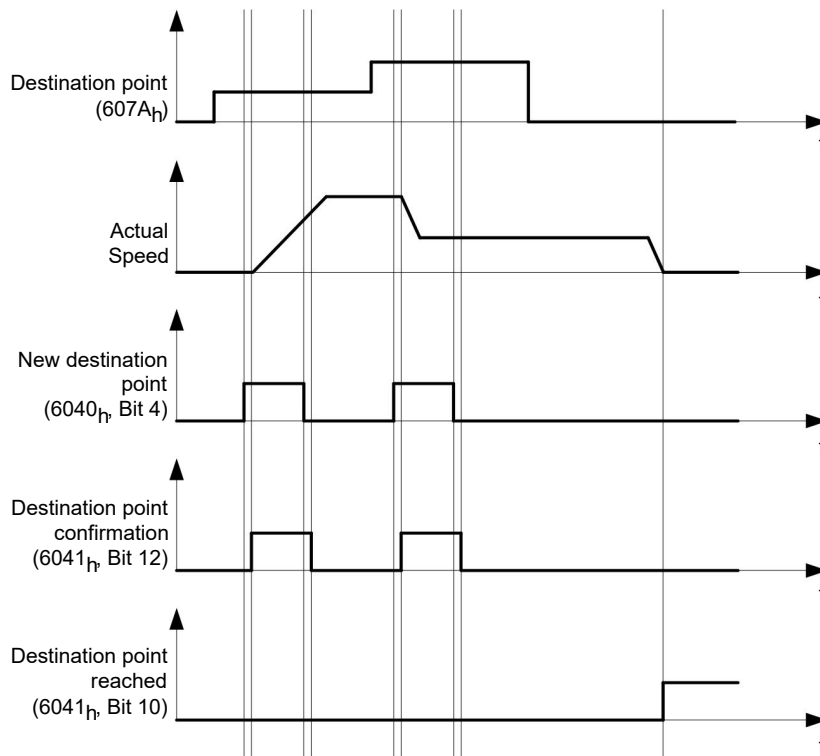
If the buffer is already full, a new set point is ignored (time 4). If bit 5 in object **6040_h** (controlword, bit: "Change Set-Point Immediately") is set, the controller operates without the buffer; new travel commands are implemented directly (time 5).

Times



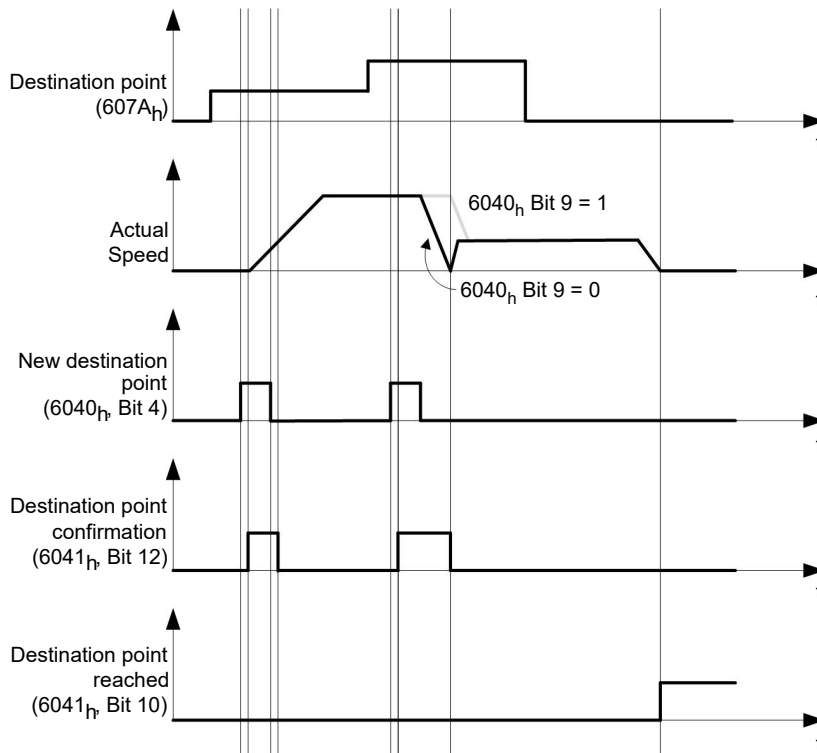
Transition procedure for second target position

The following graphic shows the transition procedure for the second target position while moving to the first target position. In this figure, bit 5 of object **6040_h** (controlword) is set to "1"; the new target value is, thus, taken over immediately.



Possibilities for moving to a target position

If bit 9 in object **6040_h** (controlword) is equal to "0", the current target position is first moved to completely. In this example, the final speed (**6082_h**) of the target position is equal to zero. If bit 9 is set to "1", the profile speed (**6081_h**) is maintained until the target position is reached; only then do the new boundary conditions apply.



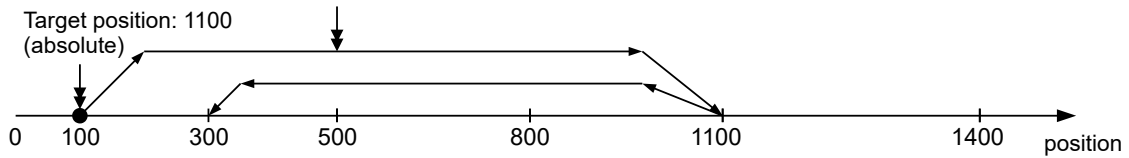
Possible combinations of travel commands

To provide a better overview of the travel commands, combinations of travel commands are listed and depicted in this chapter.

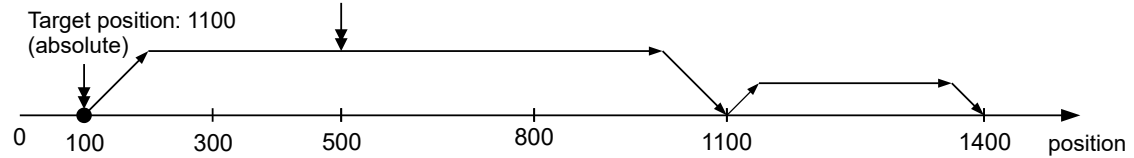
The following applies for the figures below:

- A double arrow indicates a new travel command.
- The first travel command at the start is always an absolute travel command to position 1100.
- The second movement is performed at a lower speed so as to present the graphs in a clear manner.

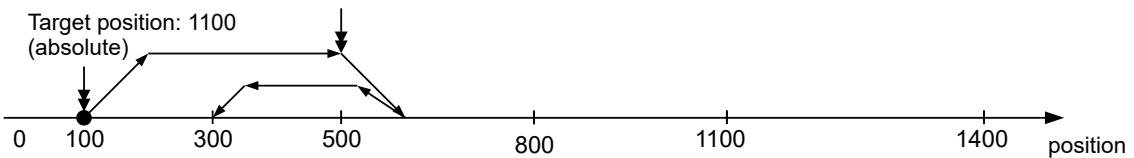
- Change on setpoint ($6040_h:00$ Bit 5 = 0)
- Move absolute ($6040_h:00$ Bit 6 = 0)
- Target position: 300



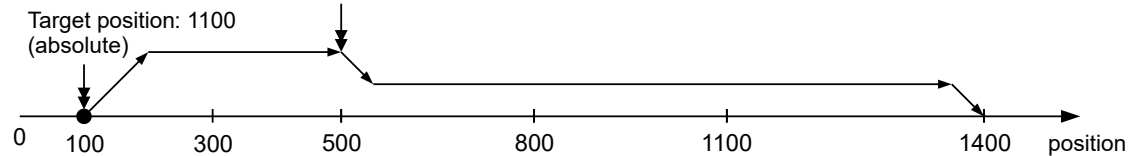
- Relative to the preceding target position ($60F2_h:00 = 0$)
- Change on setpoint ($6040_h:00$ Bit 5 = 0)
- Move relative ($6040_h:00$ Bit 6 = 1)
- Target position: 300

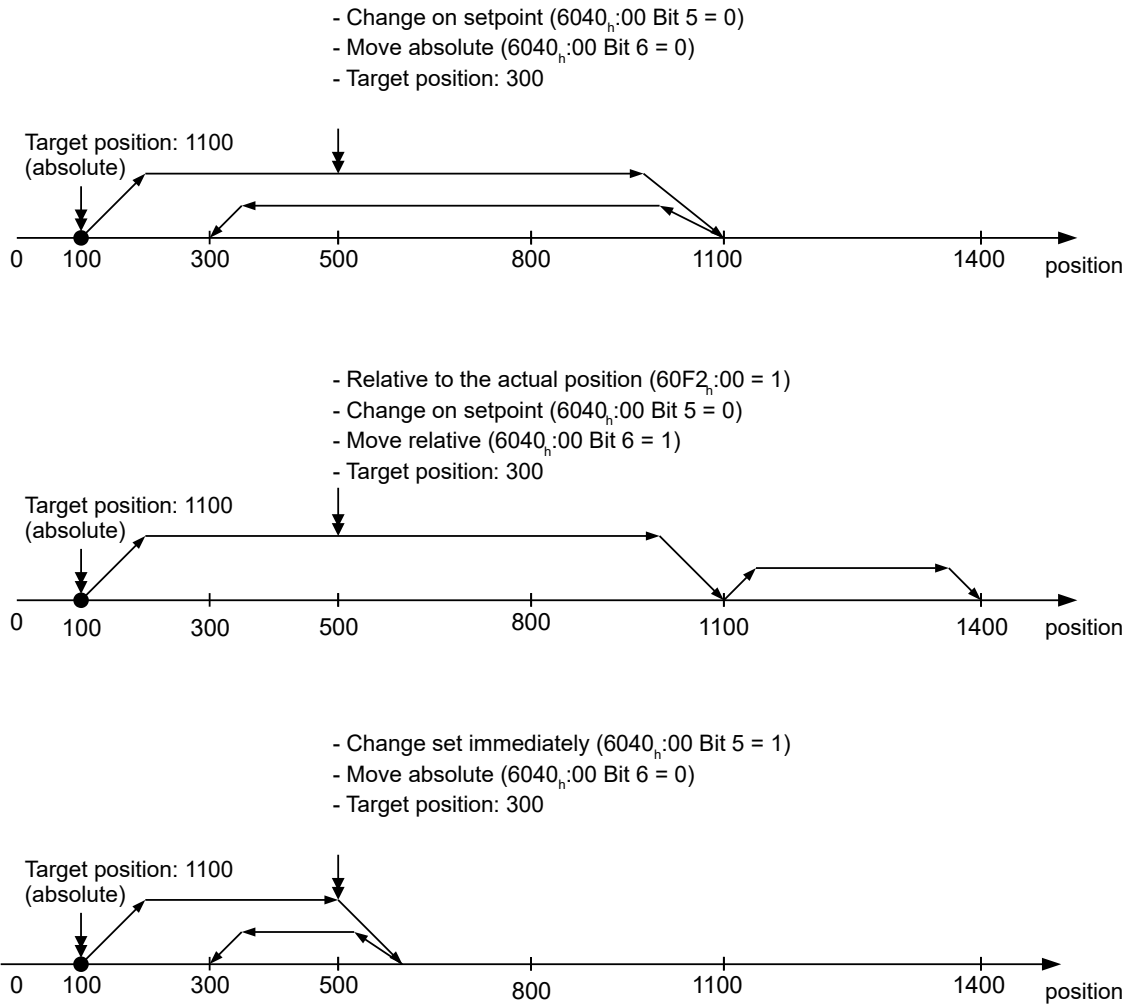


- Change set immediately ($6040_h:00$ Bit 5 = 1)
- Move absolute ($6040_h:00$ Bit 6 = 0)
- Target position: 300



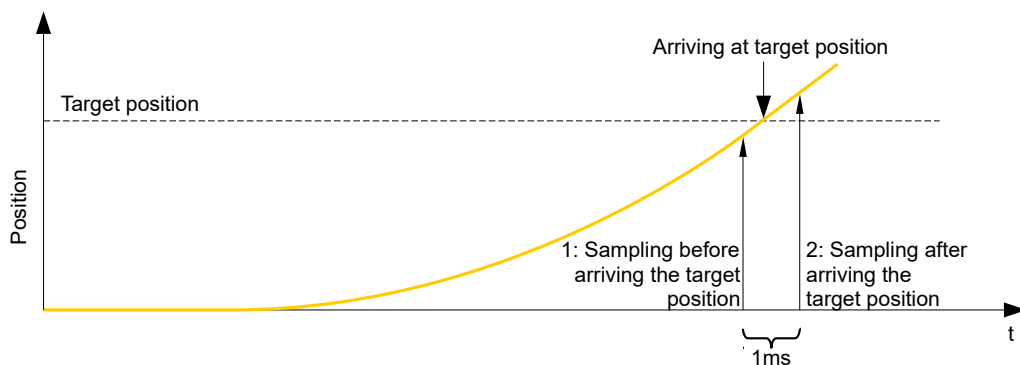
- Relative to the preceding target position ($60F2_h:00 = 0$)
- Change set immediately ($6040_h:00$ Bit 5 = 1)
- Move relative ($6040_h:00$ Bit 6 = 1)
- Target position: 300





6.1.3 Loss of accuracy for relative movements

When linking together relative movements, a loss of accuracy may occur if the final speed is not set to zero. The following graphic illustrates the reason.



The current position is sampled once per millisecond. It is possible that the target position is reached between two samples. If the final speed is not equal to zero, then, after the target position is reached, the sample is used as an offset as the basis for the subsequent movement. As a result, the subsequent movement may go somewhat farther than expected.

6.1.4 Boundary conditions for a positioning move

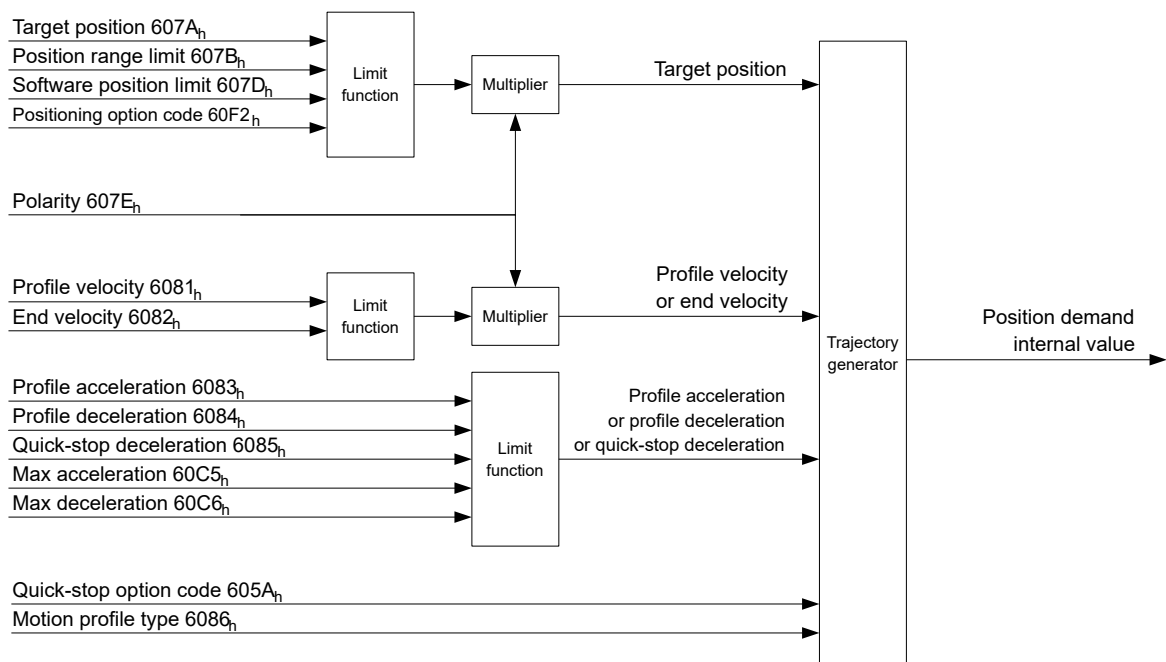
Object entries

The boundary conditions for the position that has been moved to can be set in the following entries of the object dictionary:

- **607A_h**: (Target Position): Planned target position
- **607D_h**: (Software Position Limit): Definition of the limit stops (see chapter **Software limit switches**)
- **607C_h**: (Home Offset): Specifies the difference between the zero position of the controller and the reference point of the machine in **user-defined units**. (See "**Homing**")
- **607B_h**: (Position Range Limit): Limits of a modulo operation for replicating an endless rotation axis
- **607_h**: (Polarity): Direction of rotation
- **6081_h**: (Profile Velocity): Maximum speed with which the position is to be approached
- **6082_h**: (End Velocity): Speed upon reaching the target position
- **6083_h**: (Profile Acceleration): Desired starting acceleration
- **6084_h**: (Profile Deceleration): Desired braking deceleration
- **6085_h**: (Quick Stop Deceleration): Emergency-stop braking deceleration in case of the "Quick stop active" state of the "CiA 402 Power State Machine"
- **6086_h**: (Motion Profile Type): Type of ramp to be traveled; if the value is "0", the jerk is not limited; if the value is "3", the values of 60A4_h:1_h–4_h are set as limits for the jerk.
- **60C5_h**: (Max Acceleration): The maximum acceleration that may not be exceeded when moving to the end position
- **60C6_h**: (Max Deceleration): The maximum braking deceleration that may not be exceeded when moving to the end position
- **60A4_h**: (Profile Jerk), subindex 01_h to 04_h: Objects for specifying the limit values for the jerk.
- The speed is limited by **607F_h** (Max Profile Velocity) and **6080_h** (Max Motor Speed); the smaller value is used as the limit.
- **60F2_h**: (Positioning Option Code): Defines the positioning behavior

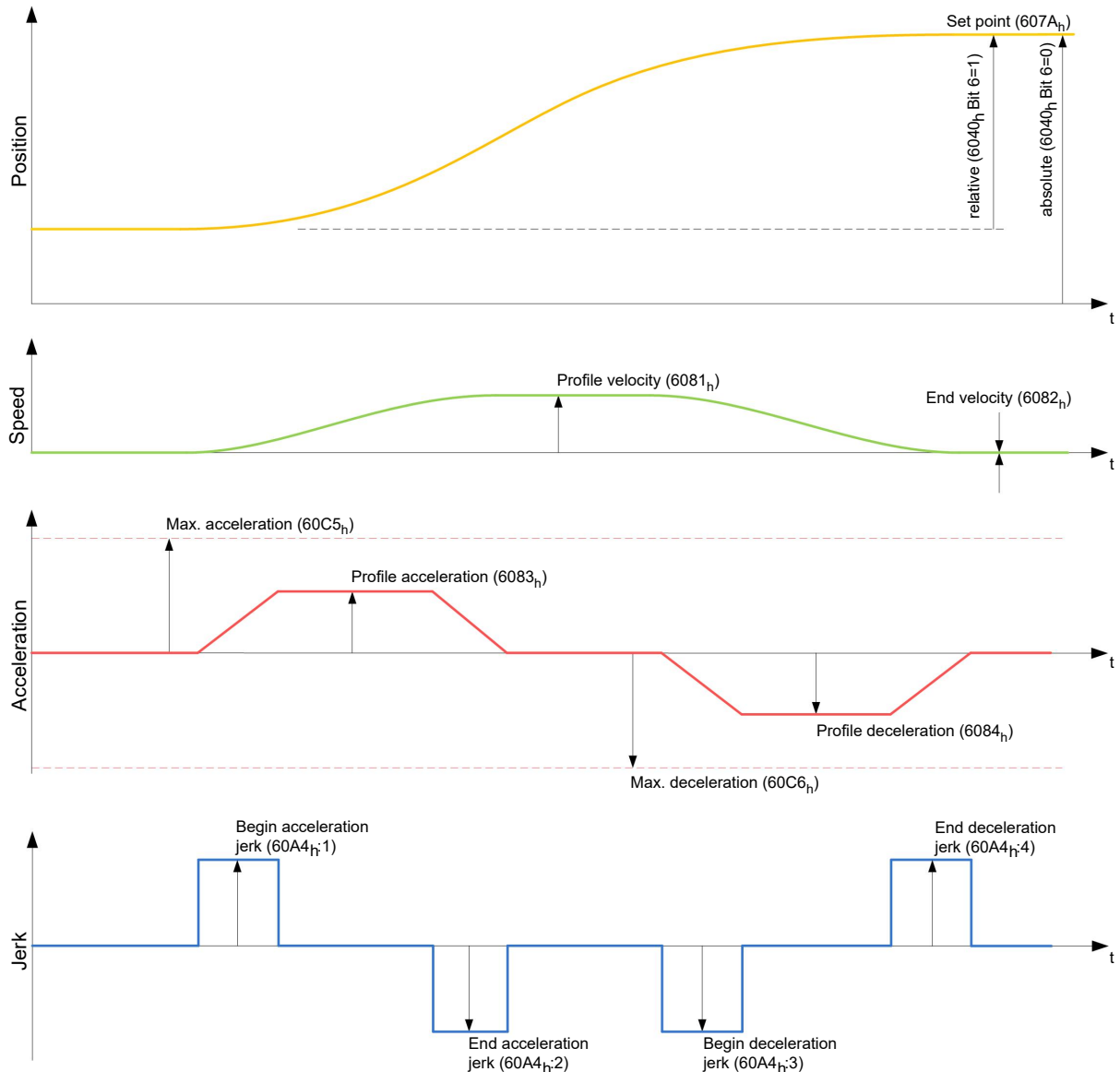
Objects for the positioning move

The following graphic shows the objects involved in the boundary conditions of the positioning move.



Parameters for the target position

The following graphic shows an overview of the parameters that are used for moving to a target position (figure not to scale).



6.1.5 Jerk-limited mode and non-jerk-limited mode

Description

A distinction is made between the "jerk-limited" and "non-jerk-limited" modes.

Jerk-limited mode

Jerk-limited positioning can be achieved by setting object **6086_h** to "3". The entries for the jerks in subindices :1_h–4_h of object **60A4** thereby become valid.

Non-jerk-limited mode

A "non-jerk-limited" ramp is traveled if the entry in object **6086_h** is set to "0" (default setting).

6.2 Velocity

6.2.1 Description

This mode operates the motor at a preset target speed, similar to a frequency inverter. Unlike the *Profile Velocity Mode*, this mode does not permit the selection of jerk-limited ramps.



Note

The limit switches and, thus, the tolerance bands are active in this mode. For further information on the limit switches, see **Limitation of the range of motion**.

6.2.2 Activation

To activate the mode, the value "2" must be set in object **6060_h** (Modes Of Operation) (see "**CiA 402 Power State Machine**").

6.2.3 Controlword

The following bits in object **6040_h** (controlword) have a special function:

- Bit 8 (Halt): If this bit is set to "1", the motor stops. On a transition from "1" to "0", the motor accelerates with the acceleration ramp to the target speed. On a transition from "0" to "1", the motor brakes according to the deceleration ramp and comes to a standstill.

6.2.4 Statusword

The following bits in object **6041_h** (statusword) have a special function:

- Bit 11: Limit exceeded: The target speed is above or below the set limit values.

6.2.5 Object entries

The following objects are necessary for controlling this mode:

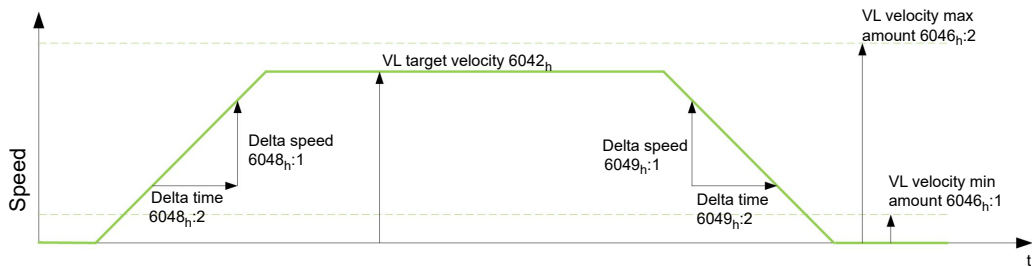
- **604C_h** (Dimension Factor):
The unit for speed values is defined here for the following objects.
Subindex 1 contains the denominator (multiplier) and subindex 2 contains the numerator (divisor) with which the internal speed values are converted to revolutions per minute. If, for example, subindex 1 is set to the value "60" and subindex 2 is set to the value "1", the speed is specified in revolutions per second (60 revolutions per 1 minute).
- **6042_h**: Target Velocity.
The target speed is set here in user-defined units.
- **6048_h**: Velocity Acceleration
This object defines the acceleration. Subindex 1 contains the change in speed, subindex 2 the corresponding time in seconds. Both together are used to calculate the acceleration:
$$\text{VL velocity acceleration} = \frac{\text{Delta speed (6048}_{h}:1)}{\text{Delta time (6048}_{h}:2)}$$
- **6049_h** (Velocity Deceleration):
This object defines the deceleration (deceleration ramp). The subindices here are arranged as described in object **6048_h**; the change in speed is to be specified with positive sign.
- **6046_h** (Velocity Min Max Amount):
The limitations of the target speeds are specified in this object.
The minimum speed is set in **6046_h:1_h**. If the target speed (**6042_h**) falls below the minimum speed, the value is limited to the minimum speed **6046_h:1_h**.
The maximum speed is set in **6046_h:2_h**. If the target speed (**6042_h**) exceeds the maximum speed, the value is limited to the maximum speed **6046_h:2_h**.

- **604A_h** (Velocity Quick Stop):
This object can be used to set the quick-stop ramp. Subindices 1 and 2 are identical to those described for object **6048_h**.
- **6080_h** (Max Motor Speed): Maximum speed

The following objects can be used to check the function:

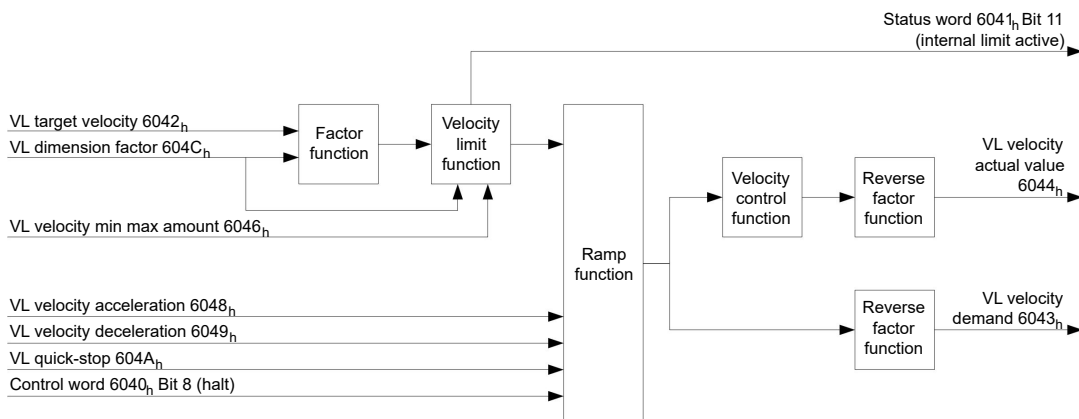
- **6043_h** (VI Velocity Demand)
- **6044_h** (VI Velocity Actual Value)

Speeds in Velocity Mode



Objects for Velocity Mode

The ramp generator follows the target speed, remaining within the set speed and acceleration limits. As long as a limit is active, bit 11 in object **6041_h** is set (internal limit active).



6.3 Profile Velocity

6.3.1 Description

This mode operates the motor in Velocity Mode with extended (jerk-limited) ramps. Unlike *Velocity Mode* (see "**Velocity**"), the **statusword** is used in this mode to indicate whether the target speed is reached.



Note

The limit switches and, thus, the tolerance bands are active in this mode. For further information on the limit switches, see **Limitation of the range of motion**.

6.3.2 Activation

To activate the mode, the value "3" must be set in object **6060_h** (Modes Of Operation) (see "**CiA 402 Power State Machine**").

6.3.3 Controlword

The following bits in object **6040_h** (controlword) have a special function:

- Bit 8 (Halt): If this bit is set to "1", the motor stops. On a transition from "1" to "0", the motor accelerates with the set start ramp to the target speed. On a transition from "0" to "1", the motor brakes and comes to a standstill.

6.3.4 Statusword

The following bits in object **6041_h** (statusword) have a special function:

- Bit 10 (target speed reached; Target Reached): In combination with bit 8 in the controlword, this bit specifies whether the target speed is reached, if braking is taking place or if the motor is at a standstill (see table).

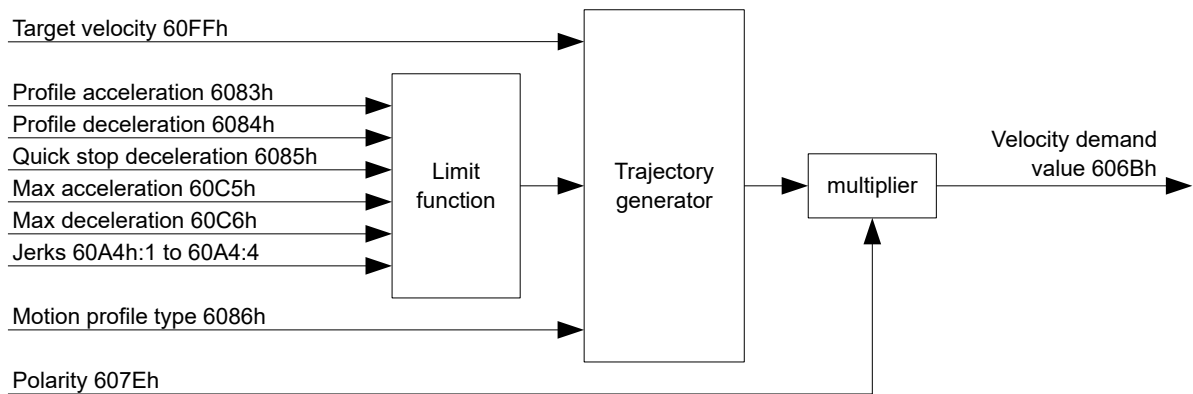
| 6041_h Bit 10 | 6040_h Bit 8 | Description |
|------------------------------------|-----------------------------------|---|
| 0 | 0 | Target speed not reached |
| 0 | 1 | Axis braking |
| 1 | 0 | Target speed within target window (defined in 606D_h and 606E_h) |
| 1 | 1 | Axis speed is 0 |

6.3.5 Object entries

The following objects are necessary for controlling this mode:

- **606B_h** (Velocity Demand Value):
This object contains the output of the ramp generator, which simultaneously serves as the preset value for the speed controller.
- **606C_h** (Velocity Actual Value):
Indicates the current actual speed.
- **606D_h** (Velocity Window):
This value specifies by how much the actual speed may vary from the set speed for bit 10 (target speed reached; Target Reached) in object **6041_h** (statusword) to be set to "1".
- **606E_h** (Velocity Window Time):
This object specifies how long the actual speed and the set speed must be close to one another (see **606D_h** "Velocity Window") for bit 10 "Target speed reached" in object **6041_h** (statusword) to be set to "1".
- **607E_h** (Polarity):
If bit 6 is set to "1" here, the sign of the target speed is reversed.
- **6083_h** (Profile acceleration):
Sets the value for the acceleration ramp in Velocity Mode.
- **6084_h** (Profile Deceleration):
Sets the value for the deceleration ramp in Velocity Mode.
- **6085_h** (Quick Stop Deceleration):
Sets the value for the deceleration ramp for rapid braking in Velocity Mode.
- **6086_h** (Motion Profile Type):
The ramp type can be selected here ("0" = trapezoidal ramp, "3" = jerk-limited ramp).
- **60FF_h** (Target Velocity):
Specifies the target speed that is to be reached.
- The speed is limited by **607F_h** (Max Profile Velocity) and **6080_h** (Max Motor Speed); the smaller value is used as the limit.

Objects in Profile Velocity Mode

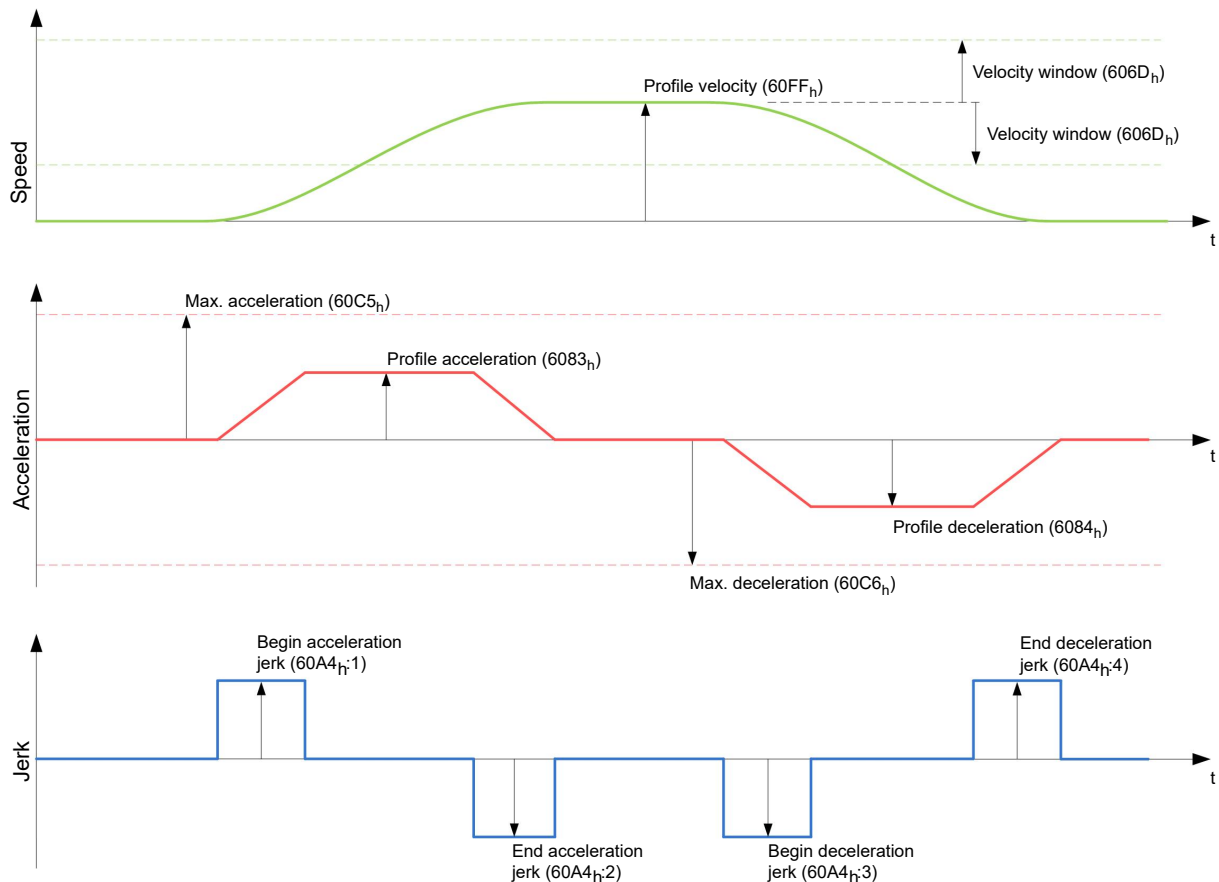


Activation

After the mode is selected in object **6060_h** (Modes Of Operation) and the "Power State machine" (see "**CiA 402 Power State Machine**") is switched to *Operation enabled*, the motor is accelerated to the target speed in object **60FF_h** (see following figures). The speed and acceleration values are taken into account here; for jerk-limited ramps, the jerk-limit values are also taken into account.

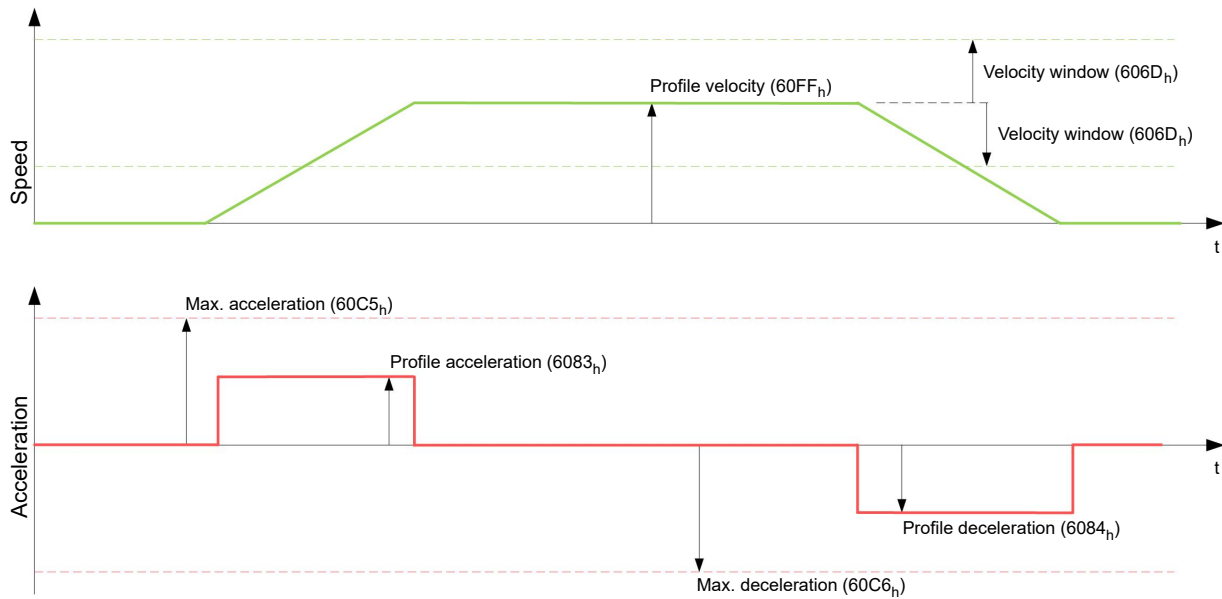
Limitations in the jerk-limited case

The following figure shows the adjustable limits in the jerk-limited case (**6086_h = 3**).



Limitations in the trapezoidal case

This figure shows the adjustable limitations for the trapezoidal case (**6086_h = 0**).



6.4 Profile Torque

6.4.1 Description

In this mode, the torque is preset as a set value and reached via a ramp function.



Note

This mode only functions if **closed loop** is activated, see also **Commissioning Closed Loop**.



Note

The limit switches and, thus, the tolerance bands are active in this mode. For further information on the limit switches, see **Limitation of the range of motion**.

6.4.2 Activation

To activate the mode, the value "4" must be set in object **6060_h** (Modes Of Operation) (see "**CiA 402 Power State Machine**").

6.4.3 Controlword

The following bits in object **6040_h** (controlword) have a special function:

- Bit 8 (Halt): If this bit is set to "1", the motor stops. If this bit is set from "1" to "0", the motor is started up according to the presets. When setting from "0" to "1", the motor is again brought to a standstill, taking the preset values into consideration.

6.4.4 Statusword

The following bits in object **6041_h** (statusword) have a special function:

- Bit 10 (Target Reached): In combination with bit 8 of object **6040_h** (controlword), this bit indicates whether the specified torque is reached (see following table). The target is considered having been met if the current torque (**6077_h Torque Actual Value**) is within a tolerance window (**203D_h Torque Window**) for a specified time (**203E_h Torque Window Time Out**).

| 6040 _h Bit 8 | 6041 _h Bit 10 | Description |
|----------------------------|-----------------------------|------------------------------|
| 0 | 0 | Specified torque not reached |
| 0 | 1 | Specified torque reached |
| 1 | 0 | Axis accelerated |
| 1 | 1 | Axis speed is 0 |

6.4.5 Object entries

All values of the following entries in the object dictionary are to be specified as a thousandth of the maximum torque, which corresponds to the rated current (**203B_h:01_h**). This includes the objects:

- **6071_h** (Target Torque):
Target torque
- **6072_h** (Max Torque):
Maximum torque during the entire ramp (accelerate, maintain torque, decelerate)
- **6074_h** (Torque Demand):
Current output value of the ramp generator (torque) for the controller
- **6087_h** (Torque Slope):
Max. change in torque per second



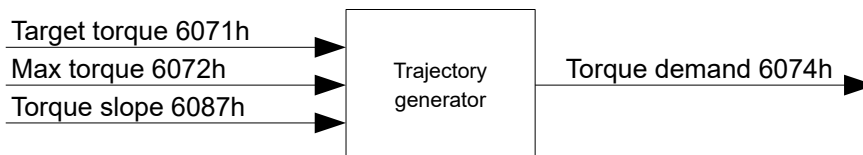
Note

These values are not limited to 100% of the rated current (**203B_h:01_h**). Torque values greater than the rated torque (generated from the rated current) can be achieved if the maximum duration of the peak current (**203B_h:02_h**) is set (see **I2t Motor overload protection**). All torque objects are limited by the peak current.

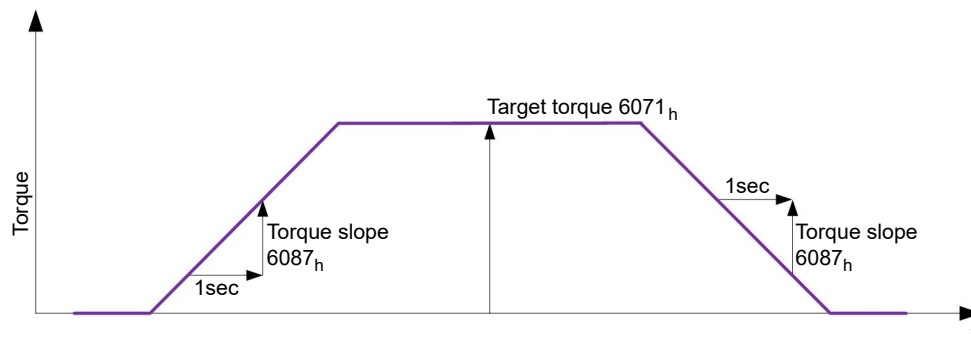
The following objects are also needed for this operating mode:

- **3202_h** Bit 5 (Motor Drive Submode Select):
If this bit is set to "0", the drive controller is operated in the torque-limited Velocity Mode, i.e., the maximum speed can be limited in object **6080_h** and the controller can operate in field weakening mode.
If this bit is set to "1", the controller operates in the ("Real") Torque Mode; the maximum speed cannot be limited here and field weakening mode is not possible.

Objects of the ramp generator



Torque curve



6.5 Homing

6.5.1 Overview

Description

The purpose of the homing method is to align the position zero point of the controller with an encoder index or position switch.

Activation

To activate the mode, the value "6" must be set in object **6060_h** (Modes Of Operation) (see "**CiA 402 Power State Machine**").

If home switches and/or limit switches are used, these special functions must first be activated in the I/O configuration (see "**Digital inputs and outputs**").

Controlword

The following bits in object **6040_h** (controlword) have a special function:

- Bit 4: If the bit is set to "1", referencing is started. This is performed until either the reference position is reached or bit 4 is reset to "0".

Statusword

The following bits in object **6041_h** (statusword) have a special function:

| Bit 13 | Bit 12 | Bit 10 | Description |
|--------|--------|--------|--|
| 0 | 0 | 0 | Homing is performed |
| 0 | 0 | 1 | Homing is interrupted or not started |
| 0 | 1 | 0 | Homing confirmed, but target not yet reached |
| 0 | 1 | 1 | Homing completed |
| 1 | 0 | 0 | Error during homing, motor still turning |
| 1 | 0 | 1 | Error during homing, motor at standstill |

Object entries

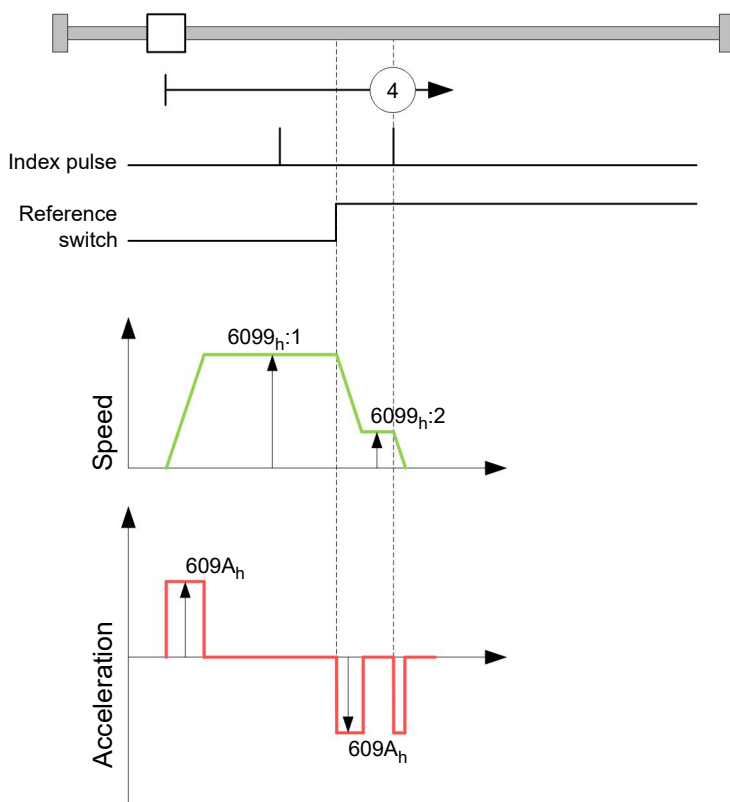
The following objects are necessary for controlling this mode:

- **607C_h** (Home Offset): Specifies the difference between the zero position of the controller and the reference point of the machine in **user-defined units**.
- **6098_h** (Homing Method): Method to be used for referencing (see "**Homing method**")

- **6099_h:01_h** (Speed During Search For Switch):
Speed for the search of the switch
- **6099_h:02_h** (Speed During Search For Zero):
Speed for the search of the index
- **6080_h** (Max Motor Speed): Maximum speed
- **609A_h** (Homing Acceleration):
Starting acceleration and braking deceleration for homing
- **2056_h** (Limit Switch Tolerance Band):
After reaching the positive or negative limit switch, the controller permits a tolerance range in which the motor can continue to run. If this tolerance range is exceeded, the motor stops and the controller switches to the "Fault" state. If limit switches can be actuated during homing, the tolerance range should be selected such that the motor does not exit the tolerance range during braking. Homing cannot otherwise be successfully performed. After homing is completed, the tolerance range can be reset to "0" if this is required by the application.
- **203A_h:01_h** (Minimum Current For Block Detection):
Minimum current threshold which, if exceeded, is to detect the blocking of the motor at a block.
- **203A_h:02_h** (Period Of Blocking):
Specifies the time in ms that the motor is to continue to run against the block after block detection.

Homing speeds

The figure shows the homing speeds using method 4 as an example:



6.5.2 Homing method

Description

The homing method is written as a number in object **6098_h** and decides whether, on a switch edge (rising/falling), a current threshold for block detection or an index pulse is referenced or in which direction homing starts. Methods that use the index pulse of the encoder lie in the number range 1 to 14, 33 and 34. Methods that do not use the index pulse of the encoder lie between 17 and 30, but are identical to methods 1 to 14 with respect to the travel profiles. These numbers are shown in circles in the

following figures. Methods for which no limit switches are used and, instead, travel against a block is to be detected, a minus must be placed before the method number when making the call.

In the following graphics, the negative movement direction is to the left. The *limit switch* is located before the respective mechanical block; the *home switch* is located between the two limit switches. The index pulses come from the connected encoder.

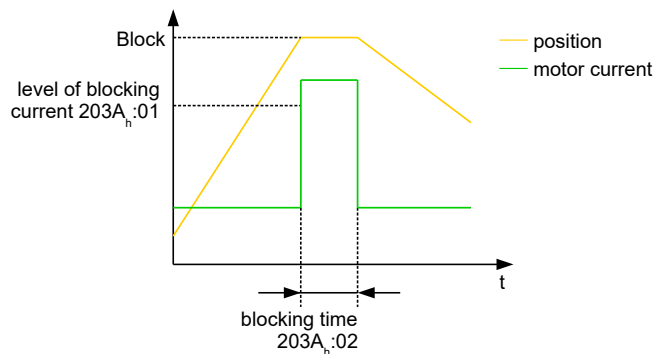
For methods that use homing on block, the same figures apply as for the methods with limit switch. Because nothing is different aside from the missing limit switches, the same figures are used. For the figures here, the limit switches must be replaced with a mechanical block.

Homing on block

Homing on block currently only functions in *closed loop* mode.

"Homing on block" functions like every homing method with the difference that instead of a limit switch, a block (limit stop) is used for positioning. Two settings are to be made here:

1. Current level: In object **203A_n:01**, the current level is defined above which movement against the block is detected.
2. Blocking duration: In object **203A_n:02**, the duration during which the motor moves against the block is set.



Overview of methods

Methods 1 to 14 as well as 33 and 34 use the index pulse of the encoder.

Methods 17 to 32 are identical to methods 1 to 14 with the difference that only limit or home switches are used for referencing and not the index pulse.

- Methods 1 to 14 use an index pulse.
- Methods 17 to 30 do not use an index pulse.
- Methods 33 and 34 reference only to the next index pulse.
- Method 35 references to the current position.

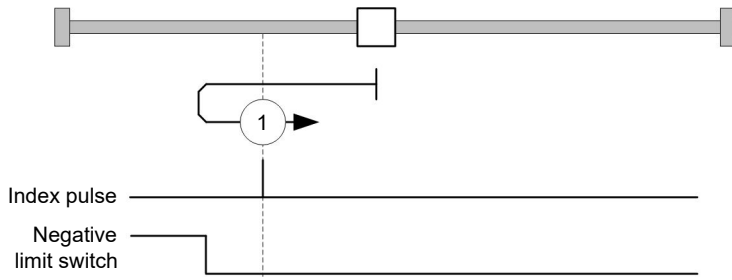
The following methods can be used for homing on block:

- Methods -1 to -2 and -7 to -14 contain an index pulse
- Methods -17 to -18 and -23 to -30 have no index pulse

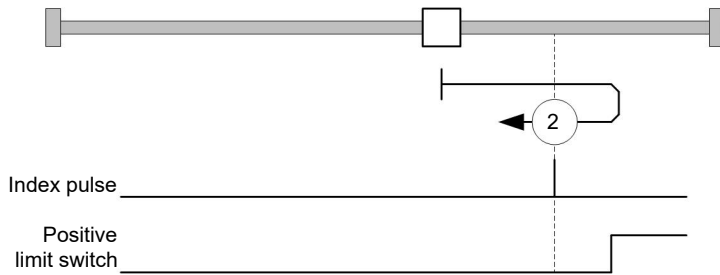
Methods 1 and 2

Reference to limit switches and index pulse.

Method 1 references to negative limit switch and index pulse:



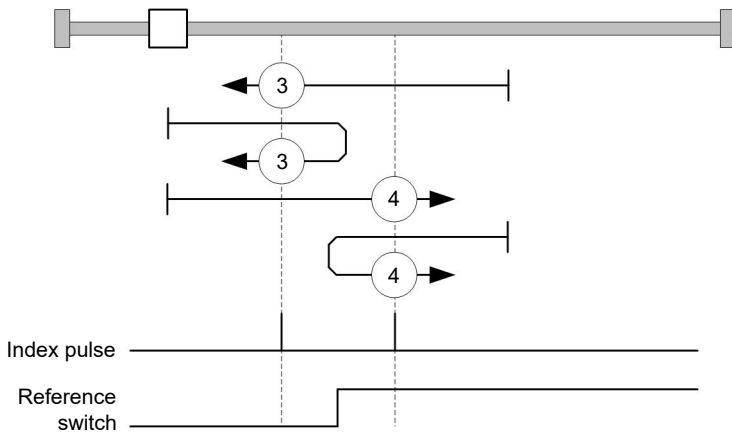
Method 2 references to positive limit switch and index pulse:



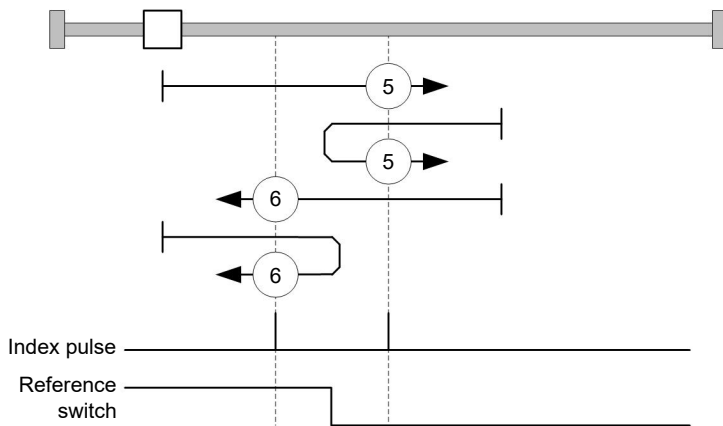
Methods 3 to 6

Reference to the switching edge of the home switch and index pulse.

With methods 3 and 4, the left switching edge of the home switch is used as reference:



With methods 5 and 6, the right switching edge of the home switch is used as reference:

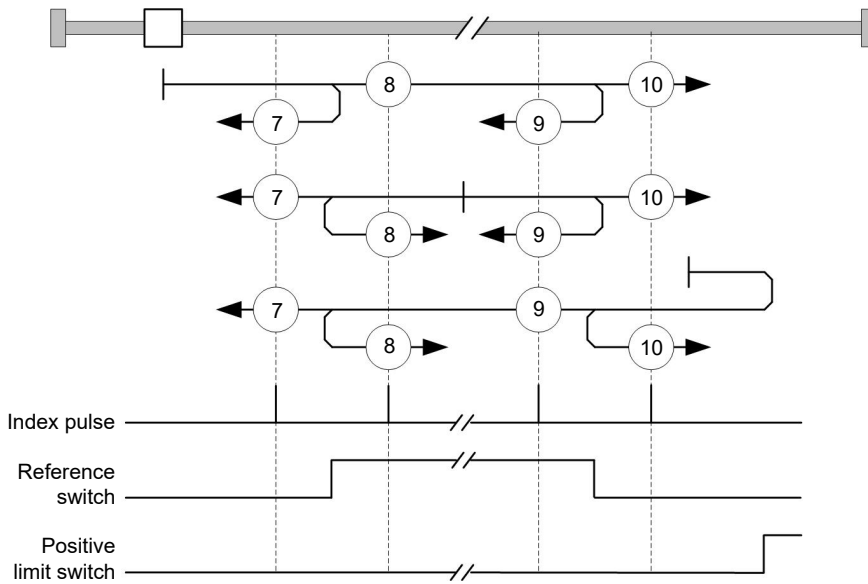


Methods 7 to 14

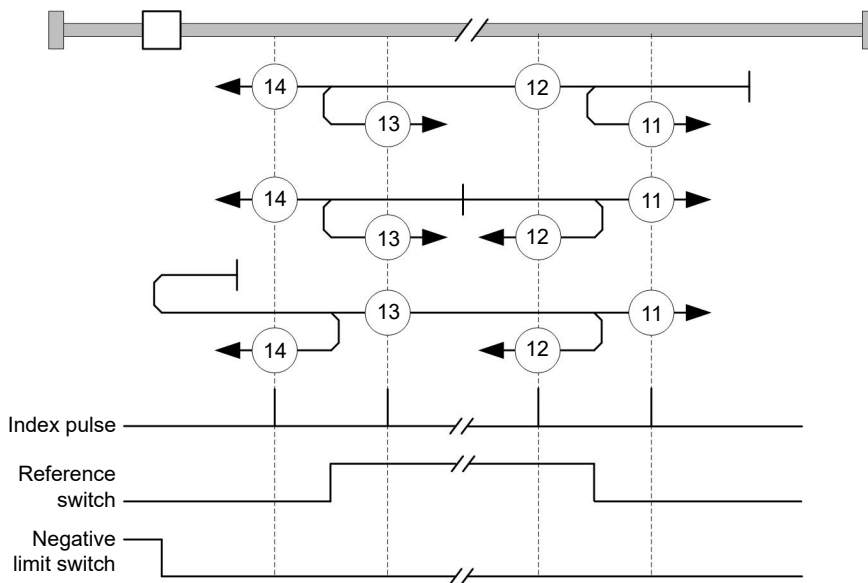
Reference to the home switch and index pulse (with limit switches).

With these methods, the current position relative to the home switch is not important. With method 10, for example, referencing is always performed to the index pulse to the right of the right edge of the home switch.

Methods 7 to 10 take the positive limit switch into account:



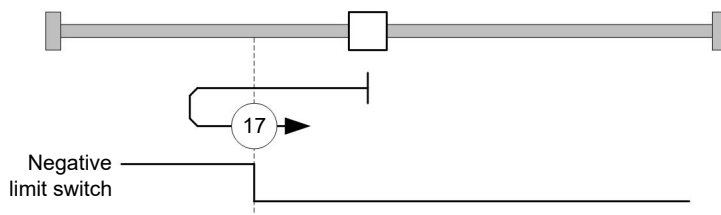
Methods 11 to 14 take the negative limit switch into account:



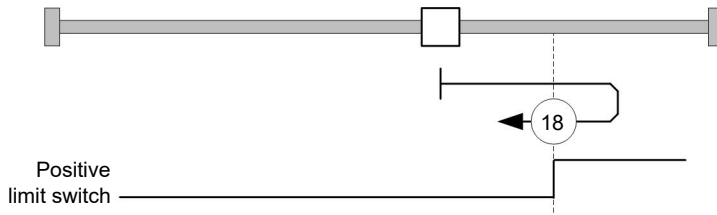
Methods 17 and 18

Reference to the limit switch without the index pulse.

Method 17 references to the negative limit switch:



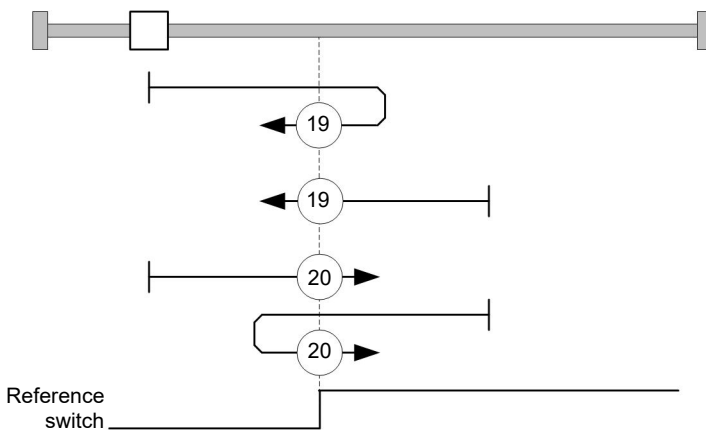
Method 18 references to the positive limit switch:



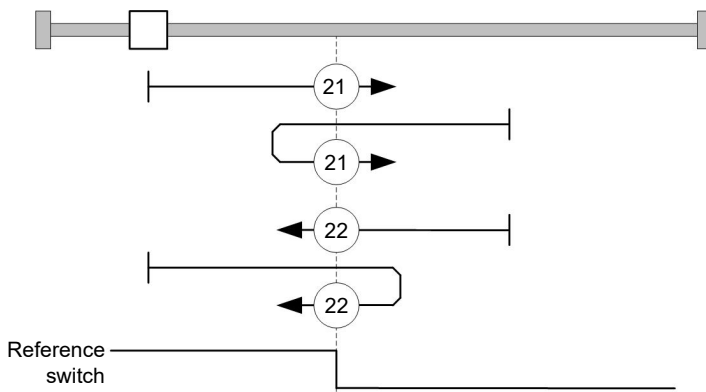
Methods 19 to 22

Reference to the switching edge of the home switch without the index pulse.

With methods 19 and 20 (equivalent to methods 3 and 4), the left switching edge of the home switch is used as reference:



With methods 21 and 22 (equivalent to methods 5 and 6), the right switching edge of the home switch is used as reference:

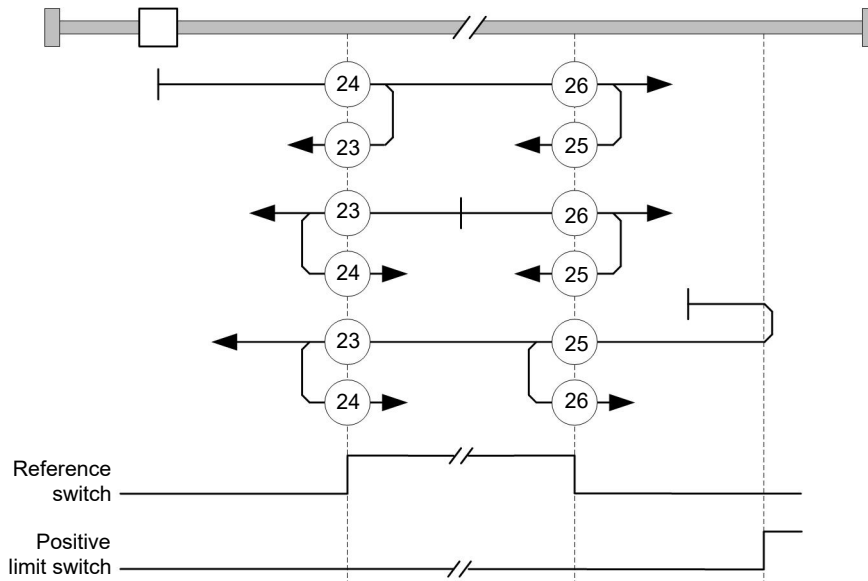


Methods 23 to 30

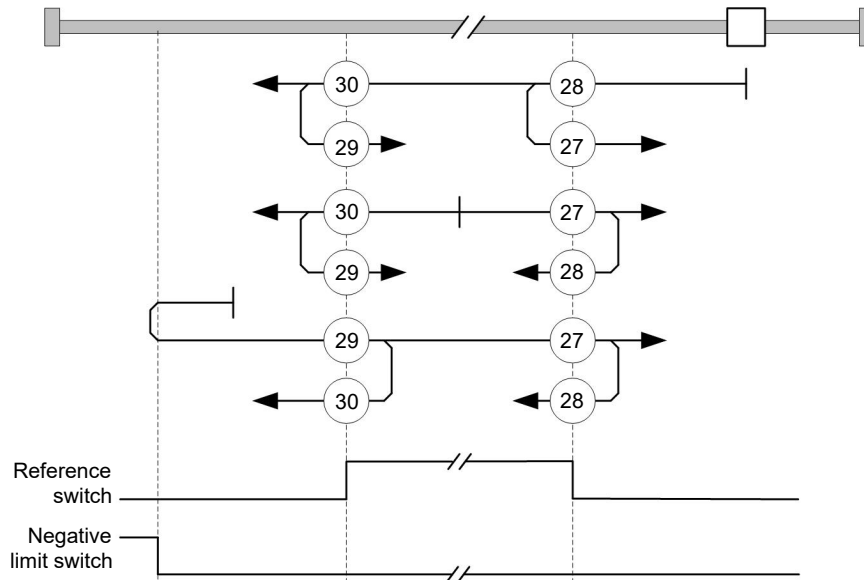
Reference to the home switch without the index pulse (with limit switches).

With these methods, the current position relative to the home switch is not important. With method 26, for example, referencing is always performed to the index pulse to the right of the right edge of the home switch.

Methods 23 to 26 take the positive home switch into account:



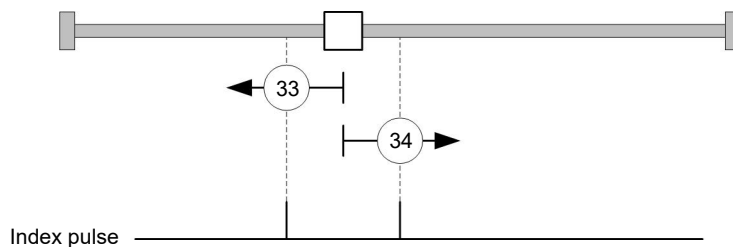
Methods 27 to 30 take the negative home switch into account:



Methods 33 and 34

Reference to the next index pulse.

With these methods referencing is only performed to the respective subsequent index pulse:



Method 35

References to the current position.



Note

For Homing Mode 35, it is not necessary to switch the **CiA 402 Power State Machine** to the "Operation enabled" state. When energizing the motor windings in *open loop* mode, it is thereby possible to prevent the current position from not being exactly 0 after Homing Mode 35.

6.6 Interpolated Position Mode

6.6.1 Overview

Description

Interpolated Position Mode is used to synchronize multiple axes. For this purpose, a higher-level controller performs the ramp and path calculation and passes the respective demand position, at which the axis is to be located at a certain time, to the controller. The controller interpolates between these intermediate position points.



Note

The limit switches and, thus, the tolerance bands are active in this mode. For further information on the limit switches, see **Limitation of the range of motion**.

Synchronization with the SYNC object

For Interpolated Position Mode, it is necessary that the controller synchronizes with the SYNC object (depending on the fieldbus). This SYNC object is to be sent by the higher-level controller in regular intervals. Synchronization occurs as soon as the controller is switched to the *Operational* NMT mode.



Note

Where possible, it is recommended that a time interval of the *SYNC object* be used.

6.6.2 Activation

To activate the mode, the value "7" must be set in object **6060_h** (Modes Of Operation) (see "**CiA 402 Power State Machine**").

6.6.3 Controlword

The following bits in object **6040_h** (controlword) have a special function:

- Bit 4 activates the interpolation when it is set to "1".
- Bit 8 (Halt): If this bit is set to "1", the motor stops. On a transition from "1" to "0", the motor accelerates with the set start ramp to the target speed. On a transition from "0" to "1", the motor brakes and comes to a standstill. The braking deceleration is dependent here on the setting of the "Halt Option Code" in object **605D_h**.

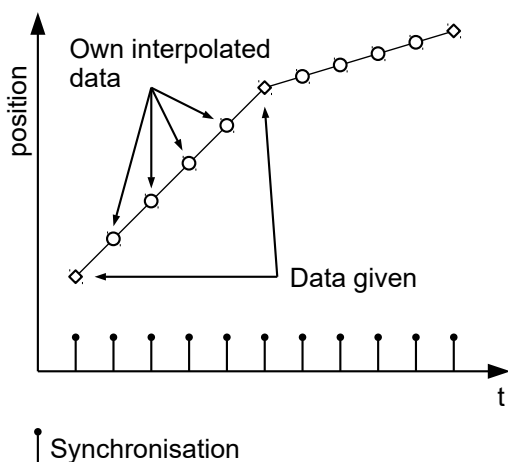
6.6.4 Statusword

The following bits in object **6041_h** (statusword) have a special function:

- Bit 10: Target position reached: This bit is set to "1" if the target position was reached (if the halt bit in the controlword is "0") or the axis has speed 0 (if the halt bit in the last control word was "1").
- Bit 12 (IP mode active): This bit is set to "1" if interpolation is active.

6.6.5 Use

The controller follows a linearly interpolated path between the current position and the preset target position. The (next) target position must be written in record **60C1_h:01_h**.



In the current implementation, only

- linear interpolation
- and a target position

are supported.

6.6.6 Setup

The following setup is necessary:

- **60C2_h:01_h**: Time between two passed target positions in ms.
- **60C4_h:06_h**: This object is to be set to "1" to be able to modify the target position in object **60C1_h:01_h**.
- **6081_h** (Profile Velocity): Maximum speed with which the position is to be approached
- The speed is limited by **607F_h** (Max Profile Velocity) and **6080_h** (Max Motor Speed); the smaller value is used as the limit.
- To be able to turn the motor, the *power state machine* is to be set to the *Operation enabled* state (see **CiA 402 Power State Machine**)

6.6.7 Operation

After setting up, the task of the higher-level controller is to write the target positions to object **60C1_h:01_h** in time.

6.7 Cyclic Synchronous Position

6.7.1 Overview

Description

In this mode, the controller receives an absolute position preset via the fieldbus at fixed time intervals (referred to in the following as a *cycle*). The controller then no longer calculates any ramps, but rather only follows the presets.

The target position is transferred cyclically (via *PDO*). Bit 4 in the controlword does not need to be set (unlike the **Profile Position** mode).



Note

The target is absolute and, thus, independent of how often it was sent per *cycle*.



Note

The limit switches and, thus, the tolerance bands are active in this mode. For further information on the limit switches, see **Limitation of the range of motion**.

Activation

To activate the mode, the value "8" must be set in object **6060_h** (Modes Of Operation) (see "**CiA 402 Power State Machine**").

Controlword

In this mode, the bits of controlword **6040_h** have no special function.

Statusword

The following bits in object **6041_h** (statusword) have a special function:

| Bit | Value | Description |
|-----|-------|--|
| 8 | 0 | The controller is not in sync with the fieldbus |
| 8 | 1 | The controller is in sync with the fieldbus |
| 10 | 0 | Reserved |
| 10 | 1 | Reserved |
| 12 | 0 | Controller does not follow the target; the preset of 607A_h (Target Position) is ignored |
| 12 | 1 | Controller follows the target; object 607A_h (Target Position) is used as the input for position control. |
| 13 | 0 | No following error |
| 13 | 1 | Following error |

6.7.2 Object entries

The following objects are necessary for controlling this mode:

- **607A_h** (Target Position): This object must be written cyclically with the position set value.
- **607B_h** (Position Range Limit): This object contains the preset for an overrun or underrun of the position specification.
- **607D_h** (Software Position Limit): This object defines the limitations within which the position specification (**607A_h**) must be located.
- **6065_h** (Following Error Window): This object specifies a tolerance corridor in both the positive and negative direction from the set specification. If the actual position is outside of this corridor for longer than the specified time (**6066_h**), a following error is reported.
- **6066_h** (Following Error Time Out): This object specifies the time range in milliseconds. If the actual position is outside of the position corridor (**6065_h**) for longer than this time range, a following error is triggered.
- **6085_h** (Quick-Stop Deceleration): This object contains the braking deceleration for the case that a quick-stop is triggered.
- **605A_h** (Quick-Stop Option Code): This object contains the option that is to be executed in the event of a quick-stop.

- **6080_h** (Max Motor Speed): Maximum speed
- **60C2_h:01_h** (Interpolation Time Period): This object specifies the time of a *cycle*; a new set value must be written in **607A_h** in these time intervals.
The following applies here: cycle time = value of **60C2_h:01_h** * 10^{value of 60C2:02} seconds.
- **60C2_h:02_h** (Interpolation Time Index): This object specifies the time basis of the cycles. Currently, only value **60C2_h:02_h=-3** is supported; this yields a time basis of 1 millisecond.

The following objects can be read in this mode:

- **6064_h** (Position Actual Value)
- **606C_h** (Velocity Actual Value)
- **60F4_h** (Following Error Actual Value)

6.8 Cyclic Synchronous Velocity

6.8.1 Overview

Description

In this mode, the controller passes a speed preset via the fieldbus at fixed time intervals (referred to in the following as a *cycle*). The controller then no longer calculates any ramps, but rather only follows the presets.



Note

The limit switches and, thus, the tolerance bands are active in this mode. For further information on the limit switches, see **Limitation of the range of motion**.

Activation

To activate the mode, the value "9" must be set in object **6060_h** (Modes Of Operation) (see "**CiA 402 Power State Machine**").

Controlword

In this mode, the bits of controlword **6040_h** have no special function.

Statusword

The following bits in object **6041_h** (statusword) have a special function:

| Bit | Value | Description |
|-----|-------|--|
| 8 | 0 | The controller is not in sync with the fieldbus |
| 8 | 1 | The controller is in sync with the fieldbus |
| 10 | 0 | Reserved |
| 10 | 1 | Reserved |
| 12 | 0 | Controller does not follow the target; the preset of 60FF_h (Target Velocity) is ignored |
| 12 | 1 | Controller follows the target; object 60FF_h (Target Velocity) is used as the input for position control. |
| 13 | 0 | Reserved |
| 13 | 1 | Reserved |

6.8.2 Object entries

The following objects are necessary for controlling this mode:

- **60FF_h** (Target Velocity): This object must be written cyclically with the speed set value.
- **6085_h** (Quick-Stop Deceleration): This object contains the braking deceleration for the case that a quick-stop is triggered (see "**CiA 402 Power State Machine**").
- **605A_h** (Quick-Stop Option Code): This object contains the option that is to be executed in the event of a quick-stop (see "**CiA 402 Power State Machine**").
- **6080_h** (Max Motor Speed): Maximum speed
- **60C2_h:01_h** (Interpolation Time Period): This object specifies the time of a *cycle*; a new set value must be written in **60FF_h** in these time intervals.
The following applies here: cycle time = value of **60C2_h:01_h** * 10^{value of 60C2:02} seconds.
- **60C2_h:02_h** (Interpolation Time Index): This object specifies the time basis of the cycles. Currently, only value **60C2_h:02_h**=-3 is supported; this yields a time basis of 1 millisecond.

The following objects can be read in this mode:

- **606C_h** (Velocity Actual Value)
- **607E_h** (Polarity)

6.9 Cyclic Synchronous Torque

6.9.1 Overview

Description

In this mode, the controller passes an absolute torque preset via the fieldbus at fixed time intervals (referred to in the following as a *cycle*). The controller then no longer calculates any ramps, but rather only follows the presets.



Note

This mode only functions if **closed loop** is activated, see also **Commissioning closed loop**.



Note

The limit switches and, thus, the tolerance bands are active in this mode. For further information on the limit switches, see **Limitation of the range of motion**.

Activation

To activate the mode, the value "10" must be set in object **6060_h** (Modes Of Operation) (see "**CiA 402 Power State Machine**").

Controlword

In this mode, the bits of controlword **6040_h** have no special function.

Statusword

The following bits in object **6041_h** (statusword) have a special function:

| Bit | Value | Description |
|-----|-------|---|
| 8 | 0 | The controller is not in sync with the fieldbus |
| 8 | 1 | The controller is in sync with the fieldbus |

| Bit | Value | Description |
|-----|-------|--|
| 10 | 0 | Reserved |
| 10 | 1 | Reserved |
| 12 | 0 | Controller does not follow the target; the preset of 6071_h (Target Torque) is ignored |
| 12 | 1 | Controller follows the target; object 6071_h (Target Torque) is used as the input for position control. |
| 13 | 0 | Reserved |
| 13 | 1 | Reserved |

6.9.2 Object entries

The following objects are necessary for controlling this mode:

- **6071_h** (Target Torque): This object must be written cyclically with the torque set value and is to be set relative to **6072_h**.
- **6072_h** (Max Torque): Describes the maximum permissible torque.
- **6080_h** (Max Motor Speed): Maximum speed
- **60C2_h:01_h** (Interpolation Time Period): This object specifies the time of a *cycle*; a new set value must be written in **60FF_h** in these time intervals.
The following applies here: cycle time = value of **60C2_h:01_h** * 10^{value of 60C2:02} seconds.
- **60C2_h:02_h** (Interpolation Time Index): This object specifies the time basis of the cycles. Currently, only value **60C2_h:02_h=-3** is supported; this yields a time basis of 1 millisecond.

The following objects can be read in this mode:

- **606C_h** (Velocity Actual Value)
- **6074_h** (Torque Demand)

6.10 Clock-direction mode

6.10.1 Description

In clock-direction mode, the motor is operated via two inputs by a higher-level positioning controller with clock and direction signal. On each clock signal, the motor moves one step in the direction corresponding to the direction signal.



Note

The limit switches and, thus, the tolerance bands are active in this mode. For further information on the limit switches, see **Limitation of the range of motion**.

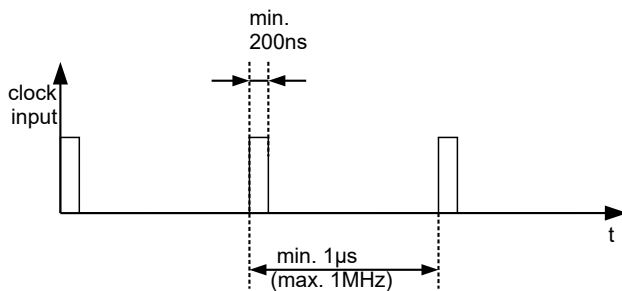
6.10.2 Activation

To activate the mode, the value "-1" (or "FFh") must be set in object **6060_h** (Modes Of Operation) (see "CiA 402 Power State Machine").

6.10.3 General

The following data apply for every subtype of the clock-direction mode:

- The maximum frequency of the input pulse is 1 MHz; the ON pulse should not be less than 200 ns.



- The steps are scaled using objects **2057_h** and **2058_h**. The following formula applies here:

$$\text{step width per pulse} = \frac{2057_{\text{h}}}{2058_{\text{h}}}$$

The "step size per pulse" value is set to 128 (**2057_h**=128 and **2058_h**=1) ex works, which corresponds to a quarter step per pulse. A full step is the value "512", a half step per pulse corresponds to "256", etc.



Note

For a stepper motor with 50 pole pairs, 200 full steps correspond to one mechanical revolution of the motor shaft.

In *clock-direction mode*, the BLDC motors are also handled as stepper motors by the controller. This means that for a BLDC motor with, e.g., 3 pole pairs, 12 (=4*3) full steps correspond to one revolution.



Note

If there is a change of direction, a time of at least 35 µs must elapse before the new clock signal is applied.

6.10.4 Statusword

The following bits in object **6041_h** (statusword) have a special function:

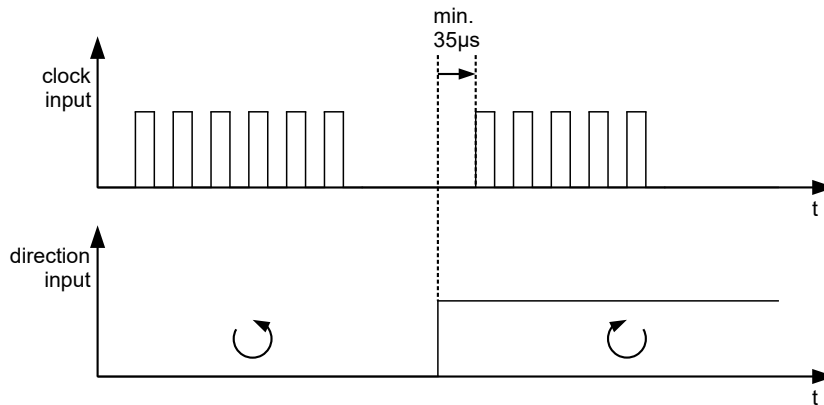
- Bit 13 (Following Error): This bit is set in *closed loop mode* if the following error is greater than the set limits (**6065_h** (Following Error Window) and **6066_h** (Following Error Time Out)).

6.10.5 Subtypes of the clock-direction mode

Clock-direction mode (TR mode)

To activate the mode, object **205B_h** must be set to the value "0" (factory settings).

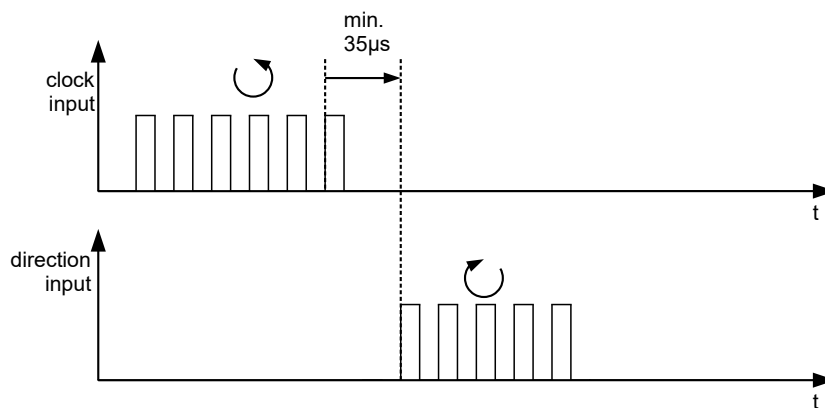
In this mode, the pulses must be preset via the clock input; the signal of the direction input specifies the direction of rotation here (see following graphic).



Right / left rotation mode (CW / CCW mode)

To activate the mode, object **205B_h** must be set to the value "1".

In this mode, the input that is used decides the direction of rotation (see following graphic).



6.11 Auto setup

6.11.1 Description

To determine a number of parameters related to the motor and the connected sensors (encoders/Hall sensors), an *auto setup* is performed. **Closed Loop** operation requires a successfully completed *auto setup*. With the Plug & Drive motors, it is not necessary to perform an *auto setup*, as this was already performed at the factory. For details, see **the corresponding section in chapter Commissioning**.



Note

The limit switches and, thus, the tolerance bands are active in this mode. For further information on the limit switches, see **Limitation of the range of motion**.

6.11.2 Activation

To activate the mode, the value "-2" ("FE_h") must be set in object **6060_h** (Modes Of Operation) (see **CiA 402 Power State Machine**).

6.11.3 Controlword

The following bits in object **6040_n** (controlword) have a special function:

- Bit 4 starts a travel command. This is carried out on a transition from "0" to "1".

6.11.4 Statusword

The following bits in object **6041_n** (statusword) have a special function:

- Bit 10: Indexed: indicates whether (= "1") or not (= "0") an encoder index was found.
- Bit 12: Aligned: this bit is set to "1" after *auto setup* has concluded

7 Special functions

7.1 Digital inputs and outputs

This controller is equipped with digital inputs and outputs.

7.1.1 Bit assignment

The software of the controller assigns each input and output two bits in the respective object (e.g., **60FDh Digital Inputs** or **60FEh Digital Outputs**):

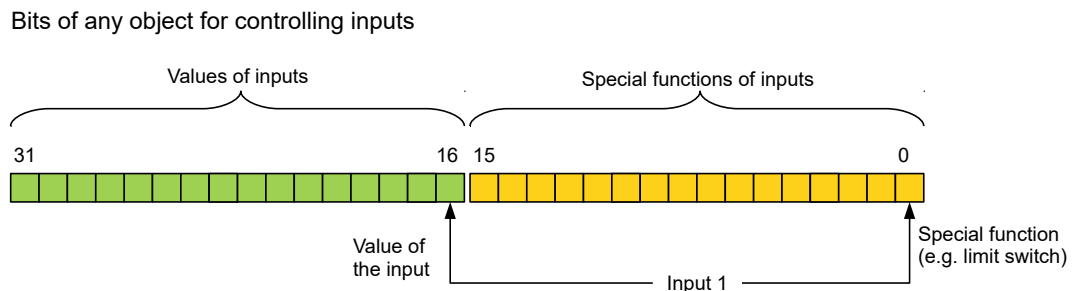
1. The first bit corresponds to the special function of an output or input. These functions are always available on bits 0 to 15 (inclusive) of the respective object. These include the limit switches and clock-direction inputs for the digital inputs and the brake control for the outputs.
2. The second bit shows the output/input as a level; these are then available on bits 16 to 31.

Example

To manipulate the value of output 2, always use bit 17 in **60FE_h**.

To activate the "negative limit switch" special function of input 1, set bit 0 in **3240_h:01_h**; to query the status of the input, read bit 0 in **60FD_h**. Bit 16 in **60FD_h** also shows the status of input 1 (independent of whether or not the special function of the input was activated).

This assignment is graphically illustrated in the following drawing.



7.1.2 Digital inputs

Overview



Note

For digital inputs with 5 V, the length of the supply lines must not exceed 3 meters.



Note

The digital inputs are sampled once per millisecond. Signal changes at the input less than one millisecond in duration are not processed.

The following inputs are available:

| Input | Special function | Switching threshold switchable | Differential / single-ended |
|-------|---|--|--|
| 1 | Negative limit switch | yes, 5 V or 24 V (see 3240_h:06_h) | The inputs can only be switched together (see 3240_h:07_h) |
| 2 | Positive limit switch / direction input in clock-direction mode | yes, 5 V or 24 V (see 3240_h:06_h) | |
| 3 | Home switch / Clock input in clock-direction mode | yes, 5 V or 24 V (see 3240_h:06_h) | |
| 4 | None | yes, 5 V or 24 V (see 3240_h:06_h) | |
| 5 | None | yes, 5 V or 24 V (see 3240_h:06_h) | |
| 6 | None | yes, 5 V or 24 V (see 3240_h:06_h) | |

If you set **3240_h:07_h** to the value "1", three differential inputs are available instead of six single-ended inputs:

| Pin | Function |
|-----|-----------|
| 2 | - Input 1 |
| 3 | Input 1 |
| 4 | - Input 2 |
| 5 | Input 2 |
| 6 | - Input 3 |
| 7 | Input 3 |

Object entries

The value of an input can be manipulated using the following OD settings, whereby only the corresponding bit acts on the input here.

- **01_h**: This bit allows special functions of an input to be switched off (value "0") or on (value "1"). If input 1 is not used as, e.g., a negative limit switch, the special function must be switched off to prevent an erroneous response to the signal generator. The object has no effect on bits 16 to 31. The firmware evaluates the following bits:

- Bit 0: Negative limit switch
- Bit 1: Positive limit switch
- Bit 2: Home switch

If, for example, two limit switches and one home switch are used, bits 0–2 in **3240_h:01_h** must be set to "1".

- **02_h**: This subindex switches from normally open logic (a logical high level at the input yields the value "1" in object **60FD_h**) to normally closed logic (the logical high level at the input yields the value "0").

This applies for the special functions (except for the clock and direction inputs) and for the normal inputs. If the bit has the value "0", normally open logic applies; for the value "1", normally closed logic applies. Bit 0 changes the logic of input 1, bit 1 changes the logic of input 2, etc.

- **03_h**: This subindex switches on the software simulation of input values if the corresponding bit is set to "1".

In this case, the actual values are no longer used in object **3240_h:04_h**, but rather the set values for the respective input. Bit 0 corresponds to input 1 here, bit 1 to input 2, etc.

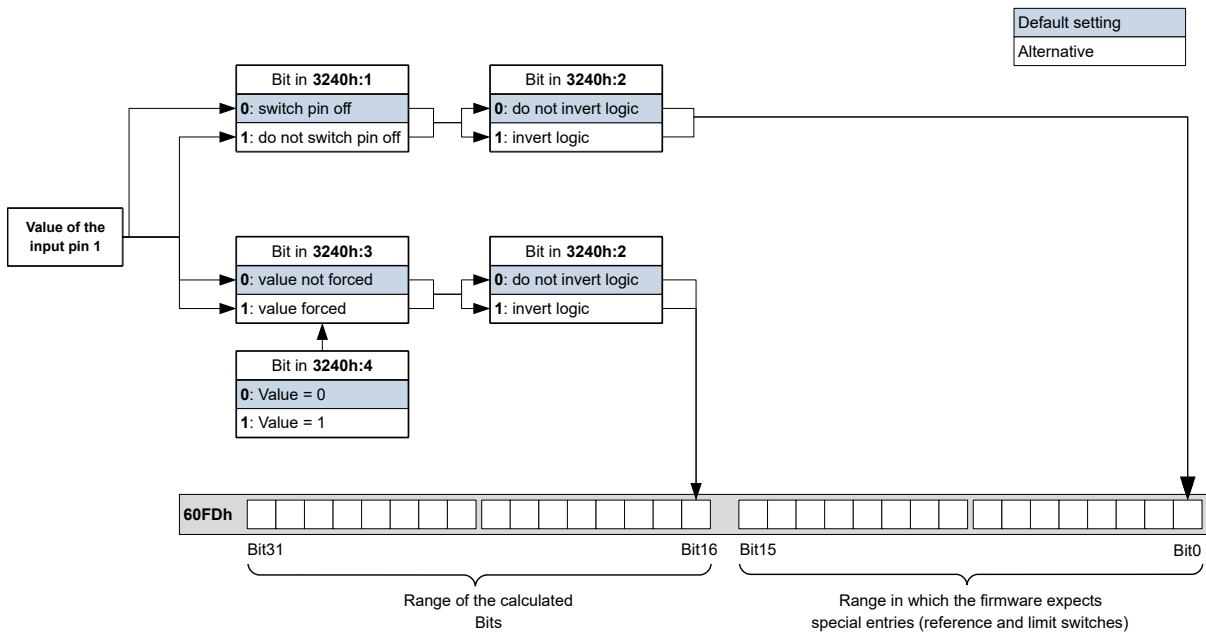
- **04_h**: This bit specifies the value that is to be read as the input value if the same bit was set in object **3240_h:03_h**.
- **05_h**: This subindex always contains the read, unmodified input value.

- **06_h:_h**: (Input Range Select): This can be used to switch inputs – that are equipped with this function – from the switching threshold of 5 V (bit is "0") to the switching threshold of 24 V (bit is "1"). Bit 0 corresponds to input 1 here, bit 1 to input 2, etc.
- **07_h**: With the inputs, this subindex switches between "single-ended input" (value "0" in the subindex) and "differential input" (value "1" in the subindex) if the input supports this function.
- **08_h**: This subindex deactivates (value "0") or activates (value "1") the input routing.
- **60FD_h** (Digital Inputs): This object contains a summary of the inputs and the special functions.

Computation of the inputs

Computation of the input signal using the example of input 1:

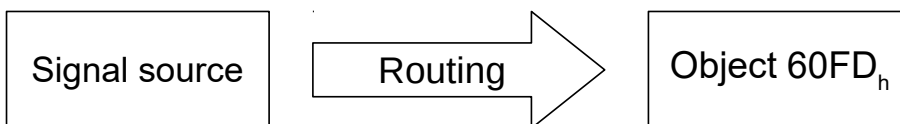
The value at bit 0 of object **60FD_h** is interpreted by the firmware as negative limit switch; the result of the complete computation is stored in bit 16.



Input Routing

Principle

To perform the assignment of the inputs more flexibly, there is a mode called *Input Routing Mode*. This assigns a signal of a source to a bit in object **60FD_h**.



Activation

This mode is activated by setting object **3240_h:08_h** (Routing Enable) to 1.



Note

Entries **3240_h:01_h** to **3240_h:04_h** then have **no** function until Input Routing is again switched off.



Note

If *Input Routing* is switched on, the initial values of **3242_h** are changed and correspond to the function of the input as it was before activation of *Input Routing*. The inputs of the controller behave the same with activation of *Input Routing*. Therefore, you should not switch back and forth between the normal mode and *Input Routing*.

Routing

Object 3242_h determines which signal source is routed to which bit of **60FD_h**. Subindex 01_h of 3242_h determines bit 0, subindex 02_h determines bit 1, and so forth. You can find the signal sources and their numbers in the following lists.

| Number | | |
|--------|-----|-----------------------|
| dec | hex | Signal source |
| 00 | 00 | Signal is always 0 |
| 01 | 01 | Physical input 1 |
| 02 | 02 | Physical input 2 |
| 03 | 03 | Physical input 3 |
| 04 | 04 | Physical input 4 |
| 05 | 05 | Physical input 5 |
| 06 | 06 | Physical input 6 |
| 07 | 07 | Physical input 7 |
| 08 | 08 | Physical input 8 |
| 09 | 09 | Physical input 9 |
| 10 | 0A | Physical input 10 |
| 11 | 0B | Physical input 11 |
| 12 | 0C | Physical input 12 |
| 13 | 0D | Physical input 13 |
| 14 | 0E | Physical input 14 |
| 15 | 0F | Physical input 15 |
| 16 | 10 | Physical input 16 |
| 68 | 44 | Encoder input "A" |
| 69 | 45 | Encoder input "B" |
| 70 | 46 | Encoder input "Index" |

The following table describes the inverted signals of the previous table.

| Number | | |
|--------|-----|---------------------------|
| dec | hex | Signal source |
| 128 | 80 | Signal is always 1 |
| 129 | 81 | Inverted physical input 1 |
| 130 | 82 | Inverted physical input 2 |
| 131 | 83 | Inverted physical input 3 |
| 132 | 84 | Inverted physical input 4 |
| 133 | 85 | Inverted physical input 5 |
| 134 | 86 | Inverted physical input 6 |

| Number | | |
|--------|-----|--------------------------------|
| dec | hex | Signal source |
| 135 | 87 | Inverted physical input 7 |
| 136 | 88 | Inverted physical input 8 |
| 137 | 89 | Inverted physical input 9 |
| 138 | 8A | Inverted physical input 10 |
| 139 | 8B | Inverted physical input 11 |
| 140 | 8C | Inverted physical input 12 |
| 141 | 8D | Inverted physical input 13 |
| 142 | 8E | Inverted physical input 14 |
| 143 | 8F | Inverted physical input 15 |
| 144 | 90 | Inverted physical input 16 |
| 196 | C4 | Inverted encoder input "A" |
| 197 | C5 | Inverted encoder input "B" |
| 198 | C6 | Inverted encoder input "Index" |

Example

Input 1 is to be routed to bit 16 of object **60FD_h**:

The number of the signal source for input 1 is "1". The routing for bit 16 is written in **3242_h:11_h**.

Hence, object **3242_h:11_h** must be set to the value "1".

7.1.3 Digital outputs

Outputs

The outputs are controlled via object **60FE_h**. Here, output 1 corresponds to bit 16 in object **60FE_h**, output 2 corresponds to bit 17, etc., as with the inputs. The outputs with special functions are again entered in the firmware in the lower bits 0 to 15. The only bit assigned at the present time is bit 0, which controls the motor brake.

Wiring



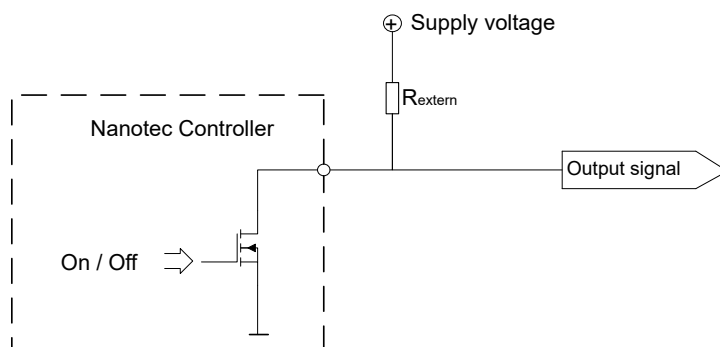
Note

Always observe the maximum capacity of the output (see **Pin assignment**).

The outputs are implemented as "open drain". Hence, an external voltage supply is always necessary.

Example

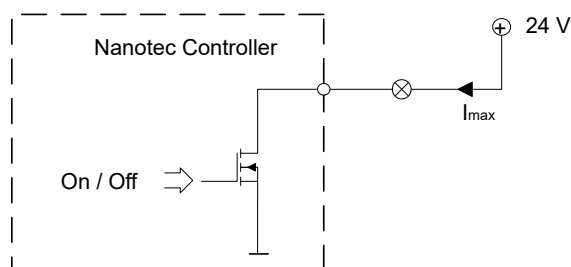
The digital output signal should continue to be used. For this purpose, a circuit as shown in the following figure is to be realized.



With a supply voltage of +24 V, a resistance value R_{external} of 10 k Ω is recommended.

Example

A simple load is to be used with the digital output.



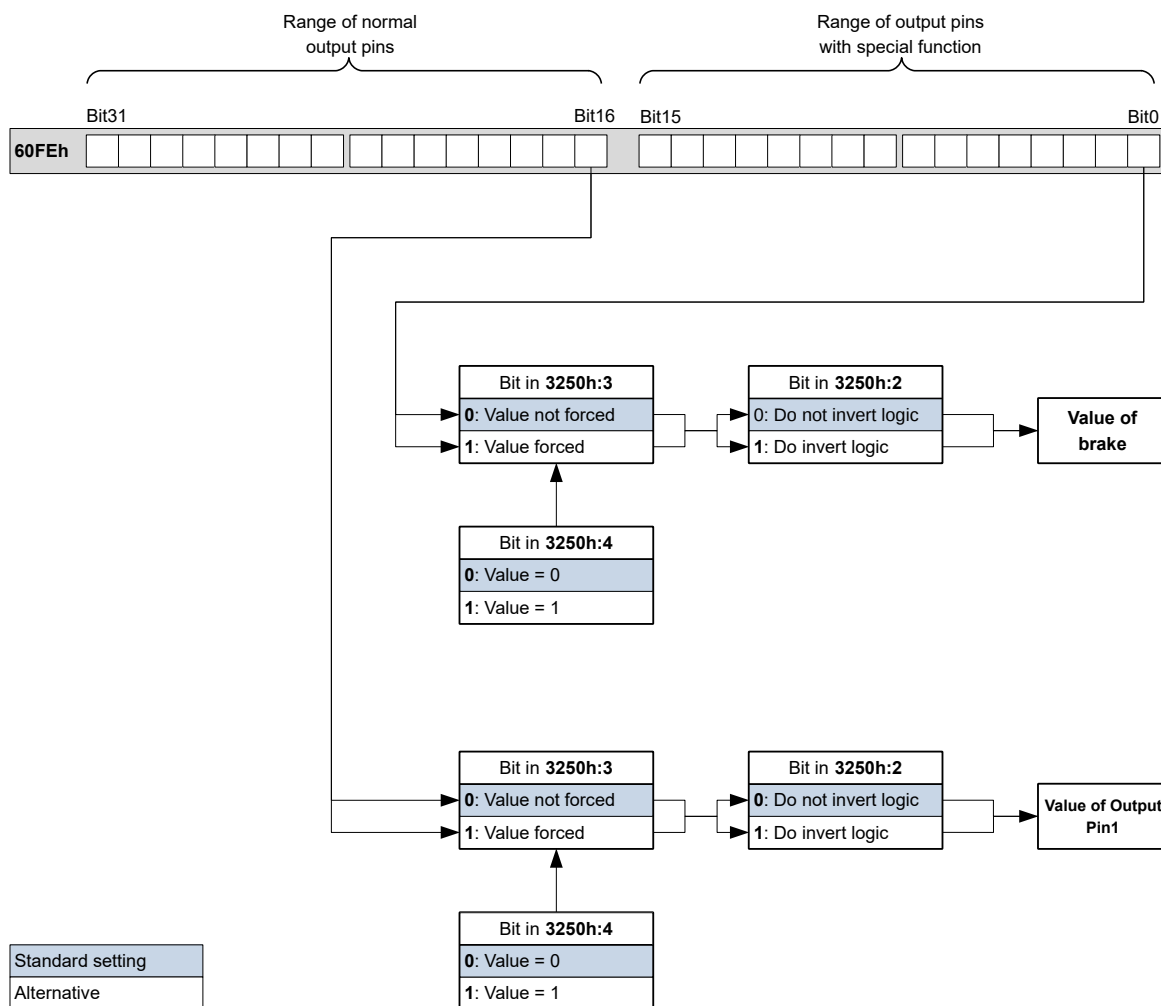
Object entries

Additional OD entries are available for manipulating the value of the outputs (see the following example for further information). As with the inputs, only the bit at the corresponding location acts on the respective output:

- **3250_h:01_h**: No function.
- **3250_h:02_h**: This is used to switch the logic from *normally open* to *normally closed*. Configured as *normally open*, the input outputs a logical high level if the bit is "1". With the *normally closed* configuration, a logical low level is output accordingly for a "1" in object **60FE_h**.
- **3250_h:03_h**: If a bit is set here, the output is controlled manually. The value for the output is then in object **3250_h:4_h**; this is also possible for the brake output.
- **3250_h:04_h**: The bits in this object specify the output value that is to be applied at the output if manual control of the output is activated by means of object **3250_h:03_h**.
- **3250_h:05_h**: This object has no function and is included for reasons of compatibility.

Computation of the outputs

Example for calculating the bits of the outputs:

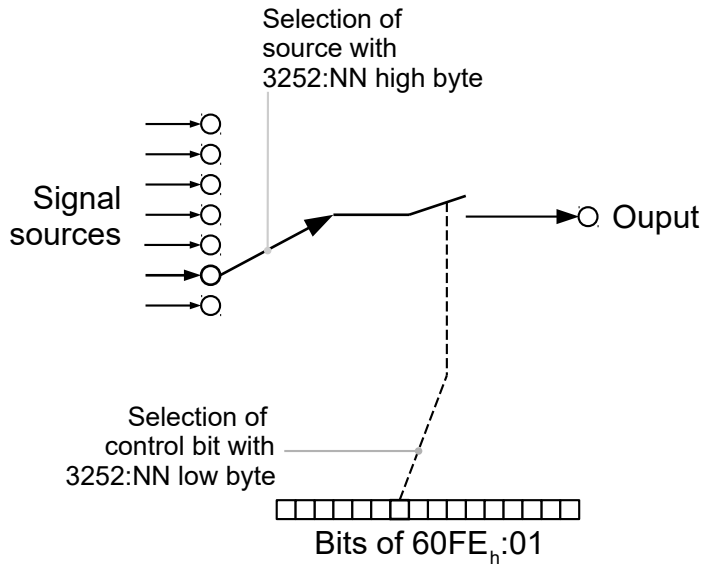


Output Routing

Principle

The "Output Routing Mode" assigns an output a signal source; a control bit in object **60FE_h:01_h** switches the signal on or off.

The source is selected with **3252_h:01** to **05** in the "high byte" (bit 15 to bit 8). The assignment of a control bit from object **60FE_h:01_h** is performed in the "low byte" (bit 7 to bit 0) of **3252_h:01_h** to **05** (see following figure).



Activation

This mode is activated by setting object **3250_h:08_h** (Routing Enable) to 1.



Note

Entries **3250_h:01_h** to **3250:04_h** then have **no** function until "Output Routing" is switched off again.

Routing

The subindex of object **3252_h** determines which signal source is routed to which output. The output assignments are listed in the following:

| Subindex 3252 _h | Output Pin |
|----------------------------|--|
| 01 _h | Configuration of the brake output (if available) |
| 02 _h | Configuration of output 1 |
| 03 _h | Configuration of output 2 (if available) |
| 04 _h | Configuration of output 3 (if available) |
| 05 _h | Configuration of output 4 (if available) |



Note

The maximum output frequency of the brake output, output 1 and output 2 is 10 kHz. All other outputs can only produce signals up to 500 Hz.

Subindices **3252_h:01_h** to **05_h** are 16 bits wide, whereby the high byte selects the signal source (e.g., the PWM generator) and the low byte determines the control bit in object **60FE_h:01**.

Bit 7 of **3252_h:01_h** to **05** inverts the controller from object **60FE_h:01**. Normally, value "1" in object **60FE_h:01** switches on the signal; if bit 7 is set, the value "0" switches on the signal.

Number in 3252:01 to 05

| | |
|-------------------|----------------------|
| 00XX _h | Output is always "1" |
|-------------------|----------------------|

Number in 3252:01 to 05

| | |
|-------------------|--|
| 01XX _h | Output is always "0" |
| 02XX _h | Encoder signal (6063 _h) with frequency divider 1 |
| 03XX _h | Encoder signal (6063 _h) with frequency divider 2 |
| 04XX _h | Encoder signal (6063 _h) with frequency divider 4 |
| 05XX _h | Encoder signal (6063 _h) with frequency divider 8 |
| 06XX _h | Encoder signal (6063 _h) with frequency divider 16 |
| 07XX _h | Encoder signal (6063 _h) with frequency divider 32 |
| 08XX _h | Encoder signal (6063 _h) with frequency divider 64 |
| 09XX _h | Position Actual Value (6064 _h) with frequency divider 1 |
| 0AXX _h | Position Actual Value (6064 _h) with frequency divider 2 |
| 0BXX _h | Position Actual Value (6064 _h) with frequency divider 4 |
| 0CXX _h | Position Actual Value (6064 _h) with frequency divider 8 |
| 0DXX _h | Position Actual Value (6064 _h) with frequency divider 16 |
| 0EXX _h | Position Actual Value (6064 _h) with frequency divider 32 |
| 0FXX _h | Position Actual Value (6064 _h) with frequency divider 64 |
| 10XX _h | Brake PWM signal that is configured with object 2038 _h :05 _h and 06 _h |
| 11XX _h | Inverted brake PWM signal that is configured with object 2038 _h :05 _h and 06 _h |

Example

The encoder signal (**6063**_h) is to be applied to output 1 with a frequency divider 4. The output is to be controlled with bit 5 of object **60FE**:01.

- **3250**_h:08_h = 1 (activate routing)
- **3252**_h:02_h = 0405_h (04XX_h + 0005_h) Dabei ist:
- 04XX_h: Encoder signal with frequency divider 4
- 0005_h: Selection of bit 5 of **60FE**:01

The output is switched on by setting bit 5 in object **60FE**:01.

Example

The brake PWM signal is to be applied to output 2. Because the automatic brake control uses bit 0 of **60FE**:01_h, this should be used as control bit.

- **3250**_h:08_h = 1 (activate routing)
- **3252**_h:03_h = 1080_h (=10XX_h + 0080_h). Where:
 - 10XX_h: Brake PWM signal
 - 0080_h: Selection of the inverted bit 0 of object **60FE**:01

7.2 I²t Motor overload protection

7.2.1 Description



Note

For stepper motors, only the rated current is specified, not a maximum current. No liability is therefore assumed when using I²t with stepper motors.

The goal of I²t motor overload protection is to protect the motor from damage and, at the same time, operate it normally up to its thermal limit.

This function is only available if the controller is in the **closed loop mode** (bit 0 of object **3202_h** must be set to "1").

There is an exception: If I²t is activated in *open loop* mode, the current is limited to the set rated current, even if the set maximum current is larger. This function was implemented for safety reasons so that one can switch from *closed loop* mode with very high, brief maximum current to *open loop* mode without damaging the motor.

7.2.2 Object entries

The following objects affect I²t motor overload protection:

- **2031_h**: Peak Current – specifies the maximum current in mA.
- **203B_h:1_h** Nominal Current – specifies the rated current in mA.
- **203B_h:2_h** Maximum Duration Of Peak Current – specifies the maximum duration of the maximum current in ms.

The following objects indicate the current state of I²t:

- **203B_h:3_h** Threshold – specifies the limit in mAs that determines whether the maximum current or rated current is switched to.
- **203B_h:4_h** CalcValue – specifies the calculated value that is compared with the threshold for setting the current.
- **203B_h:5_h** LimitedCurrent – shows the momentary current value that was set by I²t.
- **203B_h:6_h** Status:
 - Value = "0": I²t deactivated
 - Value = "1": I²t activated

7.2.3 Activation

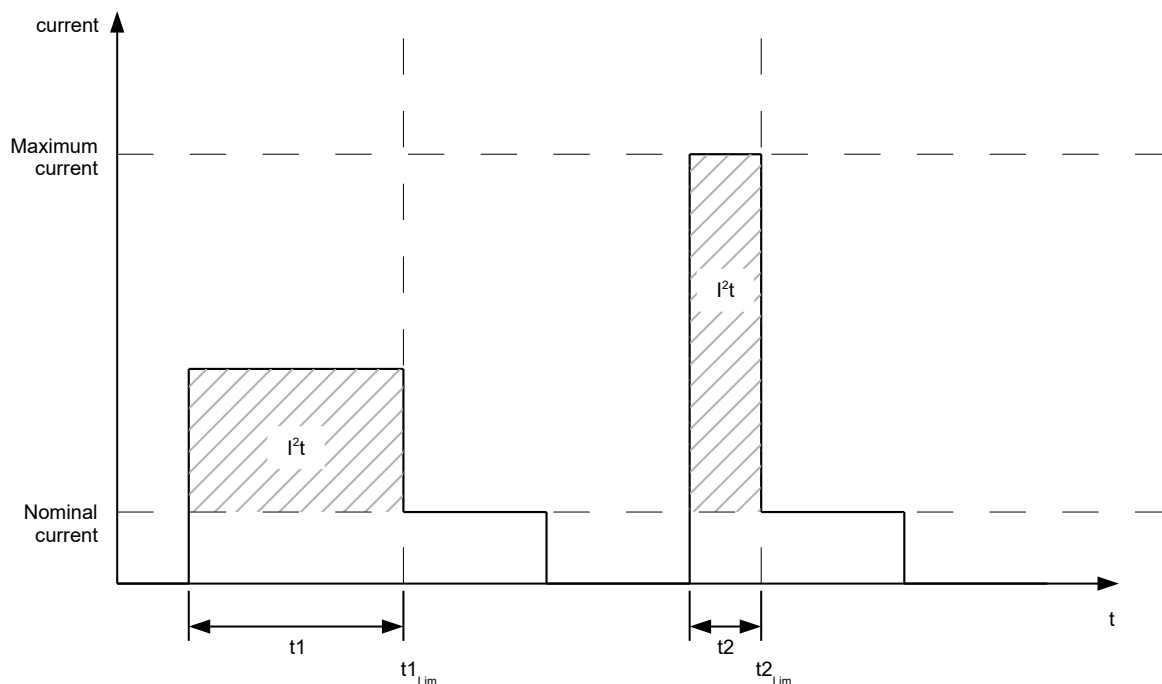
Closed loop must be activated, (bit 0 of object **3202_h** set to "1", see also chapter **Closed Loop**). To activate the mode, the three object entries mentioned above (**2031_h**, **203B_h:1_h**, **203B_h:2_h**) must have been appropriately specified. This means that the maximum current must be greater than the rated current and a time value for the maximum duration of the maximum current must be entered. If these conditions are not met, the I²t functionality remains deactivated.

7.2.4 Function of I²t

From the specification of rated current, maximum current and maximum duration of the maximum current, an I²t_{Lim} is calculated.

The motor can run with maximum current until the calculated I²t_{Lim} is reached. The current is then immediately reduced to the rated current.

The relationships are illustrated again in the following diagram.



In the first section, t_1 , the current value is higher than the rated current. At time t_{1_Lim} , $I^2_{t_Lim}$ is reached and the current is limited to the rated current. A current that corresponds to the maximum current then occurs for a period of time t_2 . Hence, the value for $I^2_{t_Lim}$ is reached more quickly than in time t_1 .

7.3 Saving objects



Note

Improper use of the function can result in it no longer being possible to start the controller. Therefore, carefully read the entire chapter before using the function.

7.3.1 General

Many objects in the object dictionary can be saved and then automatically reloaded the next time the controller is switched on or reset. Furthermore, the saved values are also retained following a firmware update.

Only entire collections of objects (referred to in the following as *categories*) can be saved together; individual objects cannot be saved.

An object can be assigned one of the following *categories*:

- Communication: Parameters related to external interfaces, such as PDO configuration etc.
- Application: Parameters related to operating modes.
- User: Parameters that are written and read by the customer/user only and are ignored by the controller firmware.
- Movement: Parameters related to the motor and the sensors (BLDC/Stepper, *Closed/Open Loop...*). Some are set and saved by auto setup.
- Tuning: Parameters related to motor and encoder that are set either by auto setup or that can be found in the data sheets, e.g., pole pairs and maximum current.

If an object is not assigned one of these *categories*, it cannot be saved, e.g., statusword and all objects whose value is dependent on the current state of the controller.

The objects in each *category* are listed below. In chapter **Description of the object dictionary**, the corresponding *category* for each object is also specified.

7.3.2 Category: communication

- **1600_h**: Receive PDO 1 Mapping Parameter
- **1601_h**: Receive PDO 2 Mapping Parameter
- **1602_h**: Receive PDO 3 Mapping Parameter
- **1603_h**: Receive PDO 4 Mapping Parameter
- **1A00_h**: Transmit PDO 1 Mapping Parameter
- **1A01_h**: Transmit PDO 2 Mapping Parameter
- **1A02_h**: Transmit PDO 3 Mapping Parameter
- **1A03_h**: Transmit PDO 4 Mapping Parameter
- **1C12_h**: Sync Manager PDO Assignment
- **1C13_h**: Sync Manager PDO Assignment
- **2102_h**: Fieldbus Module Control

7.3.3 Category: application

- **2034_h**: Upper Voltage Warning Level
- **2035_h**: Lower Voltage Warning Level
- **2036_h**: Open Loop Current Reduction Idle Time
- **2037_h**: Open Loop Current Reduction Value/factor
- **2038_h**: Brake Controller Timing
- **203A_h**: Homing On Block Configuration
- **203D_h**: Torque Window
- **203E_h**: Torque Window Time Out
- **203F_h**: Max Slippage Time Out
- **2056_h**: Limit Switch Tolerance Band
- **2057_h**: Clock Direction Multiplier
- **2058_h**: Clock Direction Divider
- **205B_h**: Clock Direction Or Clockwise/Counter Clockwise Mode
- **2084_h**: Bootup Delay
- **2300_h**: NanoJ Control
- **2410_h**: NanoJ Init Parameters
- **2800_h**: Bootloader And Reboot Settings
- **3210_h**: Motor Drive Parameter Set
- **3212_h**: Motor Drive Flags
- **3221_h**: Analogue Inputs Control
- **3240_h**: Digital Inputs Control
- **3241_h**: Digital Input Capture
- **3242_h**: Digital Input Routing
- **3243_h**: Digital Input Homing Capture
- **3250_h**: Digital Outputs Control
- **3252_h**: Digital Output Routing
- **3321_h**: Analogue Input Offset
- **3322_h**: Analogue Input Pre-scaling
- **3700_h**: Deviation Error Option Code
- **4013_h**: HW Configuration
- **4015_h**: Special Drive Modes
- **6040_h**: Controlword
- **6042_h**: VI Target Velocity
- **6046_h**: VI Velocity Min Max Amount
- **6048_h**: VI Velocity Acceleration
- **6049_h**: VI Velocity Deceleration
- **604A_h**: VI Velocity Quick Stop

- **604C_h**: VI Dimension Factor
- **605A_h**: Quick Stop Option Code
- **605B_h**: Shutdown Option Code
- **605C_h**: Disable Option Code
- **605D_h**: Halt Option Code
- **605E_h**: Fault Option Code
- **6060_h**: Modes Of Operation
- **6065_h**: Following Error Window
- **6066_h**: Following Error Time Out
- **6067_h**: Position Window
- **6068_h**: Position Window Time
- **606D_h**: Velocity Window
- **606E_h**: Velocity Window Time
- **6071_h**: Target Torque
- **6072_h**: Max Torque
- **607A_h**: Target Position
- **607B_h**: Position Range Limit
- **607C_h**: Home Offset
- **607D_h**: Software Position Limit
- **607E_h**: Polarity
- **607F_h**: Max Profile Velocity
- **6081_h**: Profile Velocity
- **6082_h**: End Velocity
- **6083_h**: Profile Acceleration
- **6084_h**: Profile Deceleration
- **6085_h**: Quick Stop Deceleration
- **6086_h**: Motion Profile Type
- **6087_h**: Torque Slope
- **6091_h**: Gear Ratio
- **6092_h**: Feed Constant
- **6096_h**: Velocity Factor
- **6097_h**: Acceleration Factor
- **6098_h**: Homing Method
- **6099_h**: Homing Speed
- **609A_h**: Homing Acceleration
- **60A2_h**: Jerk Factor
- **60A4_h**: Profile Jerk
- **60A8_h**: SI Unit Position
- **60A9_h**: SI Unit Velocity
- **60B0_h**: Position Offset
- **60B1_h**: Velocity Offset
- **60B2_h**: Torque Offset
- **60C1_h**: Interpolation Data Record
- **60C2_h**: Interpolation Time Period
- **60C4_h**: Interpolation Data Configuration
- **60C5_h**: Max Acceleration
- **60C6_h**: Max Deceleration
- **60E8_h**: Additional Gear Ratio - Motor Shaft Revolutions
- **60E9_h**: Additional Feed Constant - Feed
- **60ED_h**: Additional Gear Ratio - Driving Shaft Revolutions
- **60EE_h**: Additional Feed Constant - Driving Shaft Revolutions
- **60F2_h**: Positioning Option Code
- **60F8_h**: Max Slippage

- **60FE_h**: Digital Outputs
- **60FF_h**: Target Velocity

7.3.4 Category: customer

- **2701_h**: Customer Storage Area

7.3.5 Category: drive

- **3202_h**: Motor Drive Submode Select
- **3203_h**: Feedback Selection

7.3.6 Category: tuning

- **2030_h**: Pole Pair Count
- **2031_h**: Maximum Current
- **203B_h**: I2t Parameters
- **2059_h**: Encoder Configuration
- **33A0_h**: Feedback Incremental A/B/I 1
- **6075_h**: Motor Rated Current
- **6080_h**: Max Motor Speed
- **608F_h**: Position Encoder Resolution
- **6090_h**: Velocity Encoder Resolution
- **60E6_h**: Additional Position Encoder Resolution - Encoder Increments
- **60EB_h**: Additional Position Encoder Resolution - Motor Revolutions

7.3.7 Starting the save process



Note

- The motor must be at a standstill during the save process and may not be started up while saving.
- Saving may take a few seconds. Under no circumstances may you interrupt the voltage supply while saving. The state of the saved objects is otherwise undefined.
- Always wait until the controller has signaled that the save process has been successfully completed with the value "1" in the corresponding subindex in object **1010_h**.

There is a subindex in object **1010_h** for each *category*. To save all objects of this *category*, the value "65766173_h" must be written in the subindex.¹ The controller signals the end of the save process by overwriting the value with a "1".

The following table shows which subindex of object **1010_h** is responsible for which *category*.

| Subindex | Category |
|-----------------|---|
| 01 _h | All categories with the exception of 06 _h (Tuning) |
| 02 _h | Communication |
| 03 _h | Application |
| 04 _h | User |
| 05 _h | Movement |
| 06 _h | Tuning |

¹ This corresponds to the decimal of 1702257011_d or the ASCII string `save`.

7.3.8 Discarding the saved data

If all objects or one *category* of saved objects is to be deleted, value "64616F6C_h" must be written in object 1011_h.² The following subindices correspond to a *category* here:

| Subindex | Category |
|-----------------|---|
| 01 _h | All categories (reset to factory settings) with the exception of 06 _h (Tuning) |
| 02 _h | Communication |
| 03 _h | Application |
| 04 _h | User |
| 05 _h | Movement |
| 06 _h | Tuning |

The saved objects are subsequently discarded; the change does not take effect until after the controller is restarted. You can restart the controller by entering the value "746F6F62_h" in 2800_h:01_h.



Note

- Objects of *category* 06_h (Tuning) are determined by **Auto setup** and are not reset when resetting to factory settings with subindex 01_h (thereby making it unnecessary to again perform an auto setup). You can reset these objects with subindex 06_h.

7.3.9 Verifying the configuration

Object 1020_h can be used to verify the configuration. It acts as a modification marker similar to common text editors: as soon as a file is modified in the editor, a marker (usually an asterisk) is added.

The entries of object 1020_h can be written with a date and time and then saved together with all other savable objects with 1010_h:01.

The entries of 1020_h are reset to "0" as soon as a savable object (including 1010_h:0x_h except for 1010_h:01_h and 1020_h) is written.

The following sequence makes verification possible:

1. An external tool or master configures the controller.
2. The tool or master sets the value in object 1020_h.
3. The tool or master activates the saving of all objects 1010_h:01_h = 65766173_h. The date and time in object 1020_h are also saved.

After the controller is restarted, the master can check the value in 1020_h:01_h and 1020:01_h. If one of the values is "0", the object dictionary was changed after the saved values were loaded. If the date or time in 1020 does not correspond to the expected value, objects were probably saved with values other than those that were expected.

² This corresponds to the decimal of 1684107116_d or the ASCII string load.

8 EtherCAT

EtherCAT references: **www.ethercat.org**.

- *ETG.1000.1 - Overview*, Date: 03.01.2013, Version 1.0.3
- *ETG.1000.2 - Physical Layer service and protocol specification*, Date: 03.01.2013, Version 1.0.3
- *ETG.1000.3 - Data Link Layer service definition*, Date: 03.01.2013, Version 1.0.3
- *ETG.1000.4 - Data-link layer protocol specification*, Date: 03.01.2013, Version 1.0.3
- *ETG.1000.5 - Application layer service definition*, Date: 03.01.2013, Version 1.0.3
- *ETG.1000.6 - Application layer protocol specification*, Date: 03.01.2013, Version 1.0.3
- *ETG.1300 - Indicator and Labeling specification*, Date: 11.11.2014, Version 1.1.0.2

9 Programming with NanoJ

NanoJ is a programming language similar to C or C++. *NanoJ* is integrated in the *Plug & Drive Studio* software. You can find further information in document *Plug & Drive Studio: Quick Start Guide* at us.nanotec.com.

9.1 NanoJ program

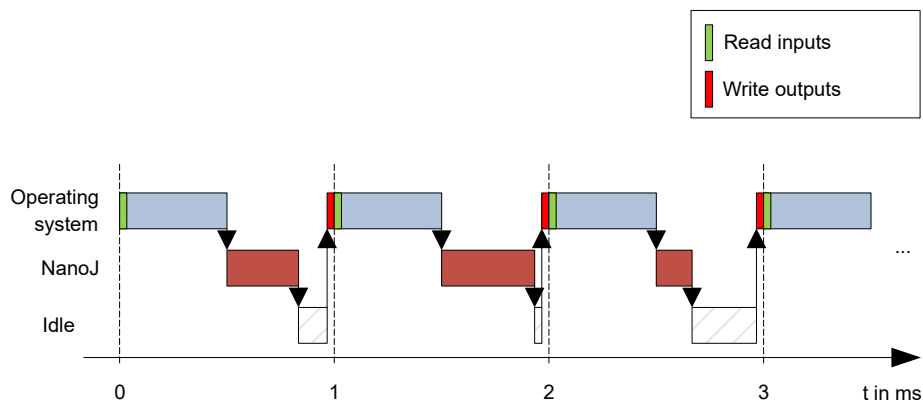
A *NanoJ program* makes a protected runtime environment available within the firmware. Here, the user can create his own processes. These can then trigger functions in the controller by, for example, reading or writing entries in the object dictionary.

Through the use of protective mechanisms, a *NanoJ program* is prevented from crashing the firmware. In the worst case, the execution is interrupted with an error code stored in the object dictionary.

If the *NanoJ program* was loaded on the controller, it is automatically executed after the controller is switched on or restarted.

9.1.1 Available computing time

A *NanoJ program* receives computing time cyclically in a 1 ms clock (see following figure). Because computing time is lost through interrupts and system functions of the firmware, only approx. 30% – 50% of computing time is available to the user program (depending on operating mode and application). In this time, the user program must run through the cycle and either complete the cycle or yield the computing time by calling the `yield()` function. In the former case, the user program is restarted with the start of the next 1 ms cycle; the latter results in the program being continued on the next 1 ms cycle with the command that follows the `yield()` function.



If the *NanoJ program* needs more time than was allotted, it is ended and an error code set in the object dictionary.



Tip

When developing user programs, the runtime behavior must be carefully examined, especially for more time-intensive tasks. For example, it is therefore recommended that tables be used instead of calculating a sine value using a `sin` function.



Note

If the *NanoJ* program does not yield the computing time after too long a time, it is ended by the operating system. In this case, the number 4 is entered in the statusword for object 2301_h; in the error register for object 2302_h, the number 5 (timeout) is noted, see **2301h NanoJ Status** and **2302h NanoJ Error Code**.

9.1.2 Sandbox

Using processor-specific features, a so-called *sandbox* is generated. When used in the sandbox, a user program can only access specially assigned memory areas and system resources. For example, an attempt to directly write to a processor IO register is acknowledged with an *MPU Fault* and the user program terminated with the corresponding error code in the object dictionary.

9.1.3 NanoJ program – communication possibilities

A *NanoJ* program has a number of possibilities for communicating with the controller:

- Read and write OD values using PDO mapping
- Directly read and write OD values using system calls
- Call other system calls (e.g., write debug output)

The OD values of the user program are made available in the form of variables via *PDO mapping*. Before a user program receives the 1 ms time slot, the firmware transfers the values from the object dictionary to the variables of the user program. As soon as the user program receives computing time, it can manipulate these variables as regular C variables. At the end of the time slot, the new values are then automatically copied by the firmware back to the respective OD entries.

To optimize the performance, three types of mapping are defined: input, output, and input/output (In, Out, InOut).

- *Input mappings* can only be read; they are not transferred back to the object dictionary.
- *Output mappings* can only be written.
- *Input/output mappings*, on the other hand, can both be read and written.

The set mappings can be read and checked via the GUI for objects 2310_h, 2320_h, and 2330_h. Up to 16 entries are allowed for each mapping.

Whether a variable is stored in the input, output or data range is controlled in *NanoJEasy* via the specification of the *linker section*.

9.1.4 Executing a NanoJ program

When executing a cycle, the *NanoJ* program essentially consists of the following three steps with respect to the PDO mapping:

1. Read values from the object dictionary and copy them to the input and output areas
2. Execute a user program
3. Copy values from the output and input areas back to the object dictionary

The configuration of the copy processes is based on the CANopen standard.

In addition, values of the object dictionary can be accessed via system calls. This is generally slower; mappings are therefore to be preferred. The number of mappings is limited (16 entries each in In/Out/InOut).



Tip

Nanotec recommends: Map OD entries that are used and changed frequently and use system calls to access OD entries that are used less frequently.

A list of available system calls can be found in chapter **System calls in a NanoJ program**.



Tip

Nanotec recommends accessing a given OD value either by mapping or using a system call with `od_write()`. If both are used simultaneously, the system call has no effect.

9.1.5 NanoJ program – OD entries

The *NanoJ program* is controlled and configured in object range 2300_h to 2330_h (see **2300h NanoJ Control**).

| OD-Index | Name and description |
|-------------------|---|
| 2300 _h | 2300h NanoJ Control |
| 2301 _h | 2301h NanoJ Status |
| 2302 _h | 2302h NanoJ Error Code |
| 2310 _h | 2310h NanoJ Input Data Selection |
| 2320 _h | 2320h NanoJ Output Data Selection |
| 2330 _h | 2330h NanoJ In/output Data Selection |

Example:

To select and start the *TEST1.USR* user program, the following sequence can, for example, be used:

- Check entry **2302_h** for error code.
- If no error:
Start the *NanoJ program* by writing object **2300_h**, bit 0 = "1".



Note

It can take up to 200 ms for the NanoJ program to start.

- Check entry **2302_h** for error code and object **2301_h**, bit 0 = "1".

To stop a running program: write entry **2300_h** with bit 0 value = "0".

9.1.6 Structure of a NanoJ program

A user program consists of at least two instructions:

- the preprocessor instruction `#include "wrapper.h"`
- the `void user() {}` function

The code to be executed can be stored in the `void user()` function.



Note

The file names of the user programs must not be longer than eight characters plus three characters in the suffix; file name `main.cpp` is permissible, file name `aLongFileName.cpp` is not permissible.



Note

In the *NanoJ program*, only global variables are permitted and they may only be initialized within code. It then follows:

- No `new` operator
- No constructors
- No initialization of global variables outside of code

Examples:

The global variable is to be initialized within the `void user()` function:

```
unsigned int i;  
void user(){  
    i = 1;  
    i += 1;  
}
```

The following assignment is not correct:

```
unsigned int i = 1;  
void user() {  
    i += 1;  
}
```

9.1.7 NanoJ program example

The example shows the programming of a square wave signal in object `2500h:01h`.

```
// file main.cpp  
map S32 outputReg1 as inout 0x2500:1  
#include "wrapper.h"  
  
// user program  
void user()  
{  
    U16 counter = 0;  
    while( 1 )  
    {  
        ++counter;  
  
        if( counter < 100 )  
            InOut.outputReg1 = 0;  
        else if( counter < 200 )  
            InOut.outputReg1 = 1;  
        else  
            counter = 0;  
  
        // yield() 5 times (delay 5ms)  
        for(U08 i = 0; i < 5; ++i )  
            yield();  
    }  
} // eof
```

You can find other examples at us.nanotec.com.

9.2 Mapping in the NanoJ program

With this method, a variable in the *NanoJ* program is linked directly with an entry in the object dictionary. The creation of the mapping must be located at the start of the file here, even before the `#include "wrapper.h"` instruction. A comment is permitted above the mapping.



Tip

Nanotec recommends:

- Use mapping if you need to access an object in the object dictionary frequently, e.g., *controlword* 6040_h or *statusword* 6041_h.
- The `od_write()` and `od_read()` functions are better suited for accessing objects a single time, see **Accessing the object dictionary**.

9.2.1 Declaration of the mapping

The declaration of the mapping is structured as follows:

```
map <TYPE> <NAME> as <input|output|inout> <INDEX>:<SUBINDEX>
```

Where:

- `<TYPE>`
The data type of the variable; U32, U16, U08, S32, S16 or S08.
- `<NAME>`
The name of the variable as it is used in the user program.
- `<input|output|inout>`
The read and write permission of a variable: a variable can be declared as an *input*, *output* or *inout*. This defines whether a variable is readable (*input*), writable (*output*) or both (*inout*) and the structure by means of which it must be addressed in the program.
- `<INDEX>:<SUBINDEX>`

Index and subindex of the object to be mapped in the object dictionary.

Each declared variable is addressed in the user program via one of the three structures: *In*, *Out* or *InOut* depending on the defined write and read direction.

9.2.2 Example of mapping

Example of a mapping and the corresponding variable accesses:

```
map U16 controlWord as output 0x6040:00
map U08 statusWord as input 0x6041:00
map U08 modeOfOperation as inout 0x6060:00

#include "wrapper.h"

void user()
{
    [...]
    Out.controlWord = 1;
    U08 tmpVar = In.statusword;
    InOut.modeOfOperation = tmpVar;
    [...]
}
```

9.2.3 Possible error at `od_write()`

A possible source of errors is a write access with the `od_write()` function (see **System calls in a NanoJ program**) of an object in the object dictionary that was simultaneously created as mapping. The code listed in the following is incorrect:

```
map U16 controlWord as output 0x6040:00
#include " wrapper.h"
void user()
{
  [...]
  Out.controlWord = 1;
  [...]
  od_write(0x6040, 0x00, 5 ); // der Wert wird durch das Mapping
  überschrieben
  [...]
}
```

The line with the `od_write(0x6040, 0x00, 5);` command has no effect. As described in the introduction, all mappings are copied to the object dictionary at the end of each millisecond.

This results in the following sequence:

1. The `od_write` function writes the value 5 in object 6040_h:00_h.
2. At the end of the 1 ms cycle, the mapping is written that also specifies object 6040_h:00_h, however, with the value 1.
3. From the perspective of the user, the `od_write` command thus serves no purpose.

9.3 System calls in a NanoJ program

With system calls, it is possible to call up functions integrated in the firmware directly from a user program. Because direct code execution is only possible in the protected area of the sandbox, this is implemented via so-called *Cortex-Supervisor-Calls* (Svc Calls). An interrupt is triggered when the function is called. The firmware thus has the possibility of temporarily allowing code execution outside of the sandbox. Developers of user programs do not need to worry about this mechanism – for them, the system calls can be called up like normal C functions. Only the *wrapper.h* file needs to be integrated as usual.

9.3.1 Accessing the object dictionary

void **od_write** (U32 index, U32 subindex, U32 value)

This function writes the transferred value to the specified location in the object dictionary.

| | |
|----------|---|
| index | Index of the object to be written in the object dictionary |
| subindex | Subindex of the object to be written in the object dictionary |
| value | Value to be written |



Note

It is highly recommended that the processor time be passed on with `yield()` after calling a `od_write()`. The value is immediately written to the OD. For the firmware to be able to trigger actions that are dependent on this, however, it must receive computing time. This, in turn, means that the user program must either be ended or interrupted with `yield()`.

U32 **od_read** (U32 index, U32 subindex)

This function reads the value at the specified location in the object dictionary and returns it.

| | |
|--------------|--|
| index | Index of the object to be read in the object dictionary |
| subindex | Subindex of the object to be read in the object dictionary |
| Output value | Content of the OD entry |



Note

Active waiting for a value in the object dictionary should always be associated with a `yield()`.

Example

```
while (od_read(2400,2) != 0) // wait until 2400:2 is set
{ yield(); }
```

9.3.2 Process control

```
void yield()
```

This function returns the processor time to the operating system. In the next time slot, the program continues at the location after the call.

```
void sleep (U32 ms)
```

This function returns the processor time to the operating system for the specified number of milliseconds. The user program is then continued at the location after the call.

| | |
|----|-----------------------------------|
| ms | Time to be waited in milliseconds |
|----|-----------------------------------|

10 Description of the object dictionary

10.1 Overview

This chapter contains a description of all objects.

You will find information here on:

- Functions
- Object descriptions ("Index")
- Value descriptions ("Subindices")
- Descriptions of bits
- Description of the object

10.2 Structure of the object description

The description of the object entries always has the same structure and usually consists of the following sections:

Function

The function of the object dictionary is briefly described in this section.

Object description

This table provides detailed information on the data type, preset values and similar. An exact description can be found in section "**Object description**"

Value description

This table is only available with the "Array" or "Record" data type and provides exact information about the sub-entries. A more exact description of the entries can be found in section "**Value description**"

Description

Here, more exact information on the individual bits of an entry is provided or any compositions explained. A more exact description can be found in section "**Description**"

10.3 Object description

The object description consists of a table that contains the following entries:

Index

Designates the object index in hexadecimal notation.

Object name

The name of the object.

Object Code

The type of object. This can be one of the following entries:

- VARIABLE: In this case, the object consists of only a variable that is indexed with subindex 0.
- ARRAY: These objects always consists of a subindex 0 – which specifies the number of sub-entries – and the sub-entries themselves, beginning with index 1. The data type within an array never changes, i.e., sub-entry 1 and all subsequent entries are always of the same data type.
- RECORD: These objects always consists of a subindex 0 – which specifies the number of sub-entries – and the sub-entries themselves, beginning with index 1. Unlike an ARRAY, the data type of the sub-entries can vary. This means that, e.g., sub-entry 1 may be of a different data type than sub-entry 2.

- **VISIBLE_STRING**: The object describes a character string coded in ASCII. The length of the string is specified in subindex 0; the individual characters are stored beginning in subindex 1. These character strings are **not** terminated by a null character.

Data type

The size and interpretation of the object is specified here. The following notation is used for the "VARIABLE" object code:

- A distinction is made between entries that are signed; these are designated with the prefix "SIGNED". For entries that are unsigned, the prefix "UNSIGNED" is used.
- The size of the variable in bits is placed before the prefix and can be 8, 16 or 32.

Savable

Described here is whether this object is savable and, if so, in which category.

Firmware version

The firmware version beginning with which the object is available is entered here.

Change history (ChangeLog)

Any changes to the object are noted here.

There are also the following table entries for the "VARIABLE" data type:

Access

The access restriction is entered here. The following restrictions are available:

- "read/write": The object can both be read as well as written
- "read only": The object can only be read from the object dictionary. It is not possible to set a value.

PDO mapping

Some bus systems, such as CANopen or EtherCAT, support PDO mapping. Described in this table entry is whether the object can be inserted into a mapping and, if so, into which. The following designations are available here:

- "no": The object may not be entered in a mapping.
- "TX-PDO": The object may be entered in an RX mapping.
- "RX-PDO": The object may be entered in a TX mapping.

Allowed values

In some cases, only certain values may be written in the object. If this is the case, these values are listed here. If there are no restrictions, the field is empty.

Preset value

To bring the controller to a secured state when switching on, it is necessary to preset a number of objects with values. The value that is written in the object when the controller is started is noted in this table entry.

10.4 Value description



Note

For the sake of clarity, a number of subindices are grouped together if the entries all have the same name.

Listed in the table with the "Value description" heading are all data for sub-entries with subindex 1 or higher. The table contains the following entries:

Subindex

Number of the currently written sub-entry.

Name

Name of the sub-entry.

Data type

The size and interpretation of the sub-entry is specified here. The following notation always applies here:

- A distinction is made between entries that are signed; these are designated with the prefix "SIGNED". For entries that are unsigned, the prefix "UNSIGNED" is used.
- The size of the variable in bits is placed before the prefix and can be 8, 16 or 32.

Access

The access restriction for the sub-entry is entered here. The following restrictions are available:

- "read/write": The object can both be read as well as written
- "read only": The object can only be read from the object dictionary. It is not possible to set a value.

PDO mapping

Some bus systems, such as CANopen or EtherCAT, support PDO mapping. Described in this table entry is whether the sub-entry can be inserted into a mapping and, if so, into which. The following designations are available here:

- "no": The object may not be entered in a mapping.
- "TX-PDO": The object may be entered in an RX mapping.
- "RX-PDO": The object may be entered in a TX mapping.

Allowed values

In some cases, only certain values may be written in the sub-entry. If this is the case, these values are listed here. If there are no restrictions, the field is empty.

Preset value

To bring the controller to a secured state when switching on, it is necessary to preset a number of sub-entries with values. The value that is written in the sub-entry when the controller is started is noted in this table entry.

10.5 Description

This section may be present if use requires additional information. If individual bits of an object or sub-entry have different meaning, diagrams as shown in the following example are used.

Example: The object is 8 bits in size; bit 0 and bit 1 have different functions. Bits 2 and 3 are grouped into one function; the same applies for bits 4 to 7.

| | | | | | | | |
|-------------|---|---|---|-------------|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Example [4] | | | | Example [2] | | B | A |

Example [4]

Description of bit 4 up to and including bit 7; these bits are logically related. The 4 in square brackets specifies the number of related bits. A list with possible values and their description is often attached at this point.

Example [2]

Description of bits 3 and 2; these bits are logically related. The 2 in square brackets specifies the number of related bits.

- Value 00_b: The description here applies if bit 2 and bit 3 are "0".
- Value 01_b: The description here applies if bit 2 is "0" and bit 3 is "1".
- Value 10_b: The description here applies if bit 2 is "1" and bit 3 is "0".
- Value 11_b: The description here applies if bit 2 and bit 3 are "1".

B

Description of bit B; no length is specified for a single bit.

A

Description of bit A; bits with a gray background are not used.

1000h Device Type

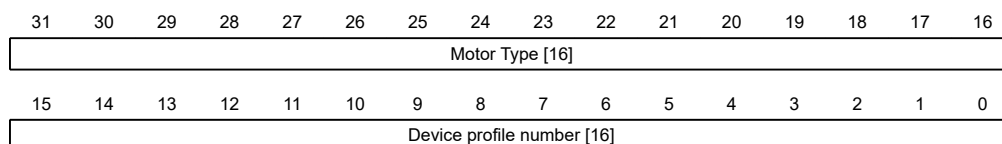
Function

Describes the controller type.

Object description

| | |
|------------------|---|
| Index | 1000 _h |
| Object name | Device Type |
| Object Code | VARIABLE |
| Data type | UNSIGNED32 |
| Savable | no |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | <ul style="list-style-type: none"> • PD4-E591L42-E-65-1: 00040192_h • PD4-E601L42-E-65-1: 00040192_h • PD4-EB59CD-E-65-1: 00020192_h |
| Firmware version | FIR-v1426 |
| Change history | |

Description



Motor Type[16]

Describes the supported motor type. The following values are possible:

- Bit 23 to bit 16: Value "1": Servo drive
- Bit 23 to bit 16: Value "2": Stepper motor

Device profile number[16]

Describes the supported CANopen standard.

Values:

0192_h or 0402_d (preset value): The CiA 402 standard is supported.

1001h Error Register

Function

Error register: The corresponding error bit is set in case of an error. If the error no longer exists, it is deleted automatically.

Object description

| | |
|------------------|-------------------|
| Index | 1001 _h |
| Object name | Error Register |
| Object Code | VARIABLE |
| Data type | UNSIGNED8 |
| Savable | no |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 00 _h |
| Firmware version | FIR-v1426 |
| Change history | |

Description

| | | | | | | | |
|-----|-----|------|-----|------|-----|-----|-----|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MAN | RES | PROF | COM | TEMP | VOL | CUR | GEN |

GEN

General error

CUR

Current

VOL

Voltage

TEMP

Temperature

COM

Communication

PROF

Relates to the device profile

RES

Reserved, always "0"

MAN

Manufacturer-specific: The motor turns in the wrong direction.

1003h Pre-defined Error Field

Function

This object contains an error stack with up to eight entries.

Object description

| | |
|------------------|-------------------------|
| Index | 1003 _h |
| Object name | Pre-defined Error Field |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | no |
| Firmware version | FIR-v1426 |
| Change history | |

Value description

| | |
|----------------|------------------|
| Subindex | 00 _h |
| Name | Number Of Errors |
| Data type | UNSIGNED8 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00 _h |

| | |
|----------------|-----------------------|
| Subindex | 01 _h |
| Name | Standard Error Field |
| Data type | UNSIGNED32 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

| | |
|----------------|----------------------|
| Subindex | 02 _h |
| Name | Standard Error Field |
| Data type | UNSIGNED32 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |

| | |
|----------------|-----------------------|
| Preset value | 00000000 _h |
| Subindex | 03 _h |
| Name | Standard Error Field |
| Data type | UNSIGNED32 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| Subindex | 04 _h |
| Name | Standard Error Field |
| Data type | UNSIGNED32 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| Subindex | 05 _h |
| Name | Standard Error Field |
| Data type | UNSIGNED32 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| Subindex | 06 _h |
| Name | Standard Error Field |
| Data type | UNSIGNED32 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| Subindex | 07 _h |
| Name | Standard Error Field |
| Data type | UNSIGNED32 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| Subindex | 08 _h |

| | |
|----------------|-----------------------|
| Name | Standard Error Field |
| Data type | UNSIGNED32 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

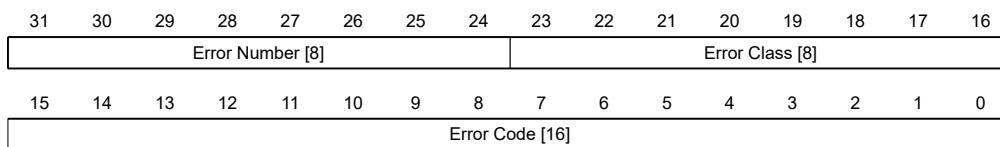
Description

General function

If a new error occurs, it is entered in subindex 1. The already existing entries in subindices 1 to 7 are moved back one position. The error in subindex 7 is thereby removed.

The number of errors that have already occurred can be read from the object with subindex 0. If no error is currently entered in the error stack, it is not possible to read one of the eight subindices 1–8 and an error (abort code = 08000024_h) is sent in response. If a "0" is written in subindex 0, counting starts again from the beginning.

Bit description



Error Number [8]

This can be used to pinpoint the cause of the error. The meaning of the number can be found in the following table.

| Error number | Description |
|--------------|---|
| 0 | Watchdog-Reset |
| 1 | Input voltage too high |
| 2 | Output current too high |
| 3 | Input voltage too low |
| 4 | Error at fieldbus |
| 5 | Motor turns – in spite of active block – in the wrong direction |
| 6 | CANopen only: NMT master takes too long to send nodeguarding request |
| 7 | Encoder error due to electrical fault or defective hardware |
| 8 | Encoder error; index not found during the auto setup |
| 9 | Error in the AB track |
| 10 | Positive limit switch and tolerance zone exceeded |
| 11 | Negative limit switch and tolerance zone exceeded |
| 12 | Device temperature above 80°C |
| 13 | The values of object 6065_h (Following Error Window) and object 6066_h (Following Error Time Out) were exceeded; a fault was triggered. |
| 14 | Warning: Nonvolatile memory full; controller must be restarted for cleanup work. |
| 15 | Motor blocked |
| 16 | Warning: Nonvolatile memory damaged; controller must be restarted for cleanup work. |

| Error number | Description |
|--------------|--|
| 17 | CANopen only: Slave took too long to send PDO messages. |
| 18 | Hall sensor faulty |
| 19 | CANopen only: PDO not processed due to a length error |
| 20 | CANopen only: PDO length exceeded |
| 21 | Warning: Nonvolatile memory full; controller must be restarted for cleanup work. |
| 22 | Rated current must be set (203B _h :01 _h) |
| 23 | Encoder resolution, number of pole pairs and some other values are incorrect. |
| 24 | Motor current is too high, adjust the PI parameters. |
| 25 | Internal software error, generic |
| 26 | Current too high at digital output |
| 27 | CANopen only: Unexpected sync length |
| 28 | EtherCAT only: The motor was stopped because EtherCAT switched state from OP to either SafeOP or PreOP without first stopping the motor. |
| 30 | Error in speed monitoring: slippage error too large |

Error Class[8]

This byte is identical to object **1001_h**

Error Code[16]

Refer to the following table for the meaning of the bytes.

| Error Code | Description |
|-------------------|--|
| 1000 _h | General error |
| 2300 _h | Current at the controller output too large |
| 3100 _h | Overvoltage/undervoltage at controller input |
| 4200 _h | Temperature error within the controller |
| 6010 _h | Software reset (watchdog) |
| 6100 _h | Internal software error, generic |
| 6320 _h | Rated current must be set (203B _h :01 _h) |
| 7121 _h | Motor blocked |
| 7305 _h | Incremental encoder or Hall sensor faulty |
| 7600 _h | Warning: Nonvolatile memory full or corrupt; restart the controller for cleanup work |
| 8000 _h | Error during fieldbus monitoring |
| 8130 _h | CANopen only: "Life Guard" error or "Heartbeat" error |
| 8200 _h | CANopen only: Slave took too long to send PDO messages. |
| 8210 _h | CANopen only: PDO was not processed due to a length error |
| 8220 _h | CANopen only: PDO length exceeded |
| 8240 _h | CANopen only: Unexpected sync length |
| 8400 _h | Error in speed monitoring: slippage error too large |
| 8611 _h | Position monitoring error: Following error too large |
| 8612 _h | Position monitoring error: Limit switch and tolerance zone exceeded |

| Error Code | Description |
|-------------------|--|
| 9000 _h | EtherCAT only: The motor was stopped because EtherCAT switched state from OP to either SafeOP or PreOP without first stopping the motor. |

1008h Manufacturer Device Name

Function

Contains the device name as character string.

Object description

| | |
|------------------|--|
| Index | 1008 _h |
| Object name | Manufacturer Device Name |
| Object Code | VARIABLE |
| Data type | VISIBLE_STRING |
| Savable | no |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | <ul style="list-style-type: none"> • PD4-E591L42-E-65-1: PD4-E591L42-E-65-1 • PD4-E601L42-E-65-1: PD4-E601L42-E-65-1 • PD4-EB59CD-E-65-1: PD4-EB59CD-E-65-1 |
| Firmware version | FIR-v1426 |
| Change history | |

1009h Manufacturer Hardware Version

Function

This object contains the hardware version as character string.

Object description

| | |
|------------------|-------------------------------|
| Index | 1009 _h |
| Object name | Manufacturer Hardware Version |
| Object Code | VARIABLE |
| Data type | VISIBLE_STRING |
| Savable | no |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 0 |
| Firmware version | FIR-v1426 |
| Change history | |

100Ah Manufacturer Software Version

Function

This object contains the software version as character string.

Object description

| | |
|------------------|-------------------------------|
| Index | 100A _h |
| Object name | Manufacturer Software Version |
| Object Code | VARIABLE |
| Data type | VISIBLE_STRING |
| Savable | no |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | FIR-v1748-B538662 |
| Firmware version | FIR-v1426 |
| Change history | |

1010h Store Parameters

Function

This object is used to start the saving of objects. See chapter **Saving objects**.

Object description

| | |
|------------------|---|
| Index | 1010 _h |
| Object name | Store Parameters |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | no |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1426 |
| Change history | <p>Firmware version FIR-v1436: "Object name" entry changed from "Store Parameter" to "Store Parameters".</p> <p>Firmware version FIR-v1436: The number of entries was changed from 3 to 4.</p> <p>Firmware version FIR-v1512: The number of entries was changed from 4 to 5.</p> <p>Firmware version FIR-v1540: The number of entries was changed from 5 to 7.</p> <p>Firmware version FIR-v1738-B501312: The number of entries was changed from 7 to 14.</p> |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 0D _h |

| | |
|----------------|--|
| Subindex | 01 _h |
| Name | Save All Parameters To Non-volatile Memory |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000001 _h |

| | |
|----------------|--|
| Subindex | 02 _h |
| Name | Save Communication Parameters To Non-volatile Memory |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000001 _h |

| | |
|----------------|--|
| Subindex | 03 _h |
| Name | Save Application Parameters To Non-volatile Memory |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000001 _h |

| | |
|----------------|---|
| Subindex | 04 _h |
| Name | Save Customer Parameters To Non-volatile Memory |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000001 _h |

| | |
|----------|-----------------|
| Subindex | 05 _h |
|----------|-----------------|

| | |
|----------------|--|
| Name | Save Drive Parameters To Non-volatile Memory |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000001 _h |
| <hr/> | |
| Subindex | 06 _h |
| Name | Save Tuning Parameters To Non-volatile Memory |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000001 _h |
| <hr/> | |
| Subindex | 07 _h |
| Name | Save Miscellaneous Configurations To Non-volatile Memory |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000001 _h |
| <hr/> | |
| Subindex | 08 _h |
| Name | Save Reserved1 Configurations To Non-volatile Memory |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 09 _h |
| Name | Save Reserved2 Configurations To Non-volatile Memory |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 0A _h |
| Name | Save CANopen Configurations To Non-volatile Memory |
| Data type | UNSIGNED32 |
| Access | read / write |

| | |
|----------------|---|
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000001 _h |
| <hr/> | |
| Subindex | 0B _h |
| Name | Save Modbus RTU Configurations To Non-volatile Memory |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000001 _h |
| <hr/> | |
| Subindex | 0C _h |
| Name | Save Ethernet Configurations To Non-volatile Memory |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000001 _h |
| <hr/> | |
| Subindex | 0D _h |
| Name | Save Profibus Configurations To Non-volatile Memory |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000001 _h |

Description

Each subindex of the object stands for a certain memory class. By reading out the entry, it is possible to determine whether (value "1") or not (value="0") this memory category can be saved.

To start the save process of a memory category, value "65766173_h" must be written in the corresponding subindex. This corresponds to the decimal of 1702257011_d or the ASCII string `save`. As soon as the saving process is completed, the save command is again overwritten with the value "1", since saving is possible again.

For a detailed description, see chapter **Saving objects**.

1011h Restore Default Parameters

Function

This object can be used to reset all or part of the object dictionary to the default values. See chapter **Saving objects**.

Object description

| | |
|------------------|--|
| Index | 1011 _h |
| Object name | Restore Default Parameters |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | no |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1426 |
| Change history | <p>Firmware version FIR-v1436: "Object name" entry changed from "Restore Default Parameter" to "Restore Default Parameters".</p> <p>Firmware version FIR-v1436: The number of entries was changed from 2 to 4.</p> <p>Firmware version FIR-v1512: The number of entries was changed from 4 to 5.</p> <p>Firmware version FIR-v1512: "Name" entry changed from "Restore The Comm Default Parameters" to "Restore Communication Default Parameters".</p> <p>Firmware version FIR-v1512: "Name" entry changed from "Restore The Application Default Parameters" to "Restore Application Default Parameters".</p> <p>Firmware version FIR-v1540: The number of entries was changed from 5 to 7.</p> <p>Firmware version FIR-v1738-B501312: The number of entries was changed from 7 to 14.</p> |

Value description

| | |
|----------------|--------------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 0D _h |
| Subindex | 01 _h |
| Name | Restore All Default Parameters |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000001 _h |

| | |
|----------------|--|
| Subindex | 02 _h |
| Name | Restore Communication Default Parameters |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000001 _h |
| Subindex | 03 _h |
| Name | Restore Application Default Parameters |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000001 _h |
| Subindex | 04 _h |
| Name | Restore Customer Default Parameters |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000001 _h |
| Subindex | 05 _h |
| Name | Restore Drive Default Parameters |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000001 _h |
| Subindex | 06 _h |
| Name | Restore Tuning Default Parameters |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| Subindex | 07 _h |
| Name | Restore Miscellaneous Configurations |
| Data type | UNSIGNED32 |

| | |
|----------------|--|
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 08 _h |
| Name | Restore Reserved1 Configurations To Non-volatile Memory |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 09 _h |
| Name | Restore Reserved2 Configurations To Non-volatile Memory |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 0A _h |
| Name | Restore CANopen Configurations To Non-volatile Memory |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000001 _h |
| <hr/> | |
| Subindex | 0B _h |
| Name | Restore Modbus RTU Configurations To Non-volatile Memory |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000001 _h |
| <hr/> | |
| Subindex | 0C _h |
| Name | Restore Ethernet Configurations To Non-volatile Memory |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |

| | |
|----------------|--|
| Preset value | 00000001 _h |
| Subindex | 0D _h |
| Name | Restore Profibus Configurations To Non-volatile Memory |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000001 _h |

Description

If the value 64616F6C_h (or 1684107116_d or ASCII `load`) is written in this object, part or all of the object dictionary is reset to the default values. The subindex that is used decides which range is reset.

For a detailed description, see chapter **Discarding the saved data**.

1018h Identity Object

Function

This object returns general information on the device, such as manufacturer, product code, revision and serial number.



Tip

Have these values ready in the event of service inquiries.

Object description

| | |
|------------------|-------------------|
| Index | 1018 _h |
| Object name | Identity Object |
| Object Code | RECORD |
| Data type | IDENTITY |
| Savable | no |
| Firmware version | FIR-v1426 |
| Change history | |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 04 _h |

| | |
|----------------|---|
| Subindex | 01 _h |
| Name | Vendor-ID |
| Data type | UNSIGNED32 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 0000026C _h |
| <hr/> | |
| Subindex | 02 _h |
| Name | Product Code |
| Data type | UNSIGNED32 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | <ul style="list-style-type: none"> • PD4-E591L42-E-65-1: 00000033_h • PD4-E601L42-E-65-1: 0000003C_h • PD4-EB59CD-E-65-1: 00000045_h |
| <hr/> | |
| Subindex | 03 _h |
| Name | Revision Number |
| Data type | UNSIGNED32 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 06D40000 _h |
| <hr/> | |
| Subindex | 04 _h |
| Name | Serial Number |
| Data type | UNSIGNED32 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

1020h Verify Configuration

Function

This object indicates the date and time that the configuration was stored.

A configuration tool or a master can use this object to verify the configuration after a reset and, if necessary, perform a new configuration.

The tool must set the date and time before the storage mechanism is started (see chapter **Saving objects**).

Object description

| | |
|------------------|-----------------------|
| Index | 1020 _h |
| Object name | Verify Configuration |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | yes, category: verify |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1540 |
| Change history | |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 02 _h |

| | |
|----------------|-----------------------|
| Subindex | 01 _h |
| Name | Configuration Date |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

| | |
|----------------|-----------------------|
| Subindex | 02 _h |
| Name | Configuration Time |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

Description

Subindex 01_h (configuration date) is to contain the number of days since 1 January 1984.

Subindex 02_h (configuration time) is to contain the number of milliseconds since midnight.

1600h Receive PDO 1 Mapping Parameter

Function

This object contains the mapping parameters for PDOs that the controller can receive (RX-PDO 1).

Object description

| | |
|------------------|--|
| Index | 1600 _h |
| Object name | Receive PDO 1 Mapping Parameter |
| Object Code | RECORD |
| Data type | PDO_MAPPING |
| Savable | yes, category: communication |
| Firmware version | FIR-v1426 |
| Change history | Firmware version FIR-v1426: "Heading" entry changed from "1600h Drive Control" to "1600h Receive PDO 1 Mapping Parameter". Firmware version FIR-v1426: "Object name" entry changed from "Drive Control" to "Receive PDO 1 Mapping Parameter". |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 04 _h |

| | |
|----------------|-------------------------|
| Subindex | 01 _h |
| Name | 1st Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 60400010 _h |

| | |
|----------------|-------------------------|
| Subindex | 02 _h |
| Name | 2nd Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 607A0020 _h |

| | |
|----------------|-------------------------|
| Subindex | 03 _h |
| Name | 3rd Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 32020020 _h |

| | |
|----------------|-------------------------|
| Subindex | 04 _h |
| Name | 4th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 60600008 _h |

| | |
|----------------|-------------------------|
| Subindex | 05 _h |
| Name | 5th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

| | |
|----------------|-------------------------|
| Subindex | 06 _h |
| Name | 6th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

| | |
|----------------|-------------------------|
| Subindex | 07 _h |
| Name | 7th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

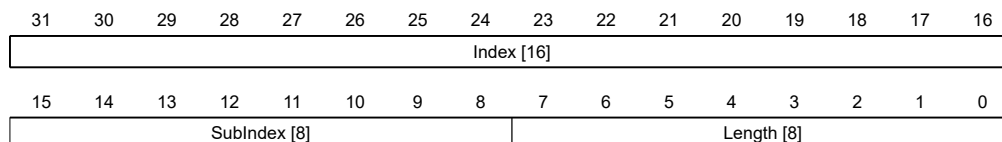
| | |
|-----------|-------------------------|
| Subindex | 08 _h |
| Name | 8th Object To Be Mapped |
| Data type | UNSIGNED32 |

| | |
|----------------|-----------------------|
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

Description

Each subindex (1–8) describes a different mapped object.

A mapping entry consists of four bytes, which are structured according to the following graphic.



Index [16]

This contains the index of the object to be mapped.

Subindex [8]

This contains the subindex of the object to be mapped.

Length [8]

This contains the length of the object to be mapped in units of bits.

1601h Receive PDO 2 Mapping Parameter

Function

This object contains the mapping parameters for PDOs that the controller can receive (RX-PDO 2).

Object description

| | |
|------------------|--|
| Index | 1601 _h |
| Object name | Receive PDO 2 Mapping Parameter |
| Object Code | RECORD |
| Data type | PDO_MAPPING |
| Savable | yes, category: communication |
| Firmware version | FIR-v1426 |
| Change history | Firmware version FIR-v1426: "Heading" entry changed from "1601h Positioning Control" to "1601h Receive PDO 2 Mapping Parameter". Firmware version FIR-v1426: "Object name" entry changed from "Positioning Control" to "Receive PDO 2 Mapping Parameter". |

Value description

| | |
|-----------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read / write |

| | |
|----------------|-------------------------|
| PDO mapping | no |
| Allowed values | |
| Preset value | 02 _h |
| <hr/> | |
| Subindex | 01 _h |
| Name | 1st Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 607A0020 _h |
| <hr/> | |
| Subindex | 02 _h |
| Name | 2nd Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 60810020 _h |
| <hr/> | |
| Subindex | 03 _h |
| Name | 3rd Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 04 _h |
| Name | 4th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 05 _h |
| Name | 5th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

| | |
|----------------|-------------------------|
| Subindex | 06 _h |
| Name | 6th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

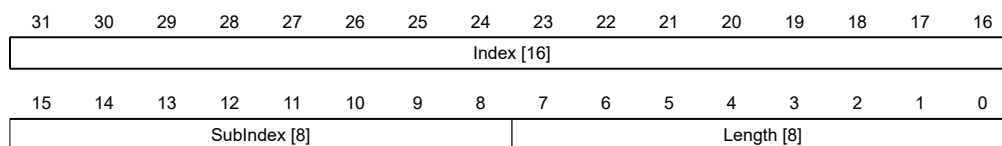
| | |
|----------------|-------------------------|
| Subindex | 07 _h |
| Name | 7th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

| | |
|----------------|-------------------------|
| Subindex | 08 _h |
| Name | 8th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

Description

Each subindex (1–8) describes a different mapped object.

A mapping entry consists of four bytes, which are structured according to the following graphic.



Index [16]

This contains the index of the object to be mapped.

Subindex [8]

This contains the subindex of the object to be mapped.

Length [8]

This contains the length of the object to be mapped in units of bits.

1602h Receive PDO 3 Mapping Parameter

Function

This object contains the mapping parameters for PDOs that the controller can receive (RX-PDO 3).

Object description

| | |
|------------------|--|
| Index | 1602 _h |
| Object name | Receive PDO 3 Mapping Parameter |
| Object Code | RECORD |
| Data type | PDO_MAPPING |
| Savable | yes, category: communication |
| Firmware version | FIR-v1426 |
| Change history | Firmware version FIR-v1426: "Heading" entry changed from "1602h Velocity Control" to "1602h Receive PDO 3 Mapping Parameter". Firmware version FIR-v1426: "Object name" entry changed from "Velocity Control" to "Receive PDO 3 Mapping Parameter". |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 01 _h |

| | |
|----------------|-------------------------|
| Subindex | 01 _h |
| Name | 1st Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 60420010 _h |

| | |
|----------------|-------------------------|
| Subindex | 02 _h |
| Name | 2nd Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

| | |
|-------------|-------------------------|
| Subindex | 03 _h |
| Name | 3rd Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |

| | |
|----------------|-------------------------|
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 04 _h |
| Name | 4th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 05 _h |
| Name | 5th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 06 _h |
| Name | 6th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 07 _h |
| Name | 7th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 08 _h |
| Name | 8th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

1603h Receive PDO 4 Mapping Parameter

Function

This object contains the mapping parameters for PDOs that the controller can receive (RX-PDO 4).

Object description

| | |
|------------------|--|
| Index | 1603 _h |
| Object name | Receive PDO 4 Mapping Parameter |
| Object Code | RECORD |
| Data type | PDO_MAPPING |
| Savable | yes, category: communication |
| Firmware version | FIR-v1426 |
| Change history | Firmware version FIR-v1426: "Heading" entry changed from "1603h Output Control" to "1603h Receive PDO 4 Mapping Parameter". Firmware version FIR-v1426: "Object name" entry changed from "Output Control" to "Receive PDO 4 Mapping Parameter". |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 01 _h |
| Subindex | 01 _h |
| Name | 1st Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 60420010 _h |
| Subindex | 02 _h |
| Name | 2nd Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

| | |
|----------------|-------------------------|
| Subindex | 03 _h |
| Name | 3rd Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

| | |
|----------------|-------------------------|
| Subindex | 04 _h |
| Name | 4th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

| | |
|----------------|-------------------------|
| Subindex | 05 _h |
| Name | 5th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

| | |
|----------------|-------------------------|
| Subindex | 06 _h |
| Name | 6th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

| | |
|----------------|-------------------------|
| Subindex | 07 _h |
| Name | 7th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

| | |
|-----------|-------------------------|
| Subindex | 08 _h |
| Name | 8th Object To Be Mapped |
| Data type | UNSIGNED32 |

| | |
|----------------|-----------------------|
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

1A00h Transmit PDO 1 Mapping Parameter

Function

This object contains the mapping parameters for PDOs that the controller can send (TX-PDO 1).

Object description

| | |
|------------------|--|
| Index | 1A00 _h |
| Object name | Transmit PDO 1 Mapping Parameter |
| Object Code | RECORD |
| Data type | PDO_MAPPING |
| Savable | yes, category: communication |
| Firmware version | FIR-v1426 |
| Change history | Firmware version FIR-v1426: "Heading" entry changed from "1A00h Drive Status" to "1A00h Transmit PDO 1 Mapping Parameter". Firmware version FIR-v1426: "Object name" entry changed from "Drive Status" to "Transmit PDO 1 Mapping Parameter". |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 03 _h |

| | |
|----------------|-------------------------|
| Subindex | 01 _h |
| Name | 1st Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 60410010 _h |

| | |
|-----------|-------------------------|
| Subindex | 02 _h |
| Name | 2nd Object To Be Mapped |
| Data type | UNSIGNED32 |

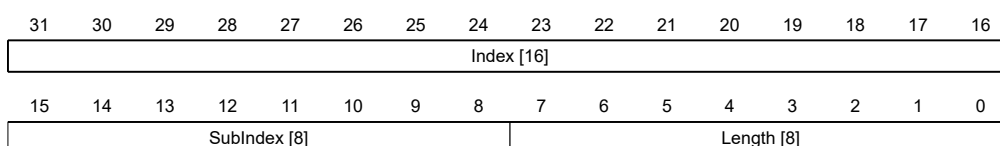
| | |
|----------------|-------------------------|
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 60640020 _h |
| <hr/> | |
| Subindex | 03 _h |
| Name | 3rd Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 60610008 _h |
| <hr/> | |
| Subindex | 04 _h |
| Name | 4th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 05 _h |
| Name | 5th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 06 _h |
| Name | 6th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 07 _h |
| Name | 7th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |

| | |
|----------------|-------------------------|
| Preset value | 00000000 _h |
| Subindex | 08 _h |
| Name | 8th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

Description

Each subindex (1–8) describes a different mapped object.

A mapping entry consists of four bytes, which are structured according to the following graphic.



Index [16]

This contains the index of the object to be mapped.

Subindex [8]

This contains the subindex of the object to be mapped.

Length [8]

This contains the length of the object to be mapped in units of bits.

1A01h Transmit PDO 2 Mapping Parameter

Function

This object contains the mapping parameters for PDOs that the controller can send (TX-PDO 2).

Object description

| | |
|------------------|--|
| Index | 1A01 _h |
| Object name | Transmit PDO 2 Mapping Parameter |
| Object Code | RECORD |
| Data type | PDO_MAPPING |
| Savable | yes, category: communication |
| Firmware version | FIR-v1426 |
| Change history | Firmware version FIR-v1426: "Heading" entry changed from "1A01h Positioning Status" to "1A01h Transmit PDO 2 Mapping Parameter". Firmware version FIR-v1426: "Object name" entry changed from "Positioning Status" to "Transmit PDO 2 Mapping Parameter". |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 01 _h |

| | |
|----------------|-------------------------|
| Subindex | 01 _h |
| Name | 1st Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 60640020 _h |

| | |
|----------------|-------------------------|
| Subindex | 02 _h |
| Name | 2nd Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

| | |
|----------------|-------------------------|
| Subindex | 03 _h |
| Name | 3rd Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

| | |
|----------------|-------------------------|
| Subindex | 04 _h |
| Name | 4th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

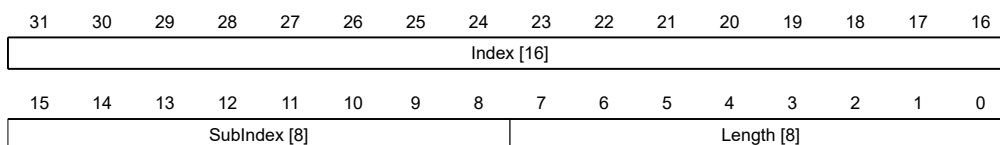
| | |
|----------|-----------------|
| Subindex | 05 _h |
|----------|-----------------|

| | |
|----------------|-------------------------|
| Name | 5th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 06 _h |
| Name | 6th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 07 _h |
| Name | 7th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 08 _h |
| Name | 8th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

Description

Each subindex (1–8) describes a different mapped object.

A mapping entry consists of four bytes, which are structured according to the following graphic.



Index [16]

This contains the index of the object to be mapped.

Subindex [8]

This contains the subindex of the object to be mapped.

Length [8]

This contains the length of the object to be mapped in units of bits.

1A02h Transmit PDO 3 Mapping Parameter

Function

This object contains the mapping parameters for PDOs that the controller can send (TX-PDO 3).

Object description

| | |
|------------------|--|
| Index | 1A02 _h |
| Object name | Transmit PDO 3 Mapping Parameter |
| Object Code | RECORD |
| Data type | PDO_MAPPING |
| Savable | yes, category: communication |
| Firmware version | FIR-v1426 |
| Change history | Firmware version FIR-v1426: "Heading" entry changed from "1A02h Velocity Status" to "1A02h Transmit PDO 3 Mapping Parameter". Firmware version FIR-v1426: "Object name" entry changed from "Velocity Status" to "Transmit PDO 3 Mapping Parameter". |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 01 _h |
| Subindex | 01 _h |
| Name | 1st Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 60440010 _h |
| Subindex | 02 _h |
| Name | 2nd Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |

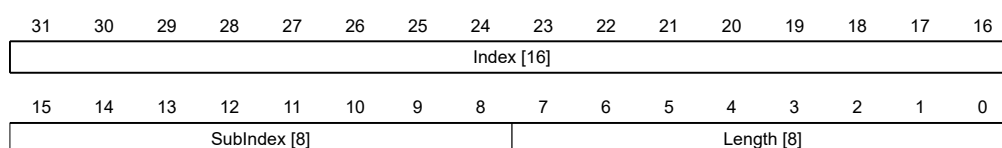
| | |
|----------------|-------------------------|
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 03 _h |
| Name | 3rd Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 04 _h |
| Name | 4th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 05 _h |
| Name | 5th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 06 _h |
| Name | 6th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 07 _h |
| Name | 7th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

| | |
|----------------|-------------------------|
| Subindex | 08 _h |
| Name | 8th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

Description

Each subindex (1–8) describes a different mapped object.

A mapping entry consists of four bytes, which are structured according to the following graphic.



Index [16]

This contains the index of the object to be mapped.

Subindex [8]

This contains the subindex of the object to be mapped.

Length [8]

This contains the length of the object to be mapped in units of bits.

1A03h Transmit PDO 4 Mapping Parameter

Function

This object contains the mapping parameters for PDOs that the controller can send (TX-PDO 4).

Object description

| | |
|------------------|--|
| Index | 1A03 _h |
| Object name | Transmit PDO 4 Mapping Parameter |
| Object Code | RECORD |
| Data type | PDO_MAPPING |
| Savable | yes, category: communication |
| Firmware version | FIR-v1426 |
| Change history | Firmware version FIR-v1426: "Heading" entry changed from "1A03h Input Status" to "1A03h Transmit PDO 4 Mapping Parameter". Firmware version FIR-v1426: "Object name" entry changed from "Input Status" to "Transmit PDO 4 Mapping Parameter". |

Value description

| | |
|----------|-----------------|
| Subindex | 00 _h |
|----------|-----------------|

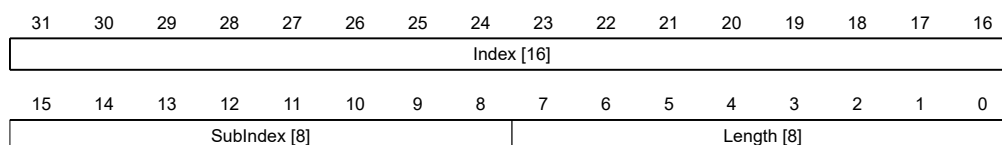
| | |
|----------------|-----------------------------|
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 03 _h |
| <hr/> | |
| Subindex | 01 _h |
| Name | 1st Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 60FD0020 _h |
| <hr/> | |
| Subindex | 02 _h |
| Name | 2nd Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 33200120 _h |
| <hr/> | |
| Subindex | 03 _h |
| Name | 3rd Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 33200220 _h |
| <hr/> | |
| Subindex | 04 _h |
| Name | 4th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 05 _h |
| Name | 5th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |

| | |
|----------------|-------------------------|
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 06 _h |
| Name | 6th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 07 _h |
| Name | 7th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 08 _h |
| Name | 8th Object To Be Mapped |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

Description

Each subindex (1–8) describes a different mapped object.

A mapping entry consists of four bytes, which are structured according to the following graphic.



Index [16]

This contains the index of the object to be mapped.

Subindex [8]

This contains the subindex of the object to be mapped.

Length [8]

This contains the length of the object to be mapped in units of bits.

1C00h Sync Manager Communication Type

Function

This object displays the assignment of all four EtherCAT SyncManager.

Object description

| | |
|------------------|---------------------------------|
| Index | 1C00 _h |
| Object name | Sync Manager Communication Type |
| Object Code | ARRAY |
| Data type | UNSIGNED8 |
| Savable | no |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1426 |
| Change history | |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 04 _h |

| | |
|----------------|---------------------------------|
| Subindex | 01 _h |
| Name | Sync Manager Communication Type |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 01 _h |

| | |
|----------------|---------------------------------|
| Subindex | 02 _h |
| Name | Sync Manager Communication Type |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 02 _h |

| | |
|----------------|---------------------------------|
| Subindex | 03 _h |
| Name | Sync Manager Communication Type |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 03 _h |

| | |
|----------------|---------------------------------|
| Subindex | 04 _h |
| Name | Sync Manager Communication Type |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 04 _h |

Description

The assignment of the SyncManager is defined by manufacturer and can't get changed.

| Subindex | Function |
|----------------------|---------------------------------|
| / Syncmanager | |
| 1 | Receive of mailbox messages |
| 2 | Transmit of mailbox messages |
| 3 | Receive of cyclic process data |
| 4 | Transmit of cyclic process data |

1C12h Sync Manager PDO Assignment

Function

This object lists all active output pdo mappings (see **1600_h** ff.) and will be written by the EtherCAT Master.

Object description

| | |
|------------------|------------------------------|
| Index | 1C12 _h |
| Object name | Sync Manager PDO Assignment |
| Object Code | ARRAY |
| Data type | UNSIGNED16 |
| Savable | yes, category: communication |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1426 |

Change history

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 01 _h |

| | |
|----------------|-------------------|
| Subindex | 01 _h |
| Name | PDO Mapping Index |
| Data type | UNSIGNED16 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 1600 _h |

| | |
|----------------|-------------------|
| Subindex | 02 _h |
| Name | PDO Mapping Index |
| Data type | UNSIGNED16 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 0000 _h |

| | |
|----------------|-------------------|
| Subindex | 03 _h |
| Name | PDO Mapping Index |
| Data type | UNSIGNED16 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 0000 _h |

| | |
|----------------|-------------------|
| Subindex | 04 _h |
| Name | PDO Mapping Index |
| Data type | UNSIGNED16 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 0000 _h |

1C13h Sync Manager PDO Assignment

Function

This object lists all active input pdo mappings (see **1A00_h** ff.) and will be written by the EtherCAT Master.

Object description

| | |
|------------------|------------------------------|
| Index | 1C13 _h |
| Object name | Sync Manager PDO Assignment |
| Object Code | ARRAY |
| Data type | UNSIGNED16 |
| Savable | yes, category: communication |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1426 |
| Change history | |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 01 _h |

| | |
|----------------|-------------------|
| Subindex | 01 _h |
| Name | PDO Mapping Index |
| Data type | UNSIGNED16 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 1A00 _h |

| | |
|----------------|-------------------|
| Subindex | 02 _h |
| Name | PDO Mapping Index |
| Data type | UNSIGNED16 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 0000 _h |

| | |
|----------------|-------------------|
| Subindex | 03 _h |
| Name | PDO Mapping Index |
| Data type | UNSIGNED16 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 0000 _h |

| | |
|----------------|-------------------|
| Subindex | 04 _h |
| Name | PDO Mapping Index |
| Data type | UNSIGNED16 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 0000 _h |

1C32h Output Sync Manager Synchronization

Function

Here all parameter for synchronization for output pdo mapping for EtherCAT are set (see **1C12_h**). These parameters will will be set by EtherCAT Master.

Object description

| | |
|------------------|---|
| Index | 1C32 _h |
| Object name | Output Sync Manager Synchronization |
| Object Code | RECORD |
| Data type | SYNCMGR_SYNCHRONIZATION |
| Savable | no |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1426 |
| Change history | <p>Firmware version FIR-v1626: Entry "Savable" modified from "No" to "yes, category: communication".</p> <p>Firmware version FIR-v1650-B472161: Entry "Savable" modified from "yes, category: communication" to "No".</p> <p>Firmware version FIR-v1650-B472161: Table entry "Access" at subindex 01 modified from "read / write" to "read only".</p> <p>Firmware version FIR-v1650-B472161: Table entry "Access" at subindex 02 modified from "read / write" to "read only".</p> <p>Firmware version FIR-v1650-B472161: Table entry "Access" at subindex 03 modified from "read / write" to "read only".</p> |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 03 _h |

| | |
|----------------|----------------------|
| Subindex | 01 _h |
| Name | Synchronization Type |
| Data type | UNSIGNED16 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 0000 _h |

| | |
|----------------|-----------------------|
| Subindex | 02 _h |
| Name | Cycle Time |
| Data type | UNSIGNED32 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

| | |
|----------------|-----------------------|
| Subindex | 03 _h |
| Name | Shift Time |
| Data type | UNSIGNED32 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

1C33h Input Sync Manager Synchronization

Function

Here all parameter for synchronization for input pdo mapping for EtherCAT are set (see **1C13_h**). These parameters will be set by EtherCAT Master.

Object description

| | |
|-------------|------------------------------------|
| Index | 1C33 _h |
| Object name | Input Sync Manager Synchronization |

| | |
|------------------|---|
| Object Code | RECORD |
| Data type | SYNCMGR_SYNCHRONIZATION |
| Savable | no |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1426 |
| Change history | <p>Firmware version FIR-v1626: Entry "Savable" modified from "No" to "yes, category: communication".</p> <p>Firmware version FIR-v1650-B472161: Entry "Savable" modified from "yes, category: communication" to "No".</p> <p>Firmware version FIR-v1650-B472161: Table entry "Access" at subindex 01 modified from "read / write" to "read only".</p> <p>Firmware version FIR-v1650-B472161: Table entry "Access" at subindex 02 modified from "read / write" to "read only".</p> <p>Firmware version FIR-v1650-B472161: Table entry "Access" at subindex 03 modified from "read / write" to "read only".</p> |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 03 _h |
| Subindex | 01 _h |
| Name | Synchronization Type |
| Data type | UNSIGNED16 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 0000 _h |
| Subindex | 02 _h |
| Name | Cycle Time |
| Data type | UNSIGNED32 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

| | |
|----------------|-----------------------|
| Subindex | 03 _h |
| Name | Shift Time |
| Data type | UNSIGNED32 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

Description

1F50h Program Data

Function

This object is used to program memory areas of the controller. Each entry stands for a certain memory area.

Object description

| | |
|------------------|-------------------|
| Index | 1F50 _h |
| Object name | Program Data |
| Object Code | ARRAY |
| Data type | DOMAIN |
| Savable | no |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1540 |
| Change history | |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 03 _h |

| | |
|-------------|----------------------------------|
| Subindex | 01 _h |
| Name | Program Data Bootloader/firmware |
| Data type | DOMAIN |
| Access | read / write |
| PDO mapping | no |

| | |
|----------------|------------------------|
| Allowed values | |
| Preset value | 0 |
| <hr/> | |
| Subindex | 02 _h |
| Name | Program Data NanoJ |
| Data type | DOMAIN |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 0 |
| <hr/> | |
| Subindex | 03 _h |
| Name | Program Data DataFlash |
| Data type | DOMAIN |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 0 |

1F51h Program Control

Function

This object is used to control the programming of memory areas of the controller. Each entry stands for a certain memory area.

Object description

| | |
|------------------|-------------------|
| Index | 1F51 _h |
| Object name | Program Control |
| Object Code | ARRAY |
| Data type | UNSIGNED8 |
| Savable | no |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1540 |
| Change history | |

Value description

| | |
|-----------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |

| | |
|----------------|-------------------------------------|
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 03 _h |
| <hr/> | |
| Subindex | 01 _h |
| Name | Program Control Bootloader/firmware |
| Data type | UNSIGNED8 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00 _h |
| <hr/> | |
| Subindex | 02 _h |
| Name | Program Control NanoJ |
| Data type | UNSIGNED8 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00 _h |
| <hr/> | |
| Subindex | 03 _h |
| Name | Program Control DataFlash |
| Data type | UNSIGNED8 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00 _h |

1F57h Program Status

Function

This object indicates the programming status during the programming of memory areas of the controller. Each entry stands for a certain memory area.

Object description

| | |
|-------------|-------------------|
| Index | 1F57 _h |
| Object name | Program Status |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | no |
| Access | read only |
| PDO mapping | no |

| | |
|------------------|-----------|
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1540 |
| Change history | |

Value description

| | |
|----------------|------------------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 03 _h |
| <hr/> | |
| Subindex | 01 _h |
| Name | Program Status Bootloader/firmware |
| Data type | UNSIGNED32 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 02 _h |
| Name | Program Status NanoJ |
| Data type | UNSIGNED32 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 03 _h |
| Name | Program Status DataFlash |
| Data type | UNSIGNED32 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

2030h Pole Pair Count

Function

Contains the number of pole pairs of the connected motor.

Object description

| | |
|------------------|---|
| Index | 2030 _h |
| Object name | Pole Pair Count |
| Object Code | VARIABLE |
| Data type | UNSIGNED32 |
| Savable | yes, category: tuning |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | <ul style="list-style-type: none"> • PD4-E591L42-E-65-1: 00000032_h • PD4-E601L42-E-65-1: 00000032_h • PD4-EB59CD-E-65-1: 00000003_h |
| Firmware version | FIR-v1426 |
| Change history | Firmware version FIR-v1540: "Savable" entry changed from "no" to "yes, category: tuning". |

2031h Maximum Current

Function

If **I²t monitoring** is not active, the rms current specified in the motor data sheet is entered here in mA. If **Closed Loop** mode is used or if **I²t monitoring** is activated, the maximum current value is specified here in mA.

Within the controller, the entered value is always interpreted as the root mean square.

Object description

| | |
|------------------|---|
| Index | 2031 _h |
| Object name | Maximum Current |
| Object Code | VARIABLE |
| Data type | UNSIGNED32 |
| Savable | yes, category: tuning |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | <ul style="list-style-type: none"> • PD4-E591L42-E-65-1: 00000834_h • PD4-E601L42-E-65-1: 00000834_h • PD4-EB59CD-E-65-1: 00001770_h |
| Firmware version | FIR-v1426 |
| Change history | <p>Firmware version FIR-v1614: "Savable" entry changed from "yes, category: application" to "yes, category: tuning".</p> <p>Firmware version FIR-v1614: "Object name" entry changed from "Peak Current" to "Max Current".</p> |

2034h Upper Voltage Warning Level

Function

This object contains the threshold value for the "overvoltage" error in millivolts.

Object description

| | |
|------------------|-----------------------------|
| Index | 2034 _h |
| Object name | Upper Voltage Warning Level |
| Object Code | VARIABLE |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 0000C92C _h |
| Firmware version | FIR-v1426 |
| Change history | |

Description

If the input voltage of the controller exceeds this threshold value, the motor is switched off and an error triggered. This error is reset automatically if the input voltage is less than (voltage of object 2034_h minus 2 volts).

2035h Lower Voltage Warning Level

Function

This object contains the threshold value for the "Undervoltage" error in millivolts.

Object description

| | |
|------------------|-----------------------------|
| Index | 2035 _h |
| Object name | Lower Voltage Warning Level |
| Object Code | VARIABLE |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00002710 _h |
| Firmware version | FIR-v1426 |
| Change history | |

Description

If the input voltage of the controller falls below this threshold value, the motor is switched off and an error triggered. The error is reset automatically if the input voltage exceeds the voltage of object 2035_h plus 2 volts.

2036h Open Loop Current Reduction Idle Time

Function

This object describes the time in milliseconds that the motor must be at a standstill before current reduction is activated.

Object description

| | |
|------------------|---------------------------------------|
| Index | 2036 _h |
| Object name | Open Loop Current Reduction Idle Time |
| Object Code | VARIABLE |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 000003E8 _h |
| Firmware version | FIR-v1426 |
| Change history | |

2037h Open Loop Current Reduction Value/factor

Function

This object describes the rms current to which the motor current is to be reduced if current reduction is activated in open loop (bit 3 in **3202_h** = "1") and the motor is at a standstill.

Object description

| | |
|------------------|--|
| Index | 2037 _h |
| Object name | Open Loop Current Reduction Value/factor |
| Object Code | VARIABLE |
| Data type | INTEGER32 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | FFFFFFCE _h |
| Firmware version | FIR-v1426 |
| Change history | |

Description

Value of 2037_h greater than or equal to 0 and less than value 2031_h

Current is reduced to the value entered here. The value is in mA and interpreted as root mean square.

Value of 2037_h in the range from -1 to -100

The entered value is interpreted as a percentage and determines the reduction of the rated current in 2037_h. The value in 2031_h is used for the calculation.

Example: Object 2031_h has the value 4200 mA. The value -60 in 2037_h reduces the current by 60% of 2031_h. The result is a current reduction to a root mean square of $2031_h * (2037_h + 100) / 100 = 1680$ mA.

The value -100 in 2037_h would, for example, mean that a current reduction is set to a root mean square of 0 mA.



Note

If the rated current is greater than 0 in 203B_h:01, the smaller of 2031_h and 203B_h:01 is used as the rated current for calculating the current reduction.

2038h Brake Controller Timing

Function

This object contains the times for the *brake control* in milliseconds as well as the PWM frequency and the duty cycle.

Object description

| | |
|------------------|----------------------------|
| Index | 2038 _h |
| Object name | Brake Controller Timing |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Firmware version | FIR-v1426 |
| Change history | |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 06 _h |

| | |
|----------|-----------------|
| Subindex | 01 _h |
|----------|-----------------|

| | |
|----------------|-----------------------|
| Name | Close Brake Idle Time |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 000003E8 _h |

| | |
|----------------|--------------------------|
| Subindex | 02 _h |
| Name | Shutdown Power Idle Time |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 000003E8 _h |

| | |
|----------------|-----------------------|
| Subindex | 03 _h |
| Name | Open Brake Delay Time |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 000003E8 _h |

| | |
|----------------|----------------------------|
| Subindex | 04 _h |
| Name | Start Operation Delay Time |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

| | |
|----------------|--|
| Subindex | 05 _h |
| Name | PWM Frequency |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | between 0 and 2000 (7D0 _h) |
| Preset value | 00000000 _h |

| | |
|-----------|-----------------|
| Subindex | 06 _h |
| Name | PWM Duty Cycle |
| Data type | UNSIGNED32 |
| Access | read / write |

| | |
|----------------|---|
| PDO mapping | no |
| Allowed values | 0, between 2 and 100 (64 _h) |
| Preset value | 00000000 _h |

Description

The subindices have the following functions:

- 01_h: Time between motor standstill and the closing of the brake.
- 02_h: Time between the closing of the brake and the switching off of the motor current.
- 03_h: Time between the switching on of the motor current and opening of the brake.
- 04_h: Time between the opening of the brake and when the *Operation enabled* state of the **CiA 402 Power State Machine** is reached.
- 05_h: Frequency of the brake PWM in hertz.
- 06_h: Duty cycle of the brake PWM in percent.

2039h Motor Currents

Function

This object contains the measured motor currents in mA.

Object description

| | |
|------------------|---|
| Index | 2039 _h |
| Object name | Motor Currents |
| Object Code | ARRAY |
| Data type | INTEGER32 |
| Savable | no |
| Firmware version | FIR-v1426 |
| Change history | <p>Firmware version FIR-v1504: "PDO mapping" table entry for subindex 01 changed from "no" to "TX-PDO".</p> <p>Firmware version FIR-v1504: "PDO mapping" table entry for subindex 02 changed from "no" to "TX-PDO".</p> <p>Firmware version FIR-v1504: "PDO mapping" table entry for subindex 03 changed from "no" to "TX-PDO".</p> <p>Firmware version FIR-v1504: "PDO mapping" table entry for subindex 04 changed from "no" to "TX-PDO".</p> |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 04 _h |

| | |
|----------------|-----------------------|
| Subindex | 01 _h |
| Name | I_d |
| Data type | INTEGER32 |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |

| | |
|----------------|-----------------------|
| Subindex | 02 _h |
| Name | I_q |
| Data type | INTEGER32 |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |

| | |
|----------------|-----------------------|
| Subindex | 03 _h |
| Name | I_a |
| Data type | INTEGER32 |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |

| | |
|----------------|-----------------------|
| Subindex | 04 _h |
| Name | I_b |
| Data type | INTEGER32 |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |

Description



Note

Motor currents I_d (subindex 01_h) and I_q (subindex 02_h) are only displayed if **closed loop** was activated; the value 0 is otherwise output.

203Ah Homing On Block Configuration

Function

This object contains the parameters for *Homing on Block* (see chapter **Homing**)

Object description

| | |
|------------------|---|
| Index | 203A _h |
| Object name | Homing On Block Configuration |
| Object Code | ARRAY |
| Data type | INTEGER32 |
| Savable | yes, category: application |
| Access | |
| PDO mapping | |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1426 |
| Change history | <p>Firmware version FIR-v1540: The number of entries was changed from 4 to 3.</p> <p>Firmware version FIR-v1540: "Name" entry changed from "Period Of Blocking" to "Block Detection time".</p> <p>Firmware version FIR-v1614: "Data Type" entry changed from "UNSIGNED32" to "INTEGER32".</p> <p>Firmware version FIR-v1614: "Savable" entry changed from "no" to "yes, category: application".</p> <p>Firmware version FIR-v1614: "Data Type" entry changed from "UNSIGNED32" to "INTEGER32".</p> <p>Firmware version FIR-v1614: "Data Type" entry changed from "UNSIGNED32" to "INTEGER32".</p> |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 02 _h |

| | |
|----------------|---|
| Subindex | 01 _h |
| Name | Minimum Current For Block Detection |
| Data type | INTEGER32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | <ul style="list-style-type: none"> • PD4-E591L42-E-65-1: 000004EC_h • PD4-E601L42-E-65-1: 000004EC_h • PD4-EB59CD-E-65-1: 000015E0_h |

| | |
|----------------|-----------------------|
| Subindex | 02 _h |
| Name | Block Detection Time |
| Data type | INTEGER32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 000000C8 _h |

Description

The subindices have the following function:

- 01_h: Specifies the current limit value above which blocking is to be detected. Positive numerical values specify the current limit in mA, negative numbers specify a percentage of object **2031_h:01_h**. Example: The value "1000" corresponds to 1000 mA (= 1 A); the value "-70" corresponds to 70% of **2031_h**.
- 02_h: Specifies the time in ms that the motor is to continue to travel against the block after block detection.

203Bh I²t Parameters

Function

This object contains the parameters for I²t monitoring.

I²t monitoring is activated by entering a value greater than 0 in **203B_h:01** and **203B_h:02** (see **I²t Motor overload protection**).

With one exception, I²t monitoring can only be used for *closed loop* mode: If I²t is activated in *open loop* mode, the current is reduced to the smaller of **203B_h** and **2031_h**.

Object description

| | |
|------------------|---|
| Index | 203B _h |
| Object name | I ² t Parameters |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | yes, category: tuning |
| Firmware version | FIR-v1426 |
| Change history | Firmware version FIR-v1512: "Savable" entry changed from "no" to "yes, category: application". Firmware version FIR-v1512: The number of entries was changed from 7 to 8. Firmware version FIR-v1614: "Savable" entry changed from "yes, category: application" to "yes, category: tuning". |

Value description

| | |
|-----------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |

| | |
|----------------|---|
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 07 _h |
| <hr/> | |
| Subindex | 01 _h |
| Name | Nominal Current |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | <ul style="list-style-type: none"> • PD4-E591L42-E-65-1: 00000000_h • PD4-E601L42-E-65-1: 00000000_h • PD4-EB59CD-E-65-1: 00000BB8_h |
| <hr/> | |
| Subindex | 02 _h |
| Name | Maximum Duration Of Peak Current |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | <ul style="list-style-type: none"> • PD4-E591L42-E-65-1: 00000000_h • PD4-E601L42-E-65-1: 00000000_h • PD4-EB59CD-E-65-1: 000003E8_h |
| <hr/> | |
| Subindex | 03 _h |
| Name | Threshold |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 04 _h |
| Name | CalcValue |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 05 _h |
| Name | LimitedCurrent |

| | |
|----------------|-----------------------|
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 06 _h |
| Name | Status |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 07 _h |
| Name | ActualResistance |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

Description

The subindices are divided into two groups: subindex 01_h and 02_h contain parameters for the control, subindices 03_h to 06_h are status values. The functions are as follows:

- 01_h: The rated current specified in the motor data sheet is entered here in mA. This must be smaller than the current entered in object **2031_h**, otherwise monitoring is not activated. The specified value is interpreted as root mean square.
- 02_h: Specifies the maximum duration of the peak current in ms.
- 03_h: Threshold, specifies the limit in mA that determines whether the maximum current or rated current is switched to.
- 04_h: CalcValue, specifies the calculated value that is compared with the threshold for setting the current.
- 05_h: LimitedCurrent, contains the momentary current as root mean square set by I^2t .
- 06_h: Current status. If the sub-entry value is "0", I^2t is deactivated; if the value is "1", I^2t is activated.

203Dh Torque Window

Function

Specifies a symmetrical range relative to the target torque within which the target is considered having been met.

If the value is set to "FFFFFFFF"_h, monitoring is switched off, the "Target reached" bit in object **6041_h** (statusword) is never set.

Object description

| | |
|------------------|--|
| Index | 203D _h |
| Object name | Torque Window |
| Object Code | VARIABLE |
| Data type | UNSIGNED16 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 0000 _h |
| Firmware version | FIR-v1540 |
| Change history | Firmware version FIR-v1614: "Savable" entry changed from "no" to "yes, category: application". |

203Eh Torque Window Time Out

Function

The current torque must be within the "Torque Window" (**203D_h**) for this time (in milliseconds) for the target torque to be considered having been met.

Object description

| | |
|------------------|--|
| Index | 203E _h |
| Object name | Torque Window Time Out |
| Object Code | VARIABLE |
| Data type | UNSIGNED16 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 0000 _h |
| Firmware version | FIR-v1540 |
| Change history | Firmware version FIR-v1614: "Savable" entry changed from "no" to "yes, category: application". Firmware version FIR-v1738-B501312: "Object name" entry changed from "Torque Window Time" to "Torque Window Time Out". |

203Fh Max Slippage Time Out

Function

Time in milliseconds until an excessively large slippage error in **Profile Velocity** mode results in an error message.

Object description

| | |
|------------------|----------------------------|
| Index | 203F _h |
| Object name | Max Slippage Time Out |
| Object Code | VARIABLE |
| Data type | UNSIGNED16 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 0064 _h |
| Firmware version | FIR-v1738-B501312 |
| Change history | |

Description

If the actual speed deviates so much from the set speed that the value (absolute value) of this object is exceeded, bit 13 in object **6041**_h is set. The deviation must last longer than the time in object **203F**_h.

A reaction to the slippage error can be set in object **3700**_h. If a reaction is defined, an error is also entered in object **1003**_h.

2056h Limit Switch Tolerance Band

Function

Specifies how far a limit switch may be passed over in the positive or negative direction before the controller triggers an error.

This tolerance band is necessary, for example, to complete homing operations – in which limit switches can be actuated – error free.

Object description

| | |
|------------------|-----------------------------|
| Index | 2056 _h |
| Object name | Limit Switch Tolerance Band |
| Object Code | VARIABLE |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 000001F4 _h |
| Firmware version | FIR-v1426 |
| Change history | |

2057h Clock Direction Multiplier

Function

The clock count value in **Clock-direction mode** is multiplied by this value before it is processed further.

Object description

| | |
|------------------|----------------------------|
| Index | 2057 _h |
| Object name | Clock Direction Multiplier |
| Object Code | VARIABLE |
| Data type | INTEGER32 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000080 _h |
| Firmware version | FIR-v1426 |
| Change history | |

2058h Clock Direction Divider

Function

The clock count value in **Clock-direction mode** is divided by this value before it is processed further.

Object description

| | |
|------------------|----------------------------|
| Index | 2058 _h |
| Object name | Clock Direction Divider |
| Object Code | VARIABLE |
| Data type | INTEGER32 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000001 _h |
| Firmware version | FIR-v1426 |
| Change history | |

2059h Encoder Configuration

Function

This object can be used to switch the supply voltage and the type of encoder.

Object description

| | |
|------------------|---|
| Index | 2059 _h |
| Object name | Encoder Configuration |
| Object Code | VARIABLE |
| Data type | UNSIGNED32 |
| Savable | yes, category: tuning |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| Firmware version | FIR-v1426 |
| Change history | Firmware version FIR-v1614: "Savable" entry changed from "yes, category: application" to "yes, category: tuning". |

Description

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | | | | | | | | | | | | TYPE | |

TYPE

Defines the type of encoder. For a differential encoder, the bit must have the value "0". For a single-ended encoder, the bit must be set to "1".

205Ah Absolute Sensor Boot Value (in User Units)

Function

The initial encoder position when switching on the controller (in **user-defined units**) can be read from this object.

Object description

| | |
|------------------|--|
| Index | 205A _h |
| Object name | Absolute Sensor Boot Value (in User Units) |
| Object Code | VARIABLE |
| Data type | INTEGER32 |
| Savable | no |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| Firmware version | FIR-v1446 |
| Change history | Firmware version FIR-v1512: "Access" table entry for subindex 00 changed from "read/write" to "read only". |

Firmware version FIR-v1738-B501312: "Object name" entry changed from "Encoder Boot Value" to "Absolute Sensor Boot Value (in User Units)".

Firmware version FIR-v1738-B501312: "Data type" entry changed from "UNSIGNED32" to "INTEGER32".

205Bh Clock Direction Or Clockwise/Counter Clockwise Mode

Function

This object can be used to switch the clock-direction mode (value = "0") to the **right/left rotation mode** (value = "1").

Object description

| | |
|------------------|---|
| Index | 205B _h |
| Object name | Clock Direction Or Clockwise/Counter Clockwise Mode |
| Object Code | VARIABLE |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| Firmware version | FIR-v1504 |
| Change history | |

2084h Bootup Delay

Function

Defines the period between the time that supply voltage is applied to the controller and the functional readiness of the controller in milliseconds.

Object description

| | |
|------------------|----------------------------|
| Index | 2084 _h |
| Object name | Bootup Delay |
| Object Code | VARIABLE |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| Firmware version | FIR-v1426 |
| Change history | |

2101h Fieldbus Module Availability

Function

Shows the available fieldbuses.

Object description

| | |
|------------------|---|
| Index | 2101 _h |
| Object name | Fieldbus Module Availability |
| Object Code | VARIABLE |
| Data type | UNSIGNED32 |
| Savable | no |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000020 _h |
| Firmware version | FIR-v1426 |
| Change history | Firmware version FIR-v1626: "Object name" entry changed from "Fieldbus Module" to "Fieldbus Module Availability". |

Description

Bits 0 to 15 represent the physical interface, bits 16 to 31 the used protocol (if necessary).

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|-----|-------|-------|-----|-------|-------|------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | | | | | | | | | | | | | E-IP | MTCP | MRTU |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | | | | | | | SPI | E-CAT | E-NET | CAN | RS232 | RS485 | USB |

USB

Value = "1": The USB fieldbus is available.

RS-485

Value = "1": An RS-485 interface is available.

RS-232

Value = "1": An RS-232 interface is available.

CAN

Value = "1": The CANopen fieldbus is available.

E-NET

Value = "1": An Ethernet interface is available.

E-CAT

Value = "1": An EtherCAT interface is available.

SPI

Value = "1": An SPI interface is available.

MRTU

Value = "1": The used protocol is Modbus RTU.

MTCP

Value = "1": The used protocol is Modbus TCP.

E-IP

Value = "1": The used protocol is EtherNet/IP™.

2102h Fieldbus Module Control

Function

This object can be used to activate/deactivate certain fieldbuses (physical interfaces and protocols).

Object description

| | |
|------------------|--|
| Index | 2102 _h |
| Object name | Fieldbus Module Control |
| Object Code | VARIABLE |
| Data type | UNSIGNED32 |
| Savable | yes, category: communication |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000020 _h |
| Firmware version | FIR-v1540 |
| Change history | Firmware version FIR-v1626: "Savable" entry changed from "yes, category: application" to "yes, category: communication". |

Description

Object **2103_h:1_h** contains all physical interfaces/protocols that can be activated/deactivated. These can be switched in this object (2102_h). The current status of the activated fieldbuses is in object **2103_h:2_h**.

The following distribution of the bits applies here:

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|-----|-------|-------|-----|-------|-------|------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | | | | | | | | | | | | | E-IP | MTCP | MRTU |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | | | | | | | SPI | E-CAT | E-NET | CAN | RS232 | RS485 | USB |

USB

USB interface

RS-485

RS-485 interface

RS-232

RS-232 interface

CAN

CANopen interface

E-NET

EtherNet interface

E-CAT

EtherCAT interface

SPI

SPI interface

MRTU

Modbus RTU protocol

MTCP

Modbus TCP protocol

E-IP

EtherNet/IP™ protocol

2103h Fieldbus Module Status

Function

Shows the active fieldbuses.

Object description

| | |
|------------------|------------------------|
| Index | 2103 _h |
| Object name | Fieldbus Module Status |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | no |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1540 |
| Change history | |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 02 _h |

| | |
|----------------|------------------------------|
| Subindex | 01 _h |
| Name | Fieldbus Module Disable Mask |
| Data type | UNSIGNED32 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

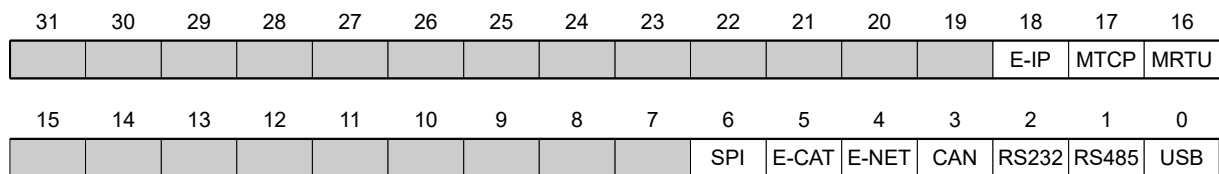
| | |
|----------------|-------------------------|
| Subindex | 02 _h |
| Name | Fieldbus Module Enabled |
| Data type | UNSIGNED32 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000020 _h |

Description

Subindex 1 (Fieldbus Module Disable Mask): This subindex contains all physical interfaces and protocols that can be activated or deactivated. A value "1" means that this fieldbus can be deactivated.

Subindex 2 (Fieldbus Module Enabled): This subindex contains all currently activated physical interfaces and protocols. The value "1" means that that the fieldbus is active.

The following distribution of the bits applies for subindices 1 and 2:



USB

USB interface

RS-485

RS-485 interface

RS-232

RS-232 interface

CAN

CANopen interface

E-NET

EtherNet interface

E-CAT

EtherCAT interface

SPI

SPI interface

MRTU

Modbus RTU protocol

MTCP

Modbus TCP protocol

E-IP

EtherNet/IP™ protocol

2110h EtherCAT Slave Status

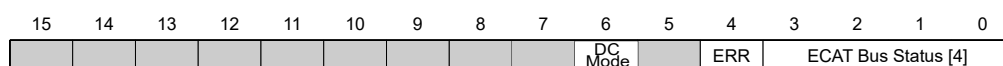
Function

Shows the operating state of the EtherCAT Slave Module.

Object description

| | |
|------------------|-----------------------|
| Index | 2110 _h |
| Object name | EtherCAT Slave Status |
| Object Code | VARIABLE |
| Data type | UNSIGNED16 |
| Savable | no |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 0000 _h |
| Firmware version | FIR-v1426 |
| Change history | |

Description



ECAT Bus Status [4]

The actual EtherCAT bus status is entered here

- Value = 01_h: Bus status INIT
- Value = 02_h: Bus status PREOPERATIONAL
- Value = 03_h: Bus status BOOT
- Value = 04_h: Bus status SAFEOPERATIONAL
- Value = 08_h: Bus status OPERATIONAL

ERR

Value = "1": An error is active

DC Mode

Value = "11_b" (3_d): EtherCAT synchronisation active (DC Mode), value "0" means "no synchronisation"

2300h NanoJ Control

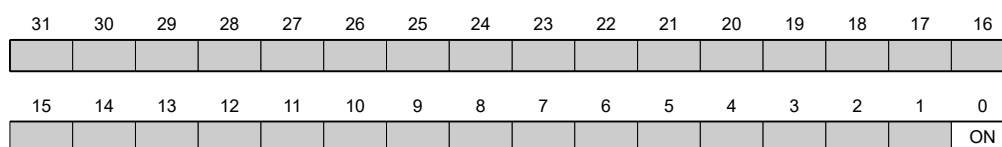
Function

Controls the execution of a NanoJ program.

Object description

| | |
|------------------|--|
| Index | 2300 _h |
| Object name | NanoJ Control |
| Object Code | VARIABLE |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |
| Firmware version | FIR-v1426 |
| Change history | Firmware version FIR-v1436: "Object name" entry changed from "VMM Control" to "NanoJ Control". |

Description



ON

Switches the NanoJ program on (value = "1") or off (value = "0").

With a rising edge in bit 0, the program is first reloaded and the variable range reset.



Note

Startup of the NanoJ program can take up to 200 ms.

2301h NanoJ Status

Function

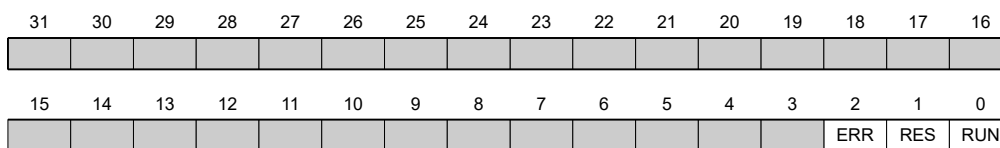
Indicates the operating state of the user program.

Object description

| | |
|-------------|-------------------|
| Index | 2301 _h |
| Object name | NanoJ Status |
| Object Code | VARIABLE |

| | |
|------------------|--|
| Data type | UNSIGNED32 |
| Savable | no |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |
| Firmware version | FIR-v1426 |
| Change history | Firmware version FIR-v1436: "Object name" entry changed from "VMM Status" to "NanoJ Status". |

Description



RUN

Value = "0": Program is stopped, value = "1": NanoJ program is running.

RES

Reserved.

ERR

Program was ended with an error. Cause of the error can be read from object **2302_h**.

2302h NanoJ Error Code

Function

Indicates which error occurred during the execution of the user program.

Object description

| | |
|------------------|--|
| Index | 2302 _h |
| Object name | NanoJ Error Code |
| Object Code | VARIABLE |
| Data type | UNSIGNED32 |
| Savable | no |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |
| Firmware version | FIR-v1426 |
| Change history | Firmware version FIR-v1436: "Object name" entry changed from "VMM Error Code" to "NanoJ Error Code". |

Description

Error codes during program execution:

| Number | Description |
|-------------------|---|
| 0000 _h | Not an error |
| 0001 _h | Firmware does not (yet) support the used function |
| 0002 _h | Not or incorrectly initialized pointer |
| 0003 _h | Impermissible access to system resource |
| 0004 _h | Hard fault (internal error) |
| 0005 _h | Code executed too long without yield() or sleep() |
| 0006 _h | Impermissible access to system resource |
| 0007 _h | Too many variables on the stack |
| 0100 _h | Invalid NanoJ program file |

Error when accessing an object:

| Number | Description |
|-----------------------|--|
| 10xxxxyy _h | Invalid mapping in the NanoJ program file: The value in "xxxx" specifies the index, the value in "yy" specifies the subindex of the object that should – but cannot – be mapped. |
| 1000 _h | Access of a nonexistent object in the object dictionary |
| 1001 _h | Write access of a write-protected entry in the OD |
| 1002 _h | Internal file system error |

File system error codes when loading the user program:

| Number | Description |
|--------------------|--|
| 10002 _h | Internal file system error |
| 10003 _h | Storage medium not ready |
| 10004 _h | File not found |
| 10005 _h | Folder not found |
| 10006 _h | Invalid file name/folder name |
| 10008 _h | Access of file not possible |
| 10009 _h | File/directory object is invalid |
| 1000A _h | Storage medium is read-only |
| 1000B _h | Drive number is invalid |
| 1000C _h | Working range of the drive is invalid |
| 1000D _h | No valid file system on the drive |
| 1000E _h | Creation of the file system failed |
| 1000F _h | Access not possible within the required time |
| 10010 _h | Access was rejected |

230Fh Uptime Seconds

Function

This object contains the operating time in seconds since the last time the controller was started.



Note

This object is not stored; counting begins with "0" again after switching on.

Object description

| | |
|------------------|-----------------------|
| Index | 230F _h |
| Object name | Uptime Seconds |
| Object Code | VARIABLE |
| Data type | UNSIGNED32 |
| Savable | no |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |
| Firmware version | FIR-v1436 |
| Change history | |

2310h NanoJ Input Data Selection

Function

Describes the object dictionary entries that are copied to the PDO mapping input of the NanoJ program.

Object description

| | |
|------------------|--|
| Index | 2310 _h |
| Object name | NanoJ Input Data Selection |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | no |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1650-B472161 |
| Change history | Firmware version FIR-v1436: "Object name" entry changed from "VMM Input Data Selection" to "NanoJ Input Data Selection". Firmware version FIR-v1650-B472161: "Savable" entry changed from "yes, category: application" to "no". |

Firmware version FIR-v1650-B472161: "Access" table entry for subindex 00 changed from "read/write" to "read only".

Firmware version FIR-v1650-B472161: "Access" table entry for subindex 01 changed from "read/write" to "read only".

Value description

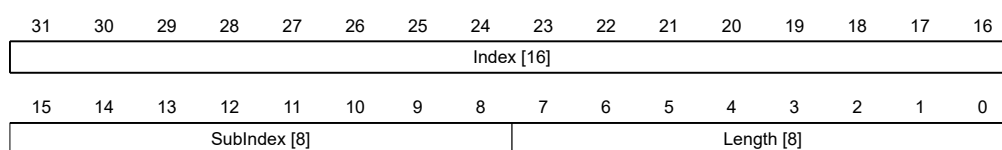
| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 10 _h |

| | |
|----------------|-----------------------------------|
| Subindex | 01 _h - 10 _h |
| Name | Mapping #1 - #16 |
| Data type | UNSIGNED32 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

Description

Each subindex (1–16) describes a different mapped object.

A mapping entry consists of four bytes, which are structured according to the following graphic.



Index [16]

This contains the index of the object to be mapped.

Subindex [8]

This contains the subindex of the object to be mapped.

Length [8]

This contains the length of the object to be mapped in units of bits.

2320h NanoJ Output Data Selection

Function

Describes the object dictionary entries that are copied into the output PDO mapping of the *NanoJ program* after it is executed.

Object description

| | |
|------------------|---|
| Index | 2320 _h |
| Object name | NanoJ Output Data Selection |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | no |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1650-B472161 |
| Change history | <p>Firmware version FIR-v1436: "Object name" entry changed from "VMM Output Data Selection" to "NanoJ Output Data Selection".</p> <p>Firmware version FIR-v1650-B472161: "Savable" entry changed from "yes, category: application" to "no".</p> <p>Firmware version FIR-v1650-B472161: "Access" table entry for subindex 00 changed from "read/write" to "read only".</p> <p>Firmware version FIR-v1650-B472161: "Access" table entry for subindex 01 changed from "read/write" to "read only".</p> |

Value description

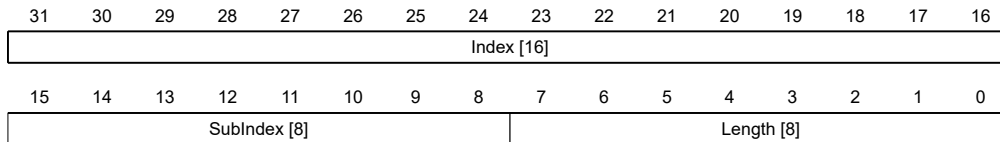
| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 10 _h |

| | |
|----------------|-----------------------------------|
| Subindex | 01 _h - 10 _h |
| Name | Mapping #1 - #16 |
| Data type | UNSIGNED32 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

Description

Each subindex (1–16) describes a different mapped object.

A mapping entry consists of four bytes, which are structured according to the following graphic.



Index [16]

This contains the index of the object to be mapped.

Subindex [8]

This contains the subindex of the object to be mapped.

Length [8]

This contains the length of the object to be mapped in units of bits.

2330h NanoJ In/output Data Selection

Function

Describes the object dictionary entries that are first copied to the input PDO mapping of the NanoJ program and, after it is executed, are copied back to the output PDO mapping.

Object description

| | |
|------------------|---|
| Index | 2330 _h |
| Object name | NanoJ In/output Data Selection |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | no |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1650-B472161 |
| Change history | <p>Firmware version FIR-v1436: "Object name" entry changed from "VMM In/output Data Selection" to "NanoJ In/output Data Selection".</p> <p>Firmware version FIR-v1650-B472161: "Savable" entry changed from "yes, category: application" to "no".</p> <p>Firmware version FIR-v1650-B472161: "Access" table entry for subindex 00 changed from "read/write" to "read only".</p> <p>Firmware version FIR-v1650-B472161: "Access" table entry for subindex 01 changed from "read/write" to "read only".</p> |

Value description

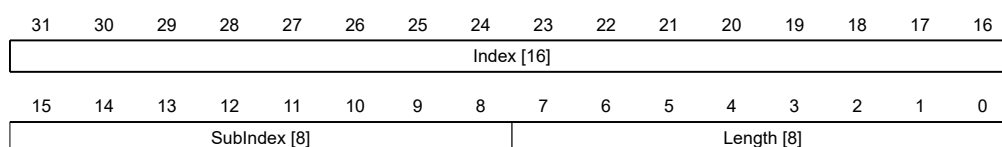
| | |
|-----------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |

| | |
|----------------|-----------------------------------|
| PDO mapping | no |
| Allowed values | |
| Preset value | 10 _h |
| <hr/> | |
| Subindex | 01 _h - 10 _h |
| Name | Mapping #1 - #16 |
| Data type | UNSIGNED32 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

Description

Each subindex (1–16) describes a different mapped object.

A mapping entry consists of four bytes, which are structured according to the following graphic.



Index [16]

This contains the index of the object to be mapped.

Subindex [8]

This contains the subindex of the object to be mapped.

Length [8]

This contains the length of the object to be mapped in units of bits.

2400h NanoJ Inputs

Function

Located here is an array with 32, 32-bit integer values that is not used within the firmware and serves only for communicating with the user program via the fieldbus.

Object description

| | |
|------------------|---|
| Index | 2400 _h |
| Object name | NanoJ Inputs |
| Object Code | ARRAY |
| Data type | INTEGER32 |
| Savable | no |
| Firmware version | FIR-v1426 |
| Change history | The number of entries was changed from 2 to 33. Firmware version FIR-v1436: "Object name" entry changed from "VMM Inputs" to "NanoJ Inputs". |

Firmware version FIR-v1436: "Name" entry changed from "VMM Input N#" to "NanoJ Input N#".

Value description

| | |
|----------------|-----------------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 20 _h |
| Subindex | 01 _h - 20 _h |
| Name | NanoJ Input #1 - #32 |
| Data type | INTEGER32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |

Description

Here, it is possible to pass, e.g., preset values, to the *NanoJ program*.

2410h NanoJ Init Parameters

Function

This object functions identically to object **2400_h** with the difference that this object can be stored.

Object description

| | |
|------------------|--|
| Index | 2410 _h |
| Object name | NanoJ Init Parameters |
| Object Code | ARRAY |
| Data type | INTEGER32 |
| Savable | yes, category: application |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1450 |
| Change history | Firmware version FIR-v1450: "Data Type" entry changed from "INTEGER32" to "UNSIGNED8". |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 20 _h |

| | |
|----------------|-----------------------------------|
| Subindex | 01 _h - 20 _h |
| Name | NanoJ Init Parameter #1 - #32 |
| Data type | INTEGER32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |

2500h NanoJ Outputs

Function

Located here is an array with 32, 32-bit integer values that is not used within the firmware and serves only for communicating with the user program via the fieldbus.

Object description

| | |
|------------------|---|
| Index | 2500 _h |
| Object name | NanoJ Outputs |
| Object Code | ARRAY |
| Data type | INTEGER32 |
| Savable | no |
| Firmware version | FIR-v1426 |
| Change history | Firmware version FIR-v1436: "Object name" entry changed from "VMM Outputs" to "NanoJ Outputs". Firmware version FIR-v1436: "Name" entry changed from "VMM Output N#" to "NanoJ Output N#". |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |

| | |
|----------------|-----------------------------------|
| Preset value | 20 _h |
| Subindex | 01 _h - 20 _h |
| Name | NanoJ Output #1 - #32 |
| Data type | INTEGER32 |
| Access | read / write |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |

Description

Here, the *NanoJ program* can store results which can then be read out via the fieldbus.

2600h NanoJ Debug Output

Function

This object contains debug output of a user program.

Object description

| | |
|------------------|--|
| Index | 2600 _h |
| Object name | NanoJ Debug Output |
| Object Code | ARRAY |
| Data type | UNSIGNED8 |
| Savable | no |
| Firmware version | FIR-v1426 |
| Change history | Firmware version FIR-v1436: "Object name" entry changed from "VMM Debug Output" to "NanoJ Debug Output". |

Value description

| | |
|----------------|-----------------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00 _h |
| Subindex | 01 _h - 40 _h |
| Name | Value #1 - #64 |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |

| | |
|----------------|-----------------|
| Allowed values | |
| Preset value | 00 _h |

Description

Here, the NanoJ program stores the debug output that was called up with the `VmmDebugOutputString()`, `VmmDebugOutputInt()` and similar functions.

2701h Customer Storage Area

Function

Data can be deposited and stored in this object.

Object description

| | |
|------------------|---|
| Index | 2701 _h |
| Object name | Customer Storage Area |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | yes, category: customer |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1540 |
| Change history | Firmware version FIR-v1540: "Data Type" entry changed from "UNSIGNED32" to "UNSIGNED8". |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | FE _h |

| | |
|----------------|-----------------------------------|
| Subindex | 01 _h - FE _h |
| Name | Storage #1 - #254 |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

2800h Bootloader And Reboot Settings

Function

With this object, a reboot of the firmware can be triggered and the short circuiting of the motor windings in boot loader mode switched off and on.

Object description

| | |
|------------------|--------------------------------|
| Index | 2800 _h |
| Object name | Bootloader And Reboot Settings |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1540 |
| Change history | |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 03 _h |

| | |
|----------------|-----------------------|
| Subindex | 01 _h |
| Name | Reboot Command |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

| | |
|----------------|-------------------------|
| Subindex | 02 _h |
| Name | Reboot Delay Time In Ms |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

| | |
|----------------|-----------------------|
| Subindex | 03 _h |
| Name | Bootloader HW Config |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

Description

The subindices have the following function:

- 01_h: If the value "746F6F62_h" is entered here, the firmware is rebooted.
- 02_h: Time in milliseconds: delays the reboot of the firmware by the respective time.
- 03_h: Bit 0 can be used to switch short circuiting of the motor windings in boot loader mode off and on:
 - Bit 0 = 1: Short circuiting of the motor windings in boot loader mode is switched off.
 - Bit 0 = 0: Short circuiting of the motor windings in boot loader mode is switched on.

3202h Motor Drive Submode Select

Function

Controls the controller mode, such as the changeover between *closed loop* / *open loop* and whether Velocity Mode is simulated via the S-controller or functions with a real V-controller in *closed loop*.

Object description

| | |
|------------------|--|
| Index | 3202 _h |
| Object name | Motor Drive Submode Select |
| Object Code | VARIABLE |
| Data type | UNSIGNED32 |
| Savable | yes, category: drive |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | <ul style="list-style-type: none"> • PD4-E591L42-E-65-1: 00000000_h • PD4-E601L42-E-65-1: 00000000_h • PD4-EB59CD-E-65-1: 00000040_h |
| Firmware version | FIR-v1426 |
| Change history | <p>Firmware version FIR-v1540: "Savable" entry changed from "yes category: application" to "yes, category: travel".</p> <p>Firmware version FIR-v1540: "Savable" entry changed from "yes category: travel" to "yes, category: movement".</p> |

Description

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|------|--------|----|--------|-------|-----|-------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | | | | | | | BLDC | Torque | | CurRed | Brake | VoS | CL/OL |

CL/OL

Changeover between *open loop* and *closed loop*

- Value = "0": *open loop*
- Value = "1": *closed loop*

VoS

Value = "1": Simulate V-controller with an S-ramp: simulate the speed modes through continuous position changes

Brake

Value = "1": Switch on automatic brake control

CurRed (Current Reduction)

Value = "1": Current reduction activated in *open loop*

Torque

only active in operating modes **Profile Torque** and **Cyclic Synchronous Torque**

Value = "1": M-controller is active, otherwise a V-controller is superimposed: no V-controller is used in the torque modes for speed limiting, thus object **6080_h** is ignored; **3210_h:3** and **3210_h:4** have no effect on the control.

BLDC

Value = "1": Motor type "BLDC" (brushless DC motor)

3203h Feedback Selection

Function

In this object, the sources of the presets are defined for the commutation and the speed and position control.

Object description

| | |
|------------------|----------------------|
| Index | 3203 _h |
| Object name | Feedback Selection |
| Object Code | ARRAY |
| Data type | UNSIGNED8 |
| Savable | yes, category: drive |
| Access | read only |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1748-B531667 |

| | |
|----------------|---|
| Change history | Firmware version FIR-v1748-B538662: entry "Savable" changed from "yes, category: movement" to "yes, category: drive". Firmware version FIR-v1748-B538662: The number of entries was changed from 4 to 3. |
|----------------|---|

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 02 _h |

| | |
|----------------|------------------------|
| Subindex | 01 _h |
| Name | 1st Feedback Interface |
| Data type | UNSIGNED8 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00 _h |

| | |
|----------------|------------------------|
| Subindex | 02 _h |
| Name | 2nd Feedback Interface |
| Data type | UNSIGNED8 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00 _h |

Description

The subindices have the following function:

- 00_h: Value="1" to "n", where "n" is the number of existing feedbacks.
- n_h:
Subindex n contains a bit mask for the respective feedback n. The bits have the following meaning here:
 - Bit 0: If the bit is set to "1", feedback n is used for position control.
 - Bit 1: If the bit is set to "1", feedback n is used for speed control.
 - Bit 2: If the bit is set to "1", feedback n is used for the commutation in **Closed Loop**.

Subindex 01_h always corresponds to the first (and always existing) *sensorless* feedback.

Which feedback the controller takes into account for the individual controllers (commutation, speed, position) is implicitly specified by the order of the feedbacks.

The search always begins with feedback 2 and continues in ascending order until all existing product-specific feedbacks have been queried. If a feedback is found whose parameterization is set, it is assigned to the corresponding controller and the search ended.



Note

If bit 0 in 3202_h is set to 0, *closed loop* is deactivated; bit 2 (commutation) then has no meaning. Bit 1 for the speed and bit 0 for the position in the respective subindices are still used for the display of the actual position and speed values.

3204h Feedback Mapping

Function

This object contains information on the existing feedbacks.

Object description

| | |
|------------------|--|
| Index | 3204 _h |
| Object name | Feedback Mapping |
| Object Code | ARRAY |
| Data type | UNSIGNED16 |
| Savable | no |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1748-B531667 |
| Change history | Firmware version FIR-v1748-B538662: The number of entries was changed from 4 to 3. |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 02 _h |

| | |
|----------------|---------------------------------|
| Subindex | 01 _h |
| Name | Index Of 1st Feedback Interface |
| Data type | UNSIGNED16 |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |

| | |
|----------------|---------------------------------|
| Preset value | 3380 _h |
| Subindex | 02 _h |
| Name | Index Of 2nd Feedback Interface |
| Data type | UNSIGNED16 |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 33A0 _h |

Description

The subindices have the following function:

- 00_h: Value="1" to "n", where "n" is the number of existing feedbacks.
- n_h:
Subindex n refers to the index of the respective object for the configuration of the corresponding feedback.
Subindex 01_h always corresponds to the first (and always existing) *sensorless* feedback.

3210h Motor Drive Parameter Set

Function

Contains the P and I components of the current, speed and position controllers for *open loop* (only current controller activated) and *closed loop*.

Object description

| | |
|------------------|--|
| Index | 3210 _h |
| Object name | Motor Drive Parameter Set |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Access | read only |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1426 |
| Change history | <p>Firmware version FIR-v1626: "Name" entry changed from "S_P" to "Position Loop, Proportional Gain (closed loop)".</p> <p>Firmware version FIR-v1626: "Name" entry changed from "S_I" to "Position Loop, Integral Gain (closed loop)".</p> <p>Firmware version FIR-v1626: "Name" entry changed from "V_P" to "Velocity Loop, Proportional Gain (closed loop)".</p> <p>Firmware version FIR-v1626: "Name" entry changed from "V_I" to "Velocity Loop, Integral Gain (closed loop)".</p> <p>Firmware version FIR-v1626: "Name" entry changed from "Id_P" to "Flux Current Loop, Proportional Gain (closed loop)".</p> |

Firmware version FIR-v1626: "Name" entry changed from "Id_I" to "Flux Current Loop, Integral Gain (closed loop)".

Firmware version FIR-v1626: "Name" entry changed from "Iq_P" to "Torque Current Loop, Proportional Gain (closed loop)".

Firmware version FIR-v1626: "Name" entry changed from "Iq_I" to "Torque Current Loop, Integral Gain (closed loop)".

Firmware version FIR-v1626: "Name" entry changed from "I_P" to "Torque Current Loop, Proportional Gain (dspDrive – Stepper Motor, open loop)".

Firmware version FIR-v1626: "Name" entry changed from "I_I" to "Torque Current Loop, Integral Gain (dspDrive – Stepper Motor, open loop)".

Firmware version FIR-v1650-B472161: "Name" entry changed from "Torque Current Loop, Proportional Gain (dspDrive – Stepper Motor, open loop)" to "Torque Current Loop, Proportional Gain (open loop)".

Firmware version FIR-v1650-B472161: "Name" entry changed from "Torque Current Loop, Integral Gain (dspDrive – Stepper Motor, open loop)" to "Torque Current Loop, Integral Gain (open loop)".

Firmware version FIR-v1650-B472161: "Data type" entry changed from "INTEGER32" to "UNSIGNED32".

Firmware version FIR-v1650-B472161: "Data type" entry changed from "INTEGER32" to "UNSIGNED32".

Firmware version FIR-v1738-B501312: The number of entries was changed from 11 to 13.

Firmware version FIR-v1738-B501312: "PDO mapping" table entry for subindex 00 to 0A changed from "no" to "RX-PDO".

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 0C _h |

| | |
|----------------|---|
| Subindex | 01 _h |
| Name | Position Loop, Proportional Gain (closed Loop) |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | <ul style="list-style-type: none"> • PD4-E591L42-E-65-1: 00002710_h • PD4-E601L42-E-65-1: 00000800_h • PD4-EB59CD-E-65-1: 00007530_h |

| | |
|----------------|---|
| Subindex | 02 _h |
| Name | Position Loop, Integral Gain (closed Loop) |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 03 _h |
| Name | Velocity Loop, Proportional Gain (closed Loop) |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | <ul style="list-style-type: none"> • PD4-E591L42-E-65-1: 00004E20_h • PD4-E601L42-E-65-1: 00001B58_h • PD4-EB59CD-E-65-1: 0000EA60_h |
| <hr/> | |
| Subindex | 04 _h |
| Name | Velocity Loop, Integral Gain (closed Loop) |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | <ul style="list-style-type: none"> • PD4-E591L42-E-65-1: 00000064_h • PD4-E601L42-E-65-1: 00000004_h • PD4-EB59CD-E-65-1: 000001F4_h |
| <hr/> | |
| Subindex | 05 _h |
| Name | Flux Current Loop, Proportional Gain (closed Loop) |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | <ul style="list-style-type: none"> • PD4-E591L42-E-65-1: 0007A120_h • PD4-E601L42-E-65-1: 000668A0_h • PD4-EB59CD-E-65-1: 000061A8_h |
| <hr/> | |
| Subindex | 06 _h |
| Name | Flux Current Loop, Integral Gain (closed Loop) |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | |

Allowed values

| | |
|--------------|---|
| Preset value | <ul style="list-style-type: none"> • PD4-E591L42-E-65-1: 00001388_h • PD4-E601L42-E-65-1: 00002EE0_h • PD4-EB59CD-E-65-1: 00000BB8_h |
|--------------|---|

| | |
|----------------|---|
| Subindex | 07 _h |
| Name | Torque Current Loop, Proportional Gain (closed Loop) |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | <ul style="list-style-type: none"> • PD4-E591L42-E-65-1: 0007A120_h • PD4-E601L42-E-65-1: 000668A0_h • PD4-EB59CD-E-65-1: 000061A8_h |

| | |
|----------------|---|
| Subindex | 08 _h |
| Name | Torque Current Loop, Integral Gain (closed Loop) |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | <ul style="list-style-type: none"> • PD4-E591L42-E-65-1: 00001388_h • PD4-E601L42-E-65-1: 00002EE0_h • PD4-EB59CD-E-65-1: 00000BB8_h |

| | |
|----------------|---|
| Subindex | 09 _h |
| Name | Torque Current Loop, Proportional Gain (open Loop) |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | <ul style="list-style-type: none"> • PD4-E591L42-E-65-1: 00027100_h • PD4-E601L42-E-65-1: 00027100_h • PD4-EB59CD-E-65-1: 0003A980_h |

| | |
|----------------|--|
| Subindex | 0A _h |
| Name | Torque Current Loop, Integral Gain (open Loop) |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | <ul style="list-style-type: none"> • PD4-E591L42-E-65-1: 000055F0_h |

- PD4-E601L42-E-65-1: 00002710_h
- PD4-EB59CD-E-65-1: 0000AFC8_h

| | |
|----------------|---|
| Subindex | 0B _h |
| Name | Velocity Feed Forward Factor In Per Mille |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 000003E8 _h |

| | |
|----------------|----------------------------------|
| Subindex | 0C _h |
| Name | Acceleration Feed Forward Factor |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |

Description

- Subindex 00_h: Number of entries
- Subindex 01_h: Proportional component of the S-controller (position)
- Subindex 02_h: Integral component of the S-controller (position)
- Subindex 03_h: Proportional component of the V-controller (speed)
- Subindex 04_h: Integral component of the V-controller (speed)
- Subindex 05_h: (Closed loop) Proportional component of the current controller of the field-forming component
- Subindex 06_h: (Closed loop) Integral component of the current controller of the field-forming component
- Subindex 07_h: (Closed loop) Proportional component of the current controller of the torque-forming component
- Subindex 08_h: (Closed loop) Integral component of the current controller of the torque-forming component
- Subindex 09_h: (Open loop) Proportional component of the current controller of the field-building component
- Subindex 0A_h: (Open loop) Integral component of the current controller of the field-forming component
- Subindex 0B_h: (Closed loop) Speed feed forward in tenths of a percent. Default is 1000 and, thus, a factor of 1.
- Subindex 0C_h: (Closed loop) Acceleration feed forward. Default is 0 (feed forward inactive). It applies during deceleration as well.

3212h Motor Drive Flags

Function

This object is used to specify whether or not **auto setup** is to adapt the controller parameters. The direction of the rotating field can also be changed.



Note

Changes in subindex 02_h do not take effect until after the controller is restarted. Afterwards, **Auto setup** must again be performed.

Object description

| | |
|------------------|---|
| Index | 3212 _h |
| Object name | Motor Drive Flags |
| Object Code | ARRAY |
| Data type | INTEGER8 |
| Savable | yes, category: application |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1450 |
| Change history | Firmware version FIR-v1512: The number of entries was changed from 2 to 3. Firmware version FIR-v1738-B501312: "Name" entry changed from "Enable Legacy Power Mode" to "Reserved". |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 03 _h |

| | |
|----------------|-----------------|
| Subindex | 01 _h |
| Name | Reserved |
| Data type | INTEGER8 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00 _h |

| | |
|-------------|--------------------------|
| Subindex | 02 _h |
| Name | Override Field Inversion |
| Data type | INTEGER8 |
| Access | read / write |
| PDO mapping | no |

| | |
|----------------|----------------------------------|
| Allowed values | |
| Preset value | 00 _h |
| <hr/> | |
| Subindex | 03 _h |
| Name | Do Not Touch Controller Settings |
| Data type | INTEGER8 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00 _h |

Description

Valid values for subindex 02_h:

- Value = "0": Use default values of the firmware
- Value = "1": Force non-inversion of the rotating field (mathematically positive)
- Value = "-1": Force inversion of the rotating field (mathematically negative)

Valid values for subindex 03_h:

- Value = "0": **Auto setup** detects the motor type (stepper motor or BLDC motor) and uses the corresponding pre-configured parameter set.
- Value = "1": Perform **auto setup** with the values for the controller that were entered in object **3210_h** before the auto setup; the values in **3210_h** are not changed.

3220h Analog Inputs

Function

Displays the instantaneous values of the analog inputs in digits.

With object **3221_h**, the respective analog input can be configured as current or voltage input.

Object description

| | |
|------------------|-------------------|
| Index | 3220 _h |
| Object name | Analog Inputs |
| Object Code | ARRAY |
| Data type | INTEGER16 |
| Savable | no |
| Firmware version | FIR-v1426 |
| Change history | |

Value description

| | |
|-------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |

| | |
|----------------|-------------------|
| Allowed values | |
| Preset value | 02 _h |
| <hr/> | |
| Subindex | 01 _h |
| Name | Analogue Input 1 |
| Data type | INTEGER16 |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 0000 _h |
| <hr/> | |
| Subindex | 02 _h |
| Name | Analogue Input 2 |
| Data type | INTEGER16 |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 0000 _h |

Description

Formulas for converting from [digits] to the respective unit:

- Voltage input: $x \text{ digits} * 10 \text{ V} / 1024 \text{ digits}$
- Current input: $x \text{ digits} * 20 \text{ mA} / 1024 \text{ digits}$

3221h Analogue Inputs Control

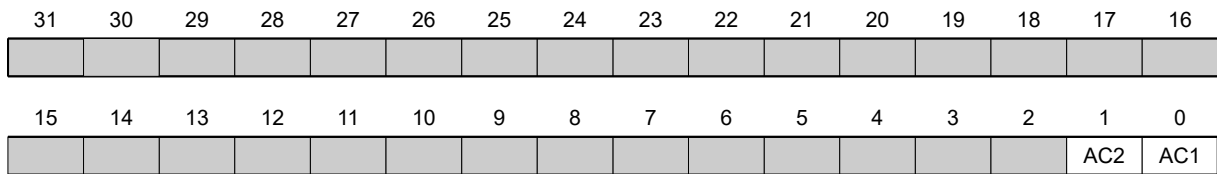
Function

With this object, an analog input can be switched from voltage measurement to current measurement.

Object description

| | |
|------------------|----------------------------|
| Index | 3221 _h |
| Object name | Analogue Inputs Control |
| Object Code | VARIABLE |
| Data type | INTEGER32 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| Firmware version | FIR-v1426 |
| Change history | |

Description



In general: If a bit is set to the value "0", the analog input measures the voltage; if the bit is set to the value "1", the current is measured.

AC1

Setting for analog input 1

AC2

Setting for analog input 2

3240h Digital Inputs Control

Function

With this object, digital inputs can be manipulated as described in chapter **Digital inputs and outputs**.

Object description

| | |
|------------------|--|
| Index | 3240 _h |
| Object name | Digital Inputs Control |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Firmware version | FIR-v1426 |
| Change history | Firmware version FIR-v1426: Subindex 01 _h : "Name" entry changed from "Special Function Disable" to "Special Function Enable" Firmware version FIR-v1512: The number of entries was changed from 8 to 9. |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 08 _h |

| | |
|-----------|-------------------------|
| Subindex | 01 _h |
| Name | Special Function Enable |
| Data type | UNSIGNED32 |

| | |
|----------------|-----------------------|
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 02 _h |
| Name | Function Inverted |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 03 _h |
| Name | Force Enable |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 04 _h |
| Name | Force Value |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 05 _h |
| Name | Raw Value |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 06 _h |
| Name | Input Range Select |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |

| | |
|----------------|-----------------------|
| Preset value | 00000000 _h |
| Subindex | 07 _h |
| Name | Differential Select |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |
| Subindex | 08 _h |
| Name | Routing Enable |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |

Description

The subindices have the following function:

- 01_h: This bit allows special functions of an input to be switched off (value "0") or on (value "1"). If input 1 is not used as, e.g., a negative limit switch, the special function must be switched off to prevent an erroneous response to the signal generator. The object has no effect on bits 16 to 31. The firmware evaluates the following bits:

- Bit 0: Negative limit switch
- Bit 1: Positive limit switch
- Bit 2: Home switch

If, for example, two limit switches and one home switch are used, bits 0–2 in **3240_h:01_h** must be set to "1".

- 02_h: This subindex switches from normally open logic (a logical high level at the input yields the value "1" in object **60FD_h**) to normally closed logic (the logical high level at the input yields the value "0"). This applies for the special functions (except for the clock and direction inputs) and for the normal inputs. If the bit has the value "0", normally open logic applies; for the value "1", normally closed logic applies. Bit 0 changes the logic of input 1, bit 1 changes the logic of input 2, etc.
- 03_h: This subindex switches on the software simulation of input values if the corresponding bit is set to "1". In this case, the actual values are no longer used in object **3240_h:04_h**, but rather the set values for the respective input. Bit 0 corresponds to input 1 here, bit 1 to input 2, etc.
- 04_h: This bit specifies the value that is to be read as the input value if the same bit was set in object **3240_h:03_h**.
- 05_h: This subindex always contains the read, unmodified input value.
- 06_h: (Input Range Select): This can be used to switch inputs – that are equipped with this function – from the switching threshold of 5 V (bit is "0") to the switching threshold of 24 V (bit is "1"). Bit 0 corresponds to input 1 here, bit 1 to input 2, etc.
- 07_h: With the inputs, this subindex switches between "single-ended input" (value "0" in the subindex) and "differential input" (value "1" in the subindex) if the input supports this function.
- 08_h: This subindex deactivates (value "0") or activates (value "1") the input routing.

3241h Digital Input Capture

Function

With this object, the encoder position can be noted automatically if a level change occurs at digital input 1.

Object description

| | |
|------------------|--|
| Index | 3241 _h |
| Object name | Digital Input Capture |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1446 |
| Change history | <p>Firmware version FIR-v1446: "Data type" entry changed from "UNSIGNED32" to "UNSIGNED8".</p> <p>Firmware version FIR-v1738-B501312: "Name" entry changed from "Encoder Raw Value" to "Sensor Raw Value".</p> <p>Firmware version FIR-v1748-B531667: "PDO mapping" table entry for subindex 00 changed from "no" to "TX-PDO".</p> <p>Firmware version FIR-v1748-B531667: "PDO mapping" table entry for subindex 01 changed from "RX-PDO" to "TX-PDO".</p> <p>Firmware version FIR-v1748-B531667: "PDO mapping" table entry for subindex 02 changed from "RX-PDO" to "TX-PDO".</p> <p>Firmware version FIR-v1748-B531667: "PDO mapping" table entry for subindex 03 changed from "RX-PDO" to "TX-PDO".</p> <p>Firmware version FIR-v1748-B531667: "PDO mapping" table entry for subindex 04 changed from "RX-PDO" to "TX-PDO".</p> |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 04 _h |

| | |
|-----------|-----------------|
| Subindex | 01 _h |
| Name | Control |
| Data type | UNSIGNED32 |

| | |
|----------------|-----------------------|
| Access | read / write |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 02 _h |
| Name | Capture Count |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 03 _h |
| Name | Value |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 04 _h |
| Name | Sensor Raw Value |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |

Description

- Subindex 01_h: This is used to select the type of level change:
 - Deactivate function: Value "0"
 - With rising edge: Value "1"
 - With falling edge: Value "2"
 - Both edges: Value "3"
- Subindex 02_h: Specifies the number of the noted level changes since the time the function was started; is reset to 0 if subindex 01_h is set to 1,2 or 3
- Subindex 03_h: Encoder position of the level change (in absolute user units from **6064_h**)
- Subindex 04_h: Encoder position of the level change

3242h Digital Input Routing

Function

This object determines the source of the input routing that ends in **60FD_h**.

Object description

| | |
|------------------|----------------------------|
| Index | 3242 _h |
| Object name | Digital Input Routing |
| Object Code | ARRAY |
| Data type | UNSIGNED8 |
| Savable | yes, category: application |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1504 |
| Change history | |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 24 _h |

| | |
|----------------|-----------------------------------|
| Subindex | 01 _h - 24 _h |
| Name | Input Source #1 - #36 |
| Data type | UNSIGNED8 |
| Access | read / write |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 00 _h |

Description

Subindex 01_h contains the source for bit 0 of object **60FD**. Subindex 02_h contains the source for bit 1 of object **60FD** and so on.

The number that is written in a subindex determines the source for the corresponding bit. The following table lists all possible signal sources.

| Number | | |
|--------|-----|--------------------|
| dec | hex | Signal source |
| 00 | 00 | Signal is always 0 |
| 01 | 01 | Physical input 1 |
| 02 | 02 | Physical input 2 |
| 03 | 03 | Physical input 3 |
| 04 | 04 | Physical input 4 |

| Number | | |
|--------|-----|--------------------------------|
| dec | hex | Signal source |
| 05 | 05 | Physical input 5 |
| 06 | 06 | Physical input 6 |
| 07 | 07 | Physical input 7 |
| 08 | 08 | Physical input 8 |
| 09 | 09 | Physical input 9 |
| 10 | 0A | Physical input 10 |
| 11 | 0B | Physical input 11 |
| 12 | 0C | Physical input 12 |
| 13 | 0D | Physical input 13 |
| 14 | 0E | Physical input 14 |
| 15 | 0F | Physical input 15 |
| 16 | 10 | Physical input 16 |
| 68 | 44 | Encoder input "A" |
| 69 | 45 | Encoder input "B" |
| 70 | 46 | Encoder input "Index" |
| 128 | 80 | Signal is always 1 |
| 129 | 81 | Inverted physical input 1 |
| 130 | 82 | Inverted physical input 2 |
| 131 | 83 | Inverted physical input 3 |
| 132 | 84 | Inverted physical input 4 |
| 133 | 85 | Inverted physical input 5 |
| 134 | 86 | Inverted physical input 6 |
| 135 | 87 | Inverted physical input 7 |
| 136 | 88 | Inverted physical input 8 |
| 137 | 89 | Inverted physical input 9 |
| 138 | 8A | Inverted physical input 10 |
| 139 | 8B | Inverted physical input 11 |
| 140 | 8C | Inverted physical input 12 |
| 141 | 8D | Inverted physical input 13 |
| 142 | 8E | Inverted physical input 14 |
| 143 | 8F | Inverted physical input 15 |
| 144 | 90 | Inverted physical input 16 |
| 196 | C4 | Inverted encoder input "A" |
| 197 | C5 | Inverted encoder input "B" |
| 198 | C6 | Inverted encoder input "Index" |

3243h Digital Input Homing Capture

Function

With this object, the encoder position can be noted automatically if a level change occurs at the digital input that is used for the home switch.

Object description

| | |
|------------------|------------------------------|
| Index | 3243 _h |
| Object name | Digital Input Homing Capture |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1738-B501312 |
| Change history | |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 04 _h |

| | |
|----------------|-----------------------|
| Subindex | 01 _h |
| Name | Control |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |

| | |
|----------------|-----------------------|
| Subindex | 02 _h |
| Name | Capture Count |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |

| | |
|-----------|-----------------|
| Subindex | 03 _h |
| Name | Value |
| Data type | UNSIGNED32 |
| Access | read / write |

| | |
|----------------|-----------------------|
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 04 _h |
| Name | Sensor Raw Value |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |

Description

- Subindex 01_h: This is used to select the type of level change:
 - Deactivate function: Value "0"
 - With rising edge: Value "1"
 - With falling edge: Value "2"
 - Both edges: Value "3"
- Subindex 02_h: Specifies the number of the noted level changes since the time the function was started; is reset to 0 if subindex 01_h is set to 1,2 or 3
- Subindex 03_h: Encoder position of the level change (in absolute user units from **6064_h**)
- Subindex 04_h: Encoder position of the level change

3250h Digital Outputs Control

Function

This object can be used to control the digital outputs as described in chapter " **Digital inputs and outputs**".

The following applies for all subindices:

- Bits 0 to 15 control the special functions.
- Bits 16 to 31 control the level of the outputs.

Object description

| | |
|------------------|---|
| Index | 3250 _h |
| Object name | Digital Outputs Control |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Firmware version | FIR-v1426 |
| Change history | Firmware version FIR-v1426: Subindex 01 _h : "Name" entry changed from "Special Function Disable" to "Special Function Enable" Firmware version FIR-v1446: "Name" entry changed from "Special Function Enable" to "No Function". |

Firmware version FIR-v1512: The number of entries was changed from 6 to 9.

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 08 _h |

| | |
|----------------|-----------------------|
| Subindex | 01 _h |
| Name | No Function |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |

| | |
|----------------|-----------------------|
| Subindex | 02 _h |
| Name | Function Inverted |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |

| | |
|----------------|-----------------------|
| Subindex | 03 _h |
| Name | Force Enable |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |

| | |
|----------------|-----------------|
| Subindex | 04 _h |
| Name | Force Value |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |

| | |
|----------------|-----------------------|
| Preset value | 00000000 _h |
| Subindex | 05 _h |
| Name | Raw Value |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |
| Subindex | 06 _h |
| Name | Reserved1 |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |
| Subindex | 07 _h |
| Name | Reserved2 |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |
| Subindex | 08 _h |
| Name | Routing Enable |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |

Description

The subindices have the following function:

- 01_h: No function.
- 02_h: This subindex is used to invert the logic (from normally closed logic to normally open logic).
- 03_h: This subindex is used to force the output value if the bit has the value "1". The level of the output is defined in subindex 4_h.
- 04_h: This subindex is used to define the level to be applied to the output. The value "0" returns a logical low level at the digital output; the value "1", on the other hand, returns a logical high level.
- 05_h: The bit combination applied to the outputs is stored in this subindex.

3252h Digital Output Routing

Function

This object assigns a signal source to an output; this signal source can be controlled with **60FE_h**.

Object description

| | |
|------------------|----------------------------|
| Index | 3252 _h |
| Object name | Digital Output Routing |
| Object Code | ARRAY |
| Data type | UNSIGNED16 |
| Savable | yes, category: application |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1540 |
| Change history | |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 05 _h |

| | |
|----------------|-----------------------------------|
| Subindex | 01 _h - 05 _h |
| Name | Output Control #1 - #5 |
| Data type | UNSIGNED16 |
| Access | read / write |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 1080 _h |

3320h Read Analogue Input

Function

Displays the instantaneous values of the analog inputs in user-defined units.

Object description

| | |
|-------|-------------------|
| Index | 3320 _h |
|-------|-------------------|

| | |
|------------------|---------------------|
| Object name | Read Analogue Input |
| Object Code | ARRAY |
| Data type | INTEGER32 |
| Savable | no |
| Firmware version | FIR-v1426 |
| Change history | |

Value description

| | |
|----------------|---------------------------|
| Subindex | 00 _h |
| Name | Number Of Analogue Inputs |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 02 _h |

| | |
|----------------|-----------------------|
| Subindex | 01 _h |
| Name | Analogue Input 1 |
| Data type | INTEGER32 |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |

| | |
|----------------|-----------------------|
| Subindex | 02 _h |
| Name | Analogue Input 2 |
| Data type | INTEGER32 |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |

Description

The user-defined units are made up of offset (**3321_h**) and pre-scaling value (**3322_h**). If both object entries are still set to the default values, the value in **3320_h** is specified in the "ADC digits" unit.

Formula for converting from digits to the respective unit:

- Voltage input: $x \text{ digits} * 10 \text{ V} / 1024 \text{ digits}$
- Current input: $x \text{ digits} * 20 \text{ mA} / 1024 \text{ digits}$

The following applies for the sub-entries:

- Subindex 00_h: Number of analog inputs
- Subindex 01_h: Analog value 1
- Subindex 02_h: Analog value 2

3321h Analogue Input Offset

Function

Offset that is added to the read analog value (3320_h) before dividing by the divisor from object 3322_h.

Object description

| | |
|------------------|----------------------------|
| Index | 3321 _h |
| Object name | Analogue Input Offset |
| Object Code | ARRAY |
| Data type | INTEGER32 |
| Savable | yes, category: application |
| Firmware version | FIR-v1426 |
| Change history | |

Value description

| | |
|----------------|---------------------------|
| Subindex | 00 _h |
| Name | Number Of Analogue Inputs |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 02 _h |

| | |
|----------------|-----------------------|
| Subindex | 01 _h |
| Name | Analogue Input 1 |
| Data type | INTEGER32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

| | |
|----------------|-----------------------|
| Subindex | 02 _h |
| Name | Analogue Input 2 |
| Data type | INTEGER32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

Description

- Subindex 00_h: Number of offsets
- Subindex 01_h: Offset for analog input 1

- Subindex 02_h: Offset for analog input 2

3322h Analogue Input Pre-scaling

Function

Value by which the read analog value (3320_h, 3321_h) is divided before it is written in object 3320_h.

Object description

| | |
|------------------|----------------------------|
| Index | 3322 _h |
| Object name | Analogue Input Pre-scaling |
| Object Code | ARRAY |
| Data type | INTEGER32 |
| Savable | yes, category: application |
| Firmware version | FIR-v1426 |
| Change history | |

Value description

| | |
|----------------|---------------------------|
| Subindex | 00 _h |
| Name | Number Of Analogue Inputs |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 02 _h |

| | |
|----------------|-------------------------------|
| Subindex | 01 _h |
| Name | Analogue Input 1 |
| Data type | INTEGER32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | All values permitted except 0 |
| Preset value | 00000001 _h |

| | |
|----------------|-------------------------------|
| Subindex | 02 _h |
| Name | Analogue Input 2 |
| Data type | INTEGER32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | All values permitted except 0 |
| Preset value | 00000001 _h |

Description

The subindices contain:

- Subindex 00_h: Number of divisors
- Subindex 01_h: Divisor for analog input 1
- Subindex 02_h: Divisor for analog input 2

33A0h Feedback Incremental A/B/I 1

Function

Contains configuration values for the first incremental encoder. The values are determined by the **Auto setup**.

Object description

| | |
|------------------|------------------------------|
| Index | 33A0 _h |
| Object name | Feedback Incremental A/B/I 1 |
| Object Code | ARRAY |
| Data type | UNSIGNED16 |
| Savable | yes, category: tuning |
| Access | read only |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1738-B501312 |
| Change history | |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 02 _h |

| | |
|----------------|-------------------|
| Subindex | 01 _h |
| Name | Configuration |
| Data type | UNSIGNED16 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 8001 _h |

| | |
|----------|-----------------|
| Subindex | 02 _h |
|----------|-----------------|

| | |
|----------------|-------------------|
| Name | Alignment |
| Data type | UNSIGNED16 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 0000 _h |

Description

The subindices have the following function:

- 00_h (Configuration): The following bits have a meaning:
 - Bit 0: Value = "0": The encoder does not have an index. Value = "1": Encoder index found and is to be used.
 - Bit 15: Value = "1": The encoder is a single-turn absolute encoder.
- 01_h (Alignment): This value specifies the offset between the index of the encoder and the electric field.
The exact determination is only possible via **auto setup**. The presence of this value is necessary for *closed loop* mode with encoder.

3700h Deviation Error Option Code

Function

The object contains the action that is to be executed if a following or slippage error is triggered.

Object description

| | |
|------------------|--|
| Index | 3700 _h |
| Object name | Deviation Error Option Code |
| Object Code | VARIABLE |
| Data type | INTEGER16 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | FFFF _h |
| Firmware version | FIR-v1426 |
| Change history | Firmware version FIR-v1738-B501312: "Object name" entry changed from "Following Error Option Code" to "Deviation Error Option Code". |

Description

| Value | Description |
|---------------|--|
| -32768 ... -2 | Reserved |
| -1 | No reaction |
| 0 | Immediate stop |
| 1 | Braking with "slow down ramp" (deceleration (deceleration ramp) depending on operating mode) |

| Value | Description |
|-------------|---|
| 2 | Braking with "quick stop ramp" (deceleration (deceleration ramp) depending on operating mode) |
| 3 ... 32767 | Reserved |

4012h HW Information

Function

This object contains information about the hardware.

Object description

| | |
|------------------|-------------------|
| Index | 4012 _h |
| Object name | HW Information |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | no |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1540 |
| Change history | |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 01 _h |

| | |
|----------------|-----------------------|
| Subindex | 01 _h |
| Name | EEPROM Size In Bytes |
| Data type | UNSIGNED32 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

Description

Subindex 01: Contains the size of the connected EEPROM in bytes. The value "0" means that no EEPROM is connected.

4013h HW Configuration

Function

This object is used to set certain hardware configurations.

Object description

| | |
|------------------|----------------------------|
| Index | 4013 _h |
| Object name | HW Configuration |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1540 |
| Change history | |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 01 _h |

| | |
|----------------|-----------------------|
| Subindex | 01 _h |
| Name | HW Configuration #1 |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |

Description

Bit 0: reserved

4014h Operating Conditions

Function

This object is used to read out the current environment values for the controller.

Object description

| | |
|------------------|---|
| Index | 4014 _h |
| Object name | Operating Conditions |
| Object Code | ARRAY |
| Data type | INTEGER32 |
| Savable | no |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1540 |
| Change history | <p>Firmware version FIR-v1650-B472161: "Access" table entry for subindex 01 changed from "read/write" to "read only".</p> <p>Firmware version FIR-v1650-B472161: "Access" table entry for subindex 02 changed from "read/write" to "read only".</p> <p>Firmware version FIR-v1650-B472161: "Name" entry changed from "Temperature PCB [d?C]" to "Temperature PCB [Celsius * 10]".</p> <p>Firmware version FIR-v1650-B472161: "Access" table entry for subindex 03 changed from "read/write" to "read only".</p> <p>Firmware version FIR-v1738-B501312: The number of entries was changed from 4 to 6.</p> |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 05 _h |

| | |
|----------------|-----------------------|
| Subindex | 01 _h |
| Name | Voltage UB Power [mV] |
| Data type | INTEGER32 |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |

| | |
|----------------|---|
| Subindex | 02 _h |
| Name | Voltage UB Logic [mV] |
| Data type | INTEGER32 |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |
| Subindex | 03 _h |
| Name | Temperature PCB [Celsius * 10] |
| Data type | INTEGER32 |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |
| Subindex | 04 _h |
| Name | Temperature Motor [Celsius * 10] |
| Data type | INTEGER32 |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |
| Subindex | 05 _h |
| Name | Temperature Microcontroller Chip [Celsius * 10] |
| Data type | INTEGER32 |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |

Description

The subindices contain:

- 01_h: Current voltage supply voltage in [mV]
- 02_h: Current logic voltage in [mV]
- 03_h: Current temperature of the control board in [d°C] (tenths of degree)
- 04_h: Reserves
- 05_h: Reserves

4015h Special Drive Modes

Function

With this object, you can switch the *special drive modes* off or on. See chapter **Special drive modes (clock-direction and analog speed)**.

Object description

| | |
|------------------|----------------------------|
| Index | 4015 _h |
| Object name | Special Drive Modes |
| Object Code | ARRAY |
| Data type | UNSIGNED8 |
| Savable | yes, category: application |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1738-B501312 |
| Change history | |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 02 _h |

| | |
|----------------|----------------------------------|
| Subindex | 01 _h |
| Name | Special Drive Mode Configuration |
| Data type | UNSIGNED8 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00 _h |

| | |
|----------------|-----------------------------|
| Subindex | 02 _h |
| Name | Virtual Config Switch Value |
| Data type | UNSIGNED8 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00 _h |

Description

The subindices have the following functions:

- 01_h:
 - Value="0"_h: The *special drive modes* are switched off
 - Value="2"_h: The *special drive modes* are switched on and the mode is set in subindex 02_h.
- 02_h: Defines the used mode.

4016h Factory Settings

Function

This object indicates whether **Auto setup** was executed in production.

Object description

| | |
|------------------|-------------------|
| Index | 4016 _h |
| Object name | Factory Settings |
| Object Code | ARRAY |
| Data type | UNSIGNED8 |
| Savable | no |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1738-B501312 |
| Change history | |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 01 _h |

| | |
|----------------|------------------------|
| Subindex | 01 _h |
| Name | Factory Autosetup Done |
| Data type | UNSIGNED8 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00 _h |

Description

Valid values for subindex 01_n:

- Value = "0": *Auto setup* was not executed.
- Value = "1": *Auto setup* was executed.

4040h Drive Serial Number

Function

This object contains the serial number of the controller.

Object description

| | |
|------------------|---------------------|
| Index | 4040 _h |
| Object name | Drive Serial Number |
| Object Code | VARIABLE |
| Data type | VISIBLE_STRING |
| Savable | no |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 0 |
| Firmware version | FIR-v1450 |
| Change history | |

4041h Device Id

Function

This object contains the ID of the device.

Object description

| | |
|------------------|-------------------|
| Index | 4041 _h |
| Object name | Device Id |
| Object Code | VARIABLE |
| Data type | OCTET_STRING |
| Savable | no |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 0 |
| Firmware version | FIR-v1540 |
| Change history | |

603Fh Error Code

Function

This object returns the error code of the last error that occurred.

It corresponds to the lower 16 bits of object **1003_h**. For the description of the error codes, refer to object **1003_h**.

Object description

| | |
|------------------|-------------------|
| Index | 603F _h |
| Object name | Error Code |
| Object Code | VARIABLE |
| Data type | UNSIGNED16 |
| Savable | no |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 0000 _h |
| Firmware version | FIR-v1426 |
| Change history | |

Description

For the meaning of the error, see object **1003_h** (Pre-defined Error Field).

6040h Controlword

Function

This object controls the **CiA 402 Power State Machine**.

Object description

| | |
|------------------|--|
| Index | 6040 _h |
| Object name | Controlword |
| Object Code | VARIABLE |
| Data type | UNSIGNED16 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 0000 _h |
| Firmware version | FIR-v1426 |
| Change history | Firmware version FIR-v1626: "Savable" entry changed from "no" to "yes, category: application". |

Description

Parts of the object are, with respect to function, dependent on the currently selected mode.

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|-----|------|----|---|---------|---|----|----|----|----|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | | | | OMS | HALT | FR | | OMS [3] | | EO | QS | EV | SO |

SO (Switched On)

Value = "1": Switches to the "Switched on" state

EV (Enable Voltage)

Value = "1": Switches to the "Enable voltage" state

QS (Quick Stop)

Value = "0": Switches to the "Quick stop" state

EO (Enable Operation)

Value = "1": Switches to the "Enable operation" state

OMS (Operation Mode Specific)

Meaning is dependent on the selected operating mode

FR (Fault Reset)

Resets an error (if possible)

HALT

Value = "1": Triggers a halt; valid in the following modes:

- **Profile Position**
- **Velocity**
- **Profile Velocity**
- **Profile Torque**
- **Interpolated Position Mode**

6041h Statusword

Function

This object returns information about the status of the **CiA 402 Power State Machine**.

Object description

| | |
|------------------|-------------------|
| Index | 6041 _h |
| Object name | Statusword |
| Object Code | VARIABLE |
| Data type | UNSIGNED16 |
| Savable | no |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 0000 _h |
| Firmware version | FIR-v1426 |
| Change history | |

Description

Parts of the object are, with respect to function, dependent on the currently selected mode.

| | | | | | | | | | | | | | | | |
|-----|----|---------|-----|------|-----|------|------|-----|----|----|-------|----|----|------|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CLA | | OMS [2] | ILA | TARG | REM | SYNC | WARN | SOD | QS | VE | FAULT | OE | SO | RTSO | |

RTSO (Ready To Switch On)

Value = "1": Controller is in the "Ready to switch on" state

SO (Switched On)

Value = "1": Controller is in the "Switched on" state

OE (Operation Enabled)

Value = "1": Controller is in the "Operation enabled" state

FAULT

Error occurred

VE (Voltage Enabled)

Voltage applied

QS (Quick Stop)

Value = "0": Controller is in the "Quick stop" state

SOD (Switched On Disabled)

Value = "1": Controller is in the "Switched on disabled" state

WARN (Warning)

Value = "1": Warning

SYNC (synchronization)

Value = "1": Controller is in sync with the fieldbus; value = "0": Controller is not in sync with the fieldbus

REM (Remote)

Remote (value of the bit is always "1")

TARG

Target reached

ILA (Internal Limit Reached)

Limit exceeded

OMS (Operation Mode Specific)

Meaning is dependent on the selected operating mode

CLA (Closed Loop Active)

Value = "1": The controller is in the *Operation enabled* state and the **Closed Loop** is activated.

6042h VI Target Velocity

Function

Specifies the target speed in **user-defined units** for **Velocity** mode.

Object description

| | |
|------------------|--|
| Index | 6042 _h |
| Object name | VI Target Velocity |
| Object Code | VARIABLE |
| Data type | INTEGER16 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00C8 _h |
| Firmware version | FIR-v1426 |
| Change history | Firmware version FIR-v1626: "Savable" entry changed from "no" to "yes, category: application". |

6043h VI Velocity Demand

Function

Speed specification in **user-defined units** for the controller in **Velocity** mode.

Object description

| | |
|------------------|--------------------|
| Index | 6043 _h |
| Object name | VI Velocity Demand |
| Object Code | VARIABLE |
| Data type | INTEGER16 |
| Savable | no |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 0000 _h |
| Firmware version | FIR-v1426 |
| Change history | |

6044h VI Velocity Actual Value

Function

Specifies the current actual speed in **user-defined units** in **Velocity** mode.

Object description

| | |
|-------------|--------------------------|
| Index | 6044 _h |
| Object name | VI Velocity Actual Value |
| Object Code | VARIABLE |
| Data type | INTEGER16 |

| | |
|------------------|-------------------|
| Savable | no |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 0000 _h |
| Firmware version | FIR-v1426 |
| Change history | |

6046h VI Velocity Min Max Amount

Function

This object can be used to set the minimum speed and maximum speed in **user-defined units**.

Object description

| | |
|------------------|----------------------------|
| Index | 6046 _h |
| Object name | VI Velocity Min Max Amount |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Firmware version | FIR-v1426 |
| Change history | |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 02 _h |

| | |
|----------------|-----------------------|
| Subindex | 01 _h |
| Name | MinAmount |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |

| | |
|-----------|-----------------|
| Subindex | 02 _h |
| Name | MaxAmount |
| Data type | UNSIGNED32 |

| | |
|----------------|-----------------------|
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00004E20 _h |

Description

Subindex 1 contains the minimum speed.

Subindex 2 contains the maximum speed.

If the value of the target speed (object **6042_h**) specified here is less than the minimum speed, the minimum speed applies and bit 11 (Internal Limit Reached) in **6041_h Statusword_h** is set.

A target speed greater than the maximum speed sets the speed to the maximum speed and bit 11 (Internal Limit Reached) in **6041_h Statusword_h** is set.

6048h VI Velocity Acceleration

Function

Sets the acceleration ramp in Velocity Mode (see **Velocity**).

Object description

| | |
|------------------|------------------------------------|
| Index | 6048 _h |
| Object name | VI Velocity Acceleration |
| Object Code | RECORD |
| Data type | VELOCITY_ACCELERATION_DECELERATION |
| Savable | yes, category: application |
| Firmware version | FIR-v1426 |
| Change history | |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 02 _h |

| | |
|----------------|-----------------|
| Subindex | 01 _h |
| Name | DeltaSpeed |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |

| | |
|----------------|-----------------------|
| Preset value | 000001F4 _h |
| Subindex | 02 _h |
| Name | DeltaTime |
| Data type | UNSIGNED16 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 0001 _h |

Description

The acceleration is specified as a fraction in user-defined units:

Speed change per change in time.

Subindex 01_h: Contains the change in speed.

Subindex 02_h: Contains the change in time.

6049h VI Velocity Deceleration

Function

Sets the deceleration (deceleration ramp) in Velocity Mode (see **Velocity**).

Object description

| | |
|------------------|------------------------------------|
| Index | 6049 _h |
| Object name | VI Velocity Deceleration |
| Object Code | RECORD |
| Data type | VELOCITY_ACCELERATION_DECELERATION |
| Savable | yes, category: application |
| Firmware version | FIR-v1426 |
| Change history | |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 02 _h |
| Subindex | 01 _h |
| Name | DeltaSpeed |

| | |
|----------------|-----------------------|
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 000001F4 _h |
| <hr/> | |
| Subindex | 02 _h |
| Name | DeltaTime |
| Data type | UNSIGNED16 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 0001 _h |

Description

The deceleration is specified as a fraction in user-defined units:

Speed change per change in time.

Subindex 01_h: Contains the change in speed.

Subindex 02_h: Contains the change in time.

604Ah VI Velocity Quick Stop

Function

This object defines the deceleration (deceleration ramp) if the Quick Stop state is initiated in **Velocity Mode**.

Object description

| | |
|------------------|------------------------------------|
| Index | 604A _h |
| Object name | VI Velocity Quick Stop |
| Object Code | RECORD |
| Data type | VELOCITY_ACCELERATION_DECELERATION |
| Savable | yes, category: application |
| Firmware version | FIR-v1426 |
| Change history | |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |

| | |
|----------------|-----------------------|
| Preset value | 02 _h |
| Subindex | 01 _h |
| Name | DeltaSpeed |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00001388 _h |
| Subindex | 02 _h |
| Name | DeltaTime |
| Data type | UNSIGNED16 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 0001 _h |

Description

The deceleration is specified as a fraction in user-defined units:

Speed change per change in time.

Subindex 01_h: Contains the change in speed.

Subindex 02_h: Contains the change in time.

604Ch VI Dimension Factor

Function

The unit for speed values is defined here for the objects associated with **Velocity Mode**.

Object description

| | |
|------------------|----------------------------|
| Index | 604C _h |
| Object name | VI Dimension Factor |
| Object Code | ARRAY |
| Data type | INTEGER32 |
| Savable | yes, category: application |
| Firmware version | FIR-v1426 |
| Change history | |

Value description

| | |
|----------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |

| | |
|----------------|---------------------------------|
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 02 _h |
| <hr/> | |
| Subindex | 01 _h |
| Name | VI Dimension Factor Numerator |
| Data type | INTEGER32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000001 _h |
| <hr/> | |
| Subindex | 02 _h |
| Name | VI Dimension Factor Denominator |
| Data type | INTEGER32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000001 _h |

Description

Subindex 1 contains the numerator (multiplier) and subindex 2 contains the denominator (divisor) with which the internal speed values are converted to revolutions per minute. If, for example, subindex 1 is set to the value "60" and subindex 2 is set to the value "1", the speed is specified in revolutions per second (60 revolutions per 1 minute).

605Ah Quick Stop Option Code

Function

The object contains the action that is to be executed on a transition of the **CiA 402 Power State Machine** to the *Quick Stop* state.

Object description

| | |
|----------------|----------------------------|
| Index | 605A _h |
| Object name | Quick Stop Option Code |
| Object Code | VARIABLE |
| Data type | INTEGER16 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 0001 _h |

| | |
|------------------|-----------|
| Firmware version | FIR-v1426 |
| Change history | |

Description

| Value | Description |
|---------------|--|
| -32768 ... -1 | Reserved |
| 0 | Immediate stop |
| 1 | Braking with "slow down ramp" (deceleration (deceleration ramp) depending on operating mode) and subsequent state change to "Switch on disabled" |
| 2 | Braking with "quick stop ramp" and subsequent state change to "Switch on disabled" |
| 3 ... 32767 | Reserved |

605Bh Shutdown Option Code

Function

This object contains the action that is to be executed on a transition of the **CiA 402 Power State Machine** from the *Operation enabled* state to the *Ready to switch on* state.

Object description

| | |
|------------------|----------------------------|
| Index | 605B _h |
| Object name | Shutdown Option Code |
| Object Code | VARIABLE |
| Data type | INTEGER16 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 0001 _h |
| Firmware version | FIR-v1426 |
| Change history | |

Description

| Value | Description |
|---------------|--|
| -32768 ... -1 | Reserved |
| 0 | Immediate stop |
| 1 | Braking with "slow down ramp" (deceleration (deceleration ramp) depending on operating mode) and subsequent state change to "Switch on disabled" |
| 2 ... 32767 | Reserved |

605Ch Disable Option Code

Function

This object contains the action that is to be executed on a transition of the **CiA 402 Power State Machine** from the *Operation enabled* state to the *Switched on* state.

Object description

| | |
|------------------|----------------------------|
| Index | 605C _h |
| Object name | Disable Option Code |
| Object Code | VARIABLE |
| Data type | INTEGER16 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 0001 _h |
| Firmware version | FIR-v1426 |
| Change history | |

Description

| Value | Description |
|---------------|--|
| -32768 ... -1 | Reserved |
| 0 | Immediate stop |
| 1 | Braking with "slow down ramp" (deceleration (deceleration ramp) depending on operating mode) and subsequent state change to "Switch on disabled" |
| 2 ... 32767 | Reserved |

605Dh Halt Option Code

Function

The object contains the action that is to be executed if bit 8 (Halt) is set in controlword **6040_h**.

Object description

| | |
|------------------|----------------------------|
| Index | 605D _h |
| Object name | Halt Option Code |
| Object Code | VARIABLE |
| Data type | INTEGER16 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 0001 _h |
| Firmware version | FIR-v1426 |

Change history

Description

| Value | Description |
|--------------|---|
| -32768 ... 0 | Reserved |
| 1 | Braking with "slow down ramp" (deceleration (deceleration ramp) depending on operating mode) |
| 2 | Braking with "quick stop ramp" (deceleration (deceleration ramp) depending on operating mode) |
| 3 ... 32767 | Reserved |

605Eh Fault Option Code

Function

The object contains the action specifying how the motor is to be brought to a standstill in case of an error.

Object description

| | |
|------------------|----------------------------|
| Index | 605E _h |
| Object name | Fault Option Code |
| Object Code | VARIABLE |
| Data type | INTEGER16 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 0002 _h |
| Firmware version | FIR-v1426 |
| Change history | |

Description

| Value | Description |
|---------------|---|
| -32768 ... -1 | Reserved |
| 0 | Immediate stop |
| 1 | Braking with "slow down ramp" (deceleration (deceleration ramp) depending on operating mode) |
| 2 | Braking with "quick stop ramp" (deceleration (deceleration ramp) depending on operating mode) |
| 3 ... 32767 | Reserved |

6060h Modes Of Operation

Function

The desired operating mode is entered in this object.

Object description

| | |
|------------------|--|
| Index | 6060 _h |
| Object name | Modes Of Operation |
| Object Code | VARIABLE |
| Data type | INTEGER8 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00 _h |
| Firmware version | FIR-v1426 |
| Change history | Firmware version FIR-v1626: "Savable" entry changed from "no" to "yes, category: application". |

Description

| Mode | Description |
|------|----------------------------------|
| -2 | Auto setup |
| -1 | Clock-direction mode |
| 0 | No mode change/no mode assigned |
| 1 | Profile Position Mode |
| 2 | Velocity Mode |
| 3 | Profile Velocity Mode |
| 4 | Profile Torque Mode |
| 5 | Reserved |
| 6 | Homing Mode |
| 7 | Interpolated Position Mode |
| 8 | Cyclic Synchronous Position Mode |
| 9 | Cyclic Synchronous Velocity Mode |
| 10 | Cyclic Synchronous Torque Mode |

6061h Modes Of Operation Display

Function

Indicates the current operating mode. See also **6060h Modes Of Operation**.

Object description

| | |
|-------|-------------------|
| Index | 6061 _h |
|-------|-------------------|

| | |
|------------------|----------------------------|
| Object name | Modes Of Operation Display |
| Object Code | VARIABLE |
| Data type | INTEGER8 |
| Savable | no |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 00 _h |
| Firmware version | FIR-v1426 |
| Change history | |

6062h Position Demand Value

Function

Indicates the current demand position in **user-defined units**.

Object description

| | |
|------------------|-----------------------|
| Index | 6062 _h |
| Object name | Position Demand Value |
| Object Code | VARIABLE |
| Data type | INTEGER32 |
| Savable | no |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |
| Firmware version | FIR-v1426 |
| Change history | |

6063h Position Actual Internal Value

Function

Contains the current rotary encoder position in increments. Unlike objects **6062_h** and **6064_h**, this value is not set to "0" following a **Homing** operation.



Note

If the encoder resolution in object **608F_h** = 0, the numerical values of this object are invalid.

Object description

| | |
|-------------|--------------------------------|
| Index | 6063 _h |
| Object name | Position Actual Internal Value |
| Object Code | VARIABLE |
| Data type | INTEGER32 |

| | |
|------------------|-----------------------|
| Savable | no |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |
| Firmware version | FIR-v1426 |
| Change history | |

6064h Position Actual Value

Function

Contains the current actual position in **user-defined units**.

Object description

| | |
|------------------|-----------------------|
| Index | 6064 _h |
| Object name | Position Actual Value |
| Object Code | VARIABLE |
| Data type | INTEGER32 |
| Savable | no |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |
| Firmware version | FIR-v1426 |
| Change history | |

6065h Following Error Window

Function

Defines the maximum allowed **following error** in **user-defined units** symmetrically to the **demand position**.

Object description

| | |
|------------------|----------------------------|
| Index | 6065 _h |
| Object name | Following Error Window |
| Object Code | VARIABLE |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000100 _h |
| Firmware version | FIR-v1426 |

| | |
|----------------|--|
| Change history | Firmware version FIR-v1504: "Savable" entry changed from "no" to "yes, category: application". |
|----------------|--|

Description

If the actual position deviates so much from the demand position that the value of this object is exceeded, bit 13 in object **6041_h** is set. The deviation must last longer than the time in object **6066_h**.

If the value of the "Following Error Window" is set to "FFFFFFFF"_h, following error monitoring is switched off.

A reaction to the following error can be set in object **3700_h**. If a reaction is defined, an error is also entered in object **1003_h**.

6066h Following Error Time Out

Function

Time in milliseconds until a larger following error results in an error message.

Object description

| | |
|------------------|--|
| Index | 6066 _h |
| Object name | Following Error Time Out |
| Object Code | VARIABLE |
| Data type | UNSIGNED16 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 0064 _h |
| Firmware version | FIR-v1426 |
| Change history | Firmware version FIR-v1504: "Savable" entry changed from "no" to "yes, category: application". |

Description

If the actual position deviates so much from the demand position that the value of object **6065_h** is exceeded, bit 13 in object **6041_h** is set. The deviation must persist for longer than the time defined in this object.

A reaction to the following error can be set in object **3700_h**. If a reaction is defined, an error is also entered in object **1003_h**.

6067h Position Window

Function

Specifies a range symmetrical to the target position within which that target is considered having been met in modes **Profile Position** and **Interpolated Position Mode**.

Object description

| | |
|------------------|--|
| Index | 6067 _h |
| Object name | Position Window |
| Object Code | VARIABLE |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 0000000A _h |
| Firmware version | FIR-v1426 |
| Change history | Firmware version FIR-v1504: "Savable" entry changed from "no" to "yes, category: application". |

Description

If the current position deviates from the target position by less than the value of this object, bit 10 in object **6041_h** is set. The condition must be satisfied for longer than the time defined in object **6066_h**.

If the value is set to "FFFFFFFF"_h, monitoring is switched off.

6068h Position Window Time

Function

The current position must be within the "Position Window" (**6067_h**) for this time in milliseconds for the target position to be considered having been met in the **Profile Position** and **Interpolated Position Mode** modes.

Object description

| | |
|------------------|--|
| Index | 6068 _h |
| Object name | Position Window Time |
| Object Code | VARIABLE |
| Data type | UNSIGNED16 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 0064 _h |
| Firmware version | FIR-v1426 |
| Change history | Firmware version FIR-v1504: "Savable" entry changed from "no" to "yes, category: application". |

Description

If the current position deviates from the target position by less than the value of object **6067_h**, bit 10 in object **6041_h** is set. The condition must be satisfied for longer than the time defined in object **6066_h**.

606Bh Velocity Demand Value

Function

Speed specification in **user-defined units** for the controller in **Profile Velocity Mode**.

Object description

| | |
|------------------|-----------------------|
| Index | 606B _h |
| Object name | Velocity Demand Value |
| Object Code | VARIABLE |
| Data type | INTEGER32 |
| Savable | no |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |
| Firmware version | FIR-v1426 |
| Change history | |

Description

This object contains the output of the ramp generator, which simultaneously serves as the preset value for the speed controller.

606Ch Velocity Actual Value

Function

Current actual speed in **user-defined units**.

Object description

| | |
|------------------|-----------------------|
| Index | 606C _h |
| Object name | Velocity Actual Value |
| Object Code | VARIABLE |
| Data type | INTEGER32 |
| Savable | no |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |
| Firmware version | FIR-v1426 |
| Change history | |

606Dh Velocity Window

Function

Specifies a symmetrical range relative to the target speed within which the target is considered having been met in the **Profile Velocity** mode.

Object description

| | |
|------------------|--|
| Index | 606D _h |
| Object name | Velocity Window |
| Object Code | VARIABLE |
| Data type | UNSIGNED16 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 001E _h |
| Firmware version | FIR-v1426 |
| Change history | Firmware version FIR-v1614: "Savable" entry changed from "no" to "yes, category: application". |

Description

If the current speed deviates from the set speed by less than the value of this object, bit 10 in object **6041_h** is set. The condition must be satisfied for longer than the time defined in object **6066_h** (see also **statusword in Profile Velocity Mode**).

606Eh Velocity Window Time

Function

The current speed must be within the "Velocity Window" (**606D_h**) for this time (in milliseconds) for the target to be considered having been met.

Object description

| | |
|------------------|--|
| Index | 606E _h |
| Object name | Velocity Window Time |
| Object Code | VARIABLE |
| Data type | UNSIGNED16 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 0000 _h |
| Firmware version | FIR-v1426 |
| Change history | Firmware version FIR-v1614: "Savable" entry changed from "no" to "yes, category: application". |

Description

Description

If the current speed deviates from the set speed by less than the value of object **606D_h**, bit 10 in object **6041_h** is set. The condition must be satisfied for longer than the time defined in object **6066** (see also **statusword in Profile Velocity Mode**).

6071h Target Torque

Function

This object contains the target torque for the **Profile Torque** and **Cyclic Synchronous Torque** modes in tenths of a percent of the rated torque.

Object description

| | |
|------------------|--|
| Index | 6071 _h |
| Object name | Target Torque |
| Object Code | VARIABLE |
| Data type | INTEGER16 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 0000 _h |
| Firmware version | FIR-v1426 |
| Change history | Firmware version FIR-v1626: "Savable" entry changed from "no" to "yes, category: application". |

Description

This object is calculated as thousandths of the torque, e.g., the value "500" means "50%" of the rated torque; "1100" is equivalent to 110%. The rated torque corresponds to the rated current in object **203B_h:01**.

The target torque may not exceed the peak torque (proportional to the peak current in **2031_h**).

6072h Max Torque

Function

The object describes the maximum torque for the **Profile Torque** and **Cyclic Synchronous Torque** modes in tenths of a percent of the rated torque.

Object description

| | |
|-------------|----------------------------|
| Index | 6072 _h |
| Object name | Max Torque |
| Object Code | VARIABLE |
| Data type | UNSIGNED16 |
| Savable | yes, category: application |

| | |
|------------------|-------------------|
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 0000 _h |
| Firmware version | FIR-v1426 |
| Change history | |

Description

This object is calculated as thousandths of the torque, e.g., the value "500" means "50%" of the rated torque; "1100" is equivalent to 110%. The rated torque corresponds to the rated current in object **203B_h:01**.

The target torque may not exceed the peak torque (proportional to the peak current in **2031_h**).

6074h Torque Demand

Function

Current torque set value requested by the ramp generator in tenths of a percent of the nominal torque for the internal controller.

Object description

| | |
|------------------|-------------------|
| Index | 6074 _h |
| Object name | Torque Demand |
| Object Code | VARIABLE |
| Data type | INTEGER16 |
| Savable | no |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 0000 _h |
| Firmware version | FIR-v1426 |
| Change history | |

Description

This object is calculated as thousandths of the torque, e.g., the value "500" means "50%" of the rated torque; "1100" is equivalent to 110%. The rated torque corresponds to the rated current in object **203B_h:01**.

The target torque may not exceed the peak torque (proportional to the peak current in **2031_h**).

6075h Motor Rated Current

Function

Contains the rated current entered in **203B_h:01_h** in mA.

Object description

| | |
|------------------|-----------------------|
| Index | 6075 _h |
| Object name | Motor Rated Current |
| Object Code | VARIABLE |
| Data type | UNSIGNED32 |
| Savable | yes, category: tuning |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000000 _h |
| Firmware version | FIR-v1738-B501312 |
| Change history | |

6077h Torque Actual Value

Function

This object indicates the current torque value in tenths of a percent of the nominal torque for the internal controller.

Object description

| | |
|------------------|---------------------|
| Index | 6077 _h |
| Object name | Torque Actual Value |
| Object Code | VARIABLE |
| Data type | INTEGER16 |
| Savable | no |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 0000 _h |
| Firmware version | FIR-v1540 |
| Change history | |

Description

This object is calculated as thousandths of the torque, e.g., the value "500" means "50%" of the rated torque; "1100" is equivalent to 110%. The rated torque corresponds to the rated current in object **203B_h:01**.

The target torque may not exceed the peak torque (proportional to the peak current in **2031_h**).

607Ah Target Position

Function

This object specifies the target position in **user-defined units** for the **Profile Position** and **Cyclic Synchronous Position** modes.

Object description

| | |
|------------------|--|
| Index | 607A _h |
| Object name | Target Position |
| Object Code | VARIABLE |
| Data type | INTEGER32 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000FA0 _h |
| Firmware version | FIR-v1426 |
| Change history | Firmware version FIR-v1626: "Savable" entry changed from "no" to "yes, category: application". |

607Bh Position Range Limit

Function

Contains the minimum and maximum position in **user-defined units**.

Object description

| | |
|------------------|----------------------------|
| Index | 607B _h |
| Object name | Position Range Limit |
| Object Code | ARRAY |
| Data type | INTEGER32 |
| Savable | yes, category: application |
| Firmware version | FIR-v1426 |
| Change history | |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 02 _h |

| | |
|-------------|--------------------------|
| Subindex | 01 _h |
| Name | Min Position Range Limit |
| Data type | INTEGER32 |
| Access | read / write |
| PDO mapping | RX-PDO |

| | |
|----------------|--------------------------|
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 02 _h |
| Name | Max Position Range Limit |
| Data type | INTEGER32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |

Description

If this range is exceeded or not reached, an overflow occurs. To prevent this overflow, limit values for the target position can be set in object **607D_h** ("Software Position Limit").

607Ch Home Offset

Function

Specifies the difference between the zero position of the controller and the reference point of the machine in **user-defined units**.

Object description

| | |
|------------------|----------------------------|
| Index | 607C _h |
| Object name | Home Offset |
| Object Code | VARIABLE |
| Data type | INTEGER32 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |
| Firmware version | FIR-v1426 |
| Change history | |

607Dh Software Position Limit

Function

Defines the limit positions relative to the reference point of the application in **user-defined units**.

Object description

| | |
|-------------|-------------------------|
| Index | 607D _h |
| Object name | Software Position Limit |
| Object Code | ARRAY |

| | |
|------------------|----------------------------|
| Data type | INTEGER32 |
| Savable | yes, category: application |
| Firmware version | FIR-v1426 |
| Change history | |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 02 _h |

| | |
|----------------|-----------------------|
| Subindex | 01 _h |
| Name | Min Position Limit |
| Data type | INTEGER32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |

| | |
|----------------|-----------------------|
| Subindex | 02 _h |
| Name | Max Position Limit |
| Data type | INTEGER32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |

Description

The target position and the demand position must lie within the limits set here. The Home Offset (**607C_h**) is not taken into account.

607Eh Polarity

Function

With this object, the direction of rotation can be reversed.

Object description

| | |
|-------------|-------------------|
| Index | 607E _h |
| Object name | Polarity |

| | |
|------------------|--|
| Object Code | VARIABLE |
| Data type | UNSIGNED8 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00 _h |
| Firmware version | FIR-v1426 |
| Change history | Firmware version FIR-v1738-B501312: "PDO mapping" table entry for subindex 00 changed from "no" to "RX-PDO". |

Description

The following generally applies for direction reversal: If a bit is set to the value "1", reversal is activated. If the value is "0", the direction of rotation is as described in the respective mode.

| | | | | | | | |
|-----|-----|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| POS | VEL | | | | | | |

VEL (Velocity)

Direction of rotation reversal in the following modes:

- **Profile Velocity Mode**
- **Cyclic Synchronous Velocity Mode**
- **Velocity Mode**

POS (Position)

Direction of rotation reversal in the following modes:

- **Profile Position Mode**
- **Cyclic Synchronous Position Mode**



Tip

You can force an inversion of the rotary field that affects all operating modes. See object **3212_h:02_h**.

607Fh Max Profile Velocity

Function

Specifies the maximum speed in **user-defined units** for **Profile Position**, **Interpolated Position Mode** and **Profile Velocity** modes .

Object description

| | |
|-------------|----------------------|
| Index | 607F _h |
| Object name | Max Profile Velocity |
| Object Code | VARIABLE |
| Data type | UNSIGNED32 |

| | |
|------------------|--|
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00030D40 _h |
| Firmware version | FIR-v1540 |
| Change history | <p>Firmware version FIR-v1738-B501312: "Object name" entry changed from "Max profile velocity" to "Max Profile Velocity".</p> <p>Firmware version FIR-v1738-B501312: "Data type" entry changed from "INTEGER16" to "UNSIGNED32".</p> <p>Firmware version FIR-v1738-B501312: "Savable" entry changed from "no" to "yes, category: application".</p> <p>Firmware version FIR-v1738-B501312: "Access" table entry for subindex 00 changed from "read only" to "read/write".</p> <p>Firmware version FIR-v1738-B501312: "PDO mapping" table entry for subindex 00 changed from "TX-PDO" to "RX-PDO".</p> |

6080h Max Motor Speed

Function

Specifies the maximum permissible speed of the motor in **user-defined units**.

Object description

| | |
|------------------|--|
| Index | 6080 _h |
| Object name | Max Motor Speed |
| Object Code | VARIABLE |
| Data type | UNSIGNED32 |
| Savable | yes, category: tuning |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | <ul style="list-style-type: none"> • PD4-E591L42-E-65-1: 00030D40_h • PD4-E601L42-E-65-1: 00030D40_h • PD4-EB59CD-E-65-1: 00001770_h |
| Firmware version | FIR-v1426 |
| Change history | <p>Firmware version FIR-v1614: "Savable" entry changed from "yes, category: application" to "yes, category: tuning".</p> <p>Firmware version FIR-v1738-B501312: "Object name" entry changed from "Maximum Speed" to "Max Motor Speed".</p> <p>Firmware version FIR-v1738-B501312: "PDO mapping" table entry for subindex 00 changed from "no" to "RX-PDO".</p> |

6081h Profile Velocity

Function

Specifies the maximum travel speed in **user-defined units**.

Object description

| | |
|------------------|----------------------------|
| Index | 6081 _h |
| Object name | Profile Velocity |
| Object Code | VARIABLE |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 000001F4 _h |
| Firmware version | FIR-v1426 |
| Change history | |

6082h End Velocity

Function

Specifies the speed at the end of the traveled ramp in **user-defined units**.

Object description

| | |
|------------------|----------------------------|
| Index | 6082 _h |
| Object name | End Velocity |
| Object Code | VARIABLE |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |
| Firmware version | FIR-v1426 |
| Change history | |

6083h Profile Acceleration

Function

Specifies the maximum acceleration in **user-defined units**.

Object description

| | |
|-------|-------------------|
| Index | 6083 _h |
|-------|-------------------|

| | |
|------------------|----------------------------|
| Object name | Profile Acceleration |
| Object Code | VARIABLE |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 000001F4 _h |
| Firmware version | FIR-v1426 |
| Change history | |

6084h Profile Deceleration

Function

Specifies the maximum deceleration (deceleration ramp) in **user-defined units**.

Object description

| | |
|------------------|----------------------------|
| Index | 6084 _h |
| Object name | Profile Deceleration |
| Object Code | VARIABLE |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 000001F4 _h |
| Firmware version | FIR-v1426 |
| Change history | |

6085h Quick Stop Deceleration

Function

Specifies the maximum Quick Stop Deceleration in **user-defined units**.

Object description

| | |
|----------------|----------------------------|
| Index | 6085 _h |
| Object name | Quick Stop Deceleration |
| Object Code | VARIABLE |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00001388 _h |

| | |
|------------------|-----------|
| Firmware version | FIR-v1426 |
| Change history | |

6086h Motion Profile Type

Function

Specifies the ramp type for the **Profile Position** and **Profile Velocity** modes.

Object description

| | |
|------------------|----------------------------|
| Index | 6086 _h |
| Object name | Motion Profile Type |
| Object Code | VARIABLE |
| Data type | INTEGER16 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 0000 _h |
| Firmware version | FIR-v1426 |
| Change history | |

Description

Value = "0": = Trapezoidal ramp

Value = "3": Ramp with limited jerk

6087h Torque Slope

Function

This object contains the slope of the torque in Torque mode.

Object description

| | |
|------------------|----------------------------|
| Index | 6087 _h |
| Object name | Torque Slope |
| Object Code | VARIABLE |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |
| Firmware version | FIR-v1426 |
| Change history | |

Description

This object is calculated as thousandths of the torque, e.g., the value "500" means "50%" of the rated torque; "1100" is equivalent to 110%. The rated torque corresponds to the rated current in object **203B_h:01**.

The target torque may not exceed the peak torque (proportional to the peak current in **2031_h**).

608Fh Position Encoder Resolution

Function

Contains the physical resolution of the encoder/sensor that is used for position control.

Object description

| | |
|------------------|---|
| Index | 608F _h |
| Object name | Position Encoder Resolution |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | yes, category: tuning |
| Firmware version | FIR-v1426 |
| Change history | Firmware version FIR-v1738-B501312: "Savable" entry changed from "yes, category: application" to "yes, category: tuning". Firmware version FIR-v1738-B501312: "PDO mapping" table entry for subindex 01 changed from "no" to "RX-PDO". Firmware version FIR-v1738-B501312: "PDO mapping" table entry for subindex 02 changed from "no" to "RX-PDO". |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 02 _h |
| Subindex | 01 _h |
| Name | Encoder Increments |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 000007D0 _h |

| | |
|----------------|-----------------------|
| Subindex | 02 _h |
| Name | Motor Revolutions |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000001 _h |

Description

Position Encoder Resolution = Encoder Increments (608F_h:01_h) / Motor Revolutions (608F_h:02_h)

6090h Velocity Encoder Resolution

Function

Contains the physical resolution of the encoder/sensor that is used for speed control.

Object description

| | |
|------------------|-----------------------------|
| Index | 6090 _h |
| Object name | Velocity Encoder Resolution |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | yes, category: tuning |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1738-B501312 |
| Change history | |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 02 _h |

| | |
|-------------|-------------------------------|
| Subindex | 01 _h |
| Name | Encoder Increments Per Second |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |

| | |
|----------------|------------------------------|
| Allowed values | |
| Preset value | 00000000 _h |
| <hr/> | |
| Subindex | 02 _h |
| Name | Motor Revolutions Per Second |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |

Description

Velocity Encoder Resolution = Encoder Increments per second (6090_h:01_h) / Motor Revolutions per second (6090_h:02_h)

6091h Gear Ratio

Function

Number of motor revolutions per output shaft revolution.

Object description

| | |
|------------------|--|
| Index | 6091 _h |
| Object name | Gear Ratio |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Firmware version | FIR-v1426 |
| Change history | Firmware version FIR-v1738-B501312: "PDO mapping" table entry for subindex 01 changed from "no" to "RX-PDO". Firmware version FIR-v1738-B501312: "PDO mapping" table entry for subindex 02 changed from "no" to "RX-PDO". |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 02 _h |
| <hr/> | |
| Subindex | 01 _h |

| | |
|----------------|-----------------------|
| Name | Motor Revolutions |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000001 _h |

| | |
|----------------|-----------------------|
| Subindex | 02 _h |
| Name | Shaft Revolutions |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000001 _h |

Description

Gear Ratio = Motor Revolutions (6091_h:01_h) / Shaft Revolutions (6091_h:02_h)

6092h Feed Constant

Function

Feed in the case of a linear drive; in **user-defined units** per output shaft revolutions.

Object description

| | |
|------------------|----------------------------|
| Index | 6092 _h |
| Object name | Feed Constant |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Firmware version | FIR-v1426 |
| Change history | |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 02 _h |

| | |
|----------|-----------------|
| Subindex | 01 _h |
|----------|-----------------|

| | |
|----------------|-----------------------|
| Name | Feed |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000001 _h |

| | |
|----------------|-----------------------|
| Subindex | 02 _h |
| Name | Shaft Revolutions |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000001 _h |

Description

Feed Constant = Feed (6092_h:01_h) / Shaft Revolutions (6092_h:02_h)

6096h Velocity Factor

Function

This object contains the factor that is used for converting from user-defined speed units. See chapter **User-defined units**.

Object description

| | |
|------------------|----------------------------|
| Index | 6096 _h |
| Object name | Velocity Factor |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1738-B501312 |
| Change history | |

Value description

| | |
|-------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |

| | |
|----------------|-----------------------|
| Allowed values | |
| Preset value | 02 _h |
| <hr/> | |
| Subindex | 01 _h |
| Name | Numerator |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000001 _h |
| <hr/> | |
| Subindex | 02 _h |
| Name | Divisor |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000001 _h |

Description

The subindices have the following functions:

- 01_h: Numerator of the factor
- 02_h: Denominator of the factor

6097h Acceleration Factor

Function

This object contains the factor that is used for converting from user-defined acceleration units. See chapter **User-defined units**.

Object description

| | |
|------------------|----------------------------|
| Index | 6097 _h |
| Object name | Acceleration Factor |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1738-B501312 |
| Change history | |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 02 _h |
| Subindex | 01 _h |
| Name | Numerator |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000001 _h |
| Subindex | 02 _h |
| Name | Divisor |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000001 _h |

Description

The subindices have the following functions:

- 01_h: Numerator of the factor
- 02_h: Denominator of the factor

6098h Homing Method

Function

This object defines the **Homing method** in **Homing Mode**.

Object description

| | |
|-------------|----------------------------|
| Index | 6098 _h |
| Object name | Homing Method |
| Object Code | VARIABLE |
| Data type | INTEGER8 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | RX-PDO |

| | |
|------------------|-----------------|
| Allowed values | |
| Preset value | 23 _h |
| Firmware version | FIR-v1426 |
| Change history | |

6099h Homing Speed

Function

Specifies the speeds for Homing Mode (6098_h) in **user-defined units**.

Object description

| | |
|------------------|----------------------------|
| Index | 6099 _h |
| Object name | Homing Speed |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Firmware version | FIR-v1426 |
| Change history | |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 02 _h |

| | |
|----------------|--------------------------------|
| Subindex | 01 _h |
| Name | Speed During Search For Switch |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000032 _h |

| | |
|----------------|------------------------------|
| Subindex | 02 _h |
| Name | Speed During Search For Zero |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |

| | |
|--------------|-----------------------|
| Preset value | 0000000A _h |
|--------------|-----------------------|

Description

The speed for the search for the switch is specified in subindex 1.

The (lower) speed for the search for the reference position is specified in subindex 2.



Note

- The speed in subindex 2 is simultaneously the initial speed when starting the acceleration ramp. If this is set too high, the motor loses steps or fails to turn at all. If the setting is too high, the index marking will be overlooked. The speed in subindex 2 should therefore be less than 1000 steps per second.
- The speed in subindex 1 must be greater than the speed in subindex 2.

609Ah Homing Acceleration

Function

Specifies the acceleration ramp for Homing Mode in **user-defined units**.

Object description

| | |
|------------------|----------------------------|
| Index | 609A _h |
| Object name | Homing Acceleration |
| Object Code | VARIABLE |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 000001F4 _h |
| Firmware version | FIR-v1426 |
| Change history | |

Description

The ramp is only used when starting up. When the switch is reached, the motor immediately switches to the lower speed; when the end position is reached, it immediately stops.

60A2h Jerk Factor

Function

This object contains the factor that is used for converting from user-defined jerk units. See chapter **User-defined units**.

Object description

| | |
|-------|-------------------|
| Index | 60A2 _h |
|-------|-------------------|

| | |
|------------------|----------------------------|
| Object name | Jerk Factor |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1738-B501312 |
| Change history | |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 02 _h |

| | |
|----------------|-----------------------|
| Subindex | 01 _h |
| Name | Numerator |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000001 _h |

| | |
|----------------|-----------------------|
| Subindex | 02 _h |
| Name | Divisor |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000001 _h |

Description

The subindices have the following functions:

- 01_h: Numerator of the factor
- 02_h: Denominator of the factor

60A4h Profile Jerk

Function

In the case of a ramp with limited jerk, the size of the jerk can be entered in this object. An entry with the value "0" means that the jerk is not limited.

Object description

| | |
|------------------|--|
| Index | 60A4 _h |
| Object name | Profile Jerk |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Firmware version | FIR-v1426 |
| Change history | Firmware version FIR-v1614: "Name" entry changed from "End Acceleration Jerk" to "Begin Deceleration Jerk". Firmware version FIR-v1614: "Name" entry changed from "Begin Deceleration Jerk" to "End Acceleration Jerk". |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 04 _h |
| Subindex | 01 _h |
| Name | Begin Acceleration Jerk |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 000003E8 _h |
| Subindex | 02 _h |
| Name | Begin Deceleration Jerk |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 000003E8 _h |

| | |
|----------------|-----------------------|
| Subindex | 03 _h |
| Name | End Acceleration Jerk |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 000003E8 _h |

| | |
|----------------|-----------------------|
| Subindex | 04 _h |
| Name | End Deceleration Jerk |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 000003E8 _h |

Description

- Subindex 01_h (*Begin Acceleration Jerk*): Initial jerk during acceleration
- Subindex 02_h (*Begin Deceleration Jerk*): Initial jerk during braking
- Subindex 03_h (*End Acceleration Jerk*): Final jerk during acceleration
- Subindex 04_h (*End Deceleration Jerk*): Final jerk during braking

60A8h SI Unit Position

Function

This object contains the position unit. See chapter **User-defined units**.

Object description

| | |
|------------------|----------------------------|
| Index | 60A8 _h |
| Object name | SI Unit Position |
| Object Code | VARIABLE |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | FF410000 _h |
| Firmware version | FIR-v1738-B501312 |
| Change history | |

Description

Object 60A8_h contains:

- Bits 16 to 23: The position unit (see chapter **Units**)

- Bits 24 to 31: The exponent of a power of ten (see chapter **Units**)

| | | | | | | | | | | | | | | | |
|----------------|----|----|----|----|----|----|----|----------------|----|----|----|----|----|----|----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Factor | | | | | | | | Unit | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| reserved (00h) | | | | | | | | reserved (00h) | | | | | | | |

60A9h SI Unit Velocity

Function

This object contains the speed unit. See chapter **User-defined units**.

Object description

| | |
|------------------|----------------------------|
| Index | 60A9 _h |
| Object name | SI Unit Velocity |
| Object Code | VARIABLE |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00B44700 _h |
| Firmware version | FIR-v1738-B501312 |
| Change history | |

Description

Object 60A9_h contains:

- Bits 8 to 15: The time unit (see chapter **Units**)
- Bits 16 to 23: The position unit (see chapter **Units**)
- Bits 24 to 31: The exponent of a power of ten (see chapter **Units**)

| | | | | | | | | | | | | | | | |
|--------------------|----|----|----|----|----|----|----|----------------------|----|----|----|----|----|----|----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Factor | | | | | | | | Nominator (Position) | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Denominator (Time) | | | | | | | | reserved (00h) | | | | | | | |

60B0h Position Offset

Function

Offset for the position set value in **user-defined units**. Is taken into account in the **Cyclic Synchronous Position** and **Clock-direction mode** modes.

Object description

| | |
|-------|-------------------|
| Index | 60B0 _h |
|-------|-------------------|

| | |
|------------------|----------------------------|
| Object name | Position Offset |
| Object Code | VARIABLE |
| Data type | INTEGER32 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |
| Firmware version | FIR-v1738-B505321 |
| Change history | |

60B1h Velocity Offset

Function

Offset for the speed set value in **user-defined units**. Is taken into account in the **Cyclic Synchronous Position**, **Cyclic Synchronous Velocity** and **Clock-direction mode** modes.

Object description

| | |
|------------------|----------------------------|
| Index | 60B1 _h |
| Object name | Velocity Offset |
| Object Code | VARIABLE |
| Data type | INTEGER32 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |
| Firmware version | FIR-v1738-B505321 |
| Change history | |

60B2h Torque Offset

Function

Offset for the torque set value in tenths of a percent. Is taken into account in the **Cyclic Synchronous Position**, **Cyclic Synchronous Velocity**, **Cyclic Synchronous Torque** and **Clock-direction mode** modes.

Object description

| | |
|-------------|----------------------------|
| Index | 60B2 _h |
| Object name | Torque Offset |
| Object Code | VARIABLE |
| Data type | INTEGER16 |
| Savable | yes, category: application |
| Access | read / write |

| | |
|------------------|-------------------|
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 0000 _h |
| Firmware version | FIR-v1738-B505321 |
| Change history | |

60C1h Interpolation Data Record

Function

This object contains the demand position in **user-defined units** for the interpolation algorithm for the **Interpolated Position** operating mode.

Object description

| | |
|------------------|--|
| Index | 60C1 _h |
| Object name | Interpolation Data Record |
| Object Code | ARRAY |
| Data type | INTEGER32 |
| Savable | yes, category: application |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1512 |
| Change history | Firmware version FIR-v1626: "Savable" entry changed from "no" to "yes, category: application". |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 01 _h |

| | |
|----------------|-----------------------|
| Subindex | 01 _h |
| Name | 1st Set-point |
| Data type | INTEGER32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |

Description

The value is taken over at the next synchronization time.

60C2h Interpolation Time Period

Function

This object contains the interpolation time.

Object description

| | |
|------------------|----------------------------|
| Index | 60C2 _h |
| Object name | Interpolation Time Period |
| Object Code | RECORD |
| Data type | INTERPOLATION_TIME_PERIOD |
| Savable | yes, category: application |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1426 |
| Change history | |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 02 _h |

| | |
|----------------|---------------------------------|
| Subindex | 01 _h |
| Name | Interpolation Time Period Value |
| Data type | UNSIGNED8 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 01 _h |

| | |
|-------------|--------------------------|
| Subindex | 02 _h |
| Name | Interpolation Time Index |
| Data type | INTEGER8 |
| Access | read / write |
| PDO mapping | no |

| | |
|----------------|-----------------|
| Allowed values | |
| Preset value | FD _h |

Description

The subindices have the following functions:

- 01_h: Interpolation time.
- 02_h: Power of ten of the interpolation time: must have the value -3 (corresponds to the time basis in milliseconds).

The following applies here: cycle time = value of 60C2_h:01_h * 10^{value of 60C2:02} seconds.

60C4h Interpolation Data Configuration

Function

This object offers the maximum buffer size, specifies the configured buffer organization of the interpolated data and offers objects for defining the size of the record and for deleting the buffer. It is also used to store the position of other data points.

Object description

| | |
|------------------|---|
| Index | 60C4 _h |
| Object name | Interpolation Data Configuration |
| Object Code | RECORD |
| Data type | INTERPOLATION_DATA_CONFIGURATION |
| Savable | yes, category: application |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1512 |
| Change history | <p>Firmware version FIR-v1540: "Access" table entry for subindex 05 changed from "read/write" to "write only".</p> <p>Firmware version FIR-v1540: "Access" table entry for subindex 06 changed from "read/write" to "write only".</p> <p>Firmware version FIR-v1626: "Savable" entry changed from "no" to "yes, category: application".</p> <p>Firmware version FIR-v1650-B472161: "Access" table entry for subindex 01 changed from "read/write" to "read only".</p> |

Value description

| | |
|-------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |

| | |
|----------------|-----------------------|
| Allowed values | |
| Preset value | 06 _h |
| <hr/> | |
| Subindex | 01 _h |
| Name | MaximumBufferSize |
| Data type | UNSIGNED32 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000001 _h |
| <hr/> | |
| Subindex | 02 _h |
| Name | ActualBufferSize |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00000001 _h |
| <hr/> | |
| Subindex | 03 _h |
| Name | BufferOrganization |
| Data type | UNSIGNED8 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00 _h |
| <hr/> | |
| Subindex | 04 _h |
| Name | BufferPosition |
| Data type | UNSIGNED16 |
| Access | read / write |
| PDO mapping | no |
| Allowed values | |
| Preset value | 0001 _h |
| <hr/> | |
| Subindex | 05 _h |
| Name | SizeOfDataRecord |
| Data type | UNSIGNED8 |
| Access | write only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 04 _h |

| | |
|----------------|-----------------|
| Subindex | 06 _h |
| Name | BufferClear |
| Data type | UNSIGNED8 |
| Access | write only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 00 _h |

Description

The value of subindex 01_h contains the maximum possible number of interpolated records.

The value of subindex 02_h contains the current number of interpolated records.

If subindex 03_h is "00_h", this means a FIFO buffer organization; if it is "01_h", it specifies a ring buffer organization.

The value of subindex 04_h is unitless and specifies the next free buffer entry point.

The value of subindex 05_h is specified in units of "byte". If the value "00_h" is written in subindex 06_h, it deletes the received data in the buffer, deactivates access and deletes all interpolated records. If the value "01_h" is written in subindex 06_h, it activates access to the input buffer.

60C5h Max Acceleration

Function

This object contains the maximum permissible acceleration for the **Profile Position** and **Profile Velocity** modes.

Object description

| | |
|------------------|----------------------------|
| Index | 60C5 _h |
| Object name | Max Acceleration |
| Object Code | VARIABLE |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00001388 _h |
| Firmware version | FIR-v1426 |
| Change history | |

60C6h Max Deceleration

Function

This object contains the maximum permissible deceleration (deceleration ramp) for the **Profile Position** and **Profile Velocity** modes.

Object description

| | |
|------------------|----------------------------|
| Index | 60C6 _h |
| Object name | Max Deceleration |
| Object Code | VARIABLE |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00001388 _h |
| Firmware version | FIR-v1426 |
| Change history | |

60E4h Additional Position Actual Value

Function

Contains the current actual position of all existing feedbacks in **user-defined units**.

Object description

| | |
|------------------|----------------------------------|
| Index | 60E4 _h |
| Object name | Additional Position Actual Value |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | no |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1738-B501312 |
| Change history | |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 02 _h |

| | |
|----------|--|
| Subindex | 01 _h - 02 _h |
| Name | Additional Position Actual Value #1 - #2 |

| | |
|----------------|-----------------------|
| Data type | UNSIGNED32 |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |

Description

The subindices have the following function:

- 00_h: Value="1" to "n", where "n" is the number of existing feedbacks.
- n_h:
Subindex n contains the current actual position of the corresponding feedback.
Subindex 01_h always corresponds to the first (and always existing) *sensorless* feedback.

60E5h Additional Velocity Actual Value

Function

Contains the current actual speed of all existing feedbacks in **user-defined units**.

Object description

| | |
|------------------|----------------------------------|
| Index | 60E5 _h |
| Object name | Additional Velocity Actual Value |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | no |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1738-B501312 |
| Change history | |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 02 _h |

| | |
|-----------|--|
| Subindex | 01 _h - 02 _h |
| Name | Additional Velocity Actual Value #1 - #2 |
| Data type | UNSIGNED32 |

| | |
|----------------|-----------------------|
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |

Description

The subindices have the following function:

- 00_h: Value="1" to "n", where "n" is the number of existing feedbacks.
- n_h:
Subindex n contains the current actual speed of the corresponding feedback.
Subindex 01_h always corresponds to the first (and always existing) *sensorless* feedback.

60E6_h Additional Position Encoder Resolution - Encoder Increments

Function

With this object and with 60EB_h, the resolution of each existing feedback is calculated.

Object description

| | |
|------------------|--|
| Index | 60E6 _h |
| Object name | Additional Position Encoder Resolution - Encoder Increments |
| Object Code | ARRAY |
| Data type | INTEGER32 |
| Savable | yes, category: tuning |
| Access | read only |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1738-B501312 |
| Change history | Firmware version FIR-v1748-B531667: "Data type" entry changed from "UNSIGNED32" to "INTEGER32". Firmware version FIR-v1748-B531667: "Data type" entry changed from "UNSIGNED32" to "INTEGER32". |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 02 _h |

| | |
|----------|-----------------------------------|
| Subindex | 01 _h - 02 _h |
|----------|-----------------------------------|

| | |
|----------------|---|
| Name | Additional Position Encoder Resolution - Encoder Increments Feedback Interface #1 - #2 |
| Data type | INTEGER32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00320000 _h |

Description

The subindices have the following function:

- 00_h: Value="1" to "n", where "n" is the number of existing feedbacks.
- n_h:
Subindex n contains the number of increments of the corresponding feedback.
Subindex 01_h always corresponds to the first (and always existing) *sensorless* feedback.

The resolution of feedback "n" is calculated as follows:

Position Encoder Resolution = Encoder Increments (60E6_h:01_h) / Motor Revolutions (60EB_h:02_h)

60E8h Additional Gear Ratio - Motor Shaft Revolutions

Function

In this object and in 60ED_h, you can set the gear ratio of each existing feedback.

Object description

| | |
|------------------|---|
| Index | 60E8 _h |
| Object name | Additional Gear Ratio - Motor Shaft Revolutions |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Access | read only |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1738-B501312 |
| Change history | |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 02 _h |

| | |
|----------------|--|
| Subindex | 01 _h - 02 _h |
| Name | Additional Gear Ratio - Motor Shaft Revolutions Feedback Interface #1 - #2 |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000001 _h |

Description

The subindices have the following function:

- 00_h: Value = "n", where "n" is the number of existing feedbacks.
- n_h: Subindex "n" contains the number of motor revolutions for the corresponding feedback. Subindex 01_h always corresponds to the first (and always existing) *sensorless* feedback.

The gear ratio of feedback "n" is calculated as follows:

$$\text{Gear Ratio} = \text{Motor Shaft Revolutions (60E8}_{n:n_h}) / \text{Driving Shaft Revolutions (60ED}_{n:n_h})$$

60E9h Additional Feed Constant - Feed

Function

In this object and in 60EE_n, you can set a feed constant for each existing feedback.

Object description

| | |
|------------------|---------------------------------|
| Index | 60E9 _h |
| Object name | Additional Feed Constant - Feed |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Access | read only |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1738-B501312 |
| Change history | |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 02 _h |

| | |
|----------------|--|
| Subindex | 01 _h - 02 _h |
| Name | Additional Feed Constant - Feed Feedback Interface #1 - #2 |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000001 _h |

Description

The subindices have the following function:

- 00_h: Value = "n", where "n" is the number of existing feedbacks.
- n_h: Subindex "n" contains the feed in **user-defined units** for the corresponding feedback. Subindex 01_h always corresponds to the first (and always existing) *sensorless* feedback.

The feed constant of feedback "n" is calculated as follows:

$$\text{Feed Constant} = \text{Feed (60E9}_{h:n_h}) / \text{Driving Shaft Revolutions (60EE}_{h:n_h})$$

60EBh Additional Position Encoder Resolution - Motor Revolutions

Function

With this object and with **60E6_h**, the resolution of each existing feedback is calculated.

Object description

| | |
|------------------|--|
| Index | 60EB _h |
| Object name | Additional Position Encoder Resolution - Motor Revolutions |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | yes, category: tuning |
| Access | read only |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1738-B501312 |
| Change history | |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 02 _h |

| | |
|----------------|---|
| Subindex | 01 _h - 02 _h |
| Name | Additional Position Encoder Resolution - Motor Revolutions Feedback Interface #1 - #2 |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000001 _h |

Description

The subindices have the following function:

- 00_h: Value="1" to "n", where "n" is the number of existing feedbacks.
- n_h:
Subindex n contains the number of motor revolutions of the corresponding feedback.
Subindex 01_h always corresponds to the first (and always existing) *sensorless* feedback.

The resolution of feedback "n" is calculated as follows:

Position Encoder Resolution = Encoder Increments (60E6_h:01_h) / Motor Revolutions (60EB_h:02_h)

60EDh Additional Gear Ratio - Driving Shaft Revolutions

Function

In this object and in 60E8_h, you can set the gear ratio of each existing feedback.

Object description

| | |
|------------------|---|
| Index | 60ED _h |
| Object name | Additional Gear Ratio - Driving Shaft Revolutions |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Access | read only |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1738-B501312 |
| Change history | |

Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | RX-PDO |
| Allowed values | |

| | |
|----------------|--|
| Preset value | 02 _h |
| Subindex | 01 _h - 02 _h |
| Name | Additional Gear Ratio - Driving Shaft Revolutions Feedback Interface #1 - #2 |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000001 _h |

Description

The subindices have the following function:

- 00_h: Value = "n", where "n" is the number of existing feedbacks.
- n_h: Subindex "n" contains the number of revolutions of the output shaft for the corresponding feedback.
Subindex 01_h always corresponds to the first (and always existing) *sensorless* feedback.

The gear ratio of feedback "n" is calculated as follows:

$$\text{Gear Ratio} = \text{Motor Shaft Revolutions (60E8}_{n:n_h}) / \text{Driving Shaft Revolutions (60ED}_{n:n_h})$$

60EE_h Additional Feed Constant - Driving Shaft Revolutions

Function

In this object and in 60E9_n, you can set a feed constant for each existing feedback.

Object description

| | |
|------------------|--|
| Index | 60EE _h |
| Object name | Additional Feed Constant - Driving Shaft Revolutions |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Access | read only |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | |
| Firmware version | FIR-v1738-B501312 |
| Change history | |

Value description

| | |
|-----------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |

| | |
|----------------|---|
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 02 _h |
| <hr/> | |
| Subindex | 01 _h - 02 _h |
| Name | Additional Feed Constant - Driving Shaft Revolutions Feedback Interface #1 - #2 |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000001 _h |

Description

The subindices have the following function:

- 00_h: Value = "n", where "n" is the number of existing feedbacks.
- n_h: Subindex "n" contains the number of revolutions of the output shaft for the corresponding feedback.
Subindex 01_h always corresponds to the first (and always existing) *sensorless* feedback.

The feed constant of feedback "n" is calculated as follows:

$$\text{Feed Constant} = \text{Feed (60E9}_{h:n_h}) / \text{Driving Shaft Revolutions (60EE}_{h:n_h})$$

60F2h Positioning Option Code

Function

The object describes the positioning behavior in **Profile Position** mode.

Object description

| | |
|------------------|--|
| Index | 60F2 _h |
| Object name | Positioning Option Code |
| Object Code | VARIABLE |
| Data type | UNSIGNED16 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 0001 _h |
| Firmware version | FIR-v1446 |
| Change history | Firmware version FIR-v1614: "Savable" entry changed from "no" to "yes, category: application". |

Description

Only the following bits are supported at the present time:

| | | | | | | | | | | | | | | | |
|----|--------------|----|----|---------------|----|---|---|----------|---|---------|---|---------|---|---------------|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MS | RESERVED [3] | | | IP OPTION [4] | | | | RADO [2] | | RRO [2] | | CIO [2] | | REL. OPT. [2] | |

REL. OPT. (Relative Option)

These bits determine the behavior with relative rotating movement in "Profile Position" mode if bit 6 of controlword **6040_h** = "1" is set.

| Bit 1 | Bit 0 | Definition |
|-------|-------|---|
| 0 | 0 | Position movements are executed relative to the previous (internal absolute) target position (each relative to 0 if there is no previous target position) |
| 0 | 1 | Position movements are executed relative to the preset value (or output) of the ramp generator. |
| 1 | 0 | Position movements are performed relative to the current position (object 6064_h). |
| 1 | 1 | Reserved |

RRO (Request-Response Option)

These bits determine the behavior when passing controlword **6040_h**, bit 5 ("new setpoint") – in this case, the controller releases the bit itself. This eliminates the need to externally reset the bit to "0" afterwards. After the bit is set to the value "0" by the controller, bit 12 ("setpoint acknowledgment") is also set to the value "0" in statusword **6041_h**.



Note

These options cause the controller to modify object controlword **6040_h**.

| Bit 3 | Bit 2 | Definition |
|-------|-------|---|
| 0 | 0 | The functionality is as described under Setting travel commands . |
| 0 | 1 | The controller releases the "new setpoint" bit as soon as the current targeted movement has reached its target. |
| 1 | 0 | The controller releases the "new setpoint" bit as soon as this is possible for the controller. |
| 1 | 1 | Reserved |

RADO (Rotary Axis Direction Option)

These bits determine the direction of rotation in "Profile Position" mode.

| Bit 7 | Bit 6 | Definition |
|-------|-------|---|
| 0 | 0 | Normal positioning similar to a linear axis: If one of the "Position Range Limits" – 607B_h:01_h and 02_h – is reached or exceeded, the preset is automatically transferred to the other end of the limit. Only with this bit combination is a movement greater than the modulo value possible. |
| 0 | 1 | Positioning only in negative direction: If the target position is greater than the current position, the axis moves to the target position via the "Min Position Range Limit" from object 607D_h:01_h . |

| Bit 7 | Bit 6 | Definition |
|-------|-------|---|
| 1 | 0 | Positioning only in positive direction: If the target position is less than the current position, the axis moves to the target position via the "Max Position Range Limit" from object 607D_h:01_h . |
| 1 | 1 | Positioning with the shortest distance to the target position. If the difference between the current position and the target position in a 360° system is less than 180°, the axis moves in the positive direction. |

60F4h Following Error Actual Value

Function

This object contains the current following error in **user-defined units**.

Object description

| | |
|------------------|------------------------------|
| Index | 60F4 _h |
| Object name | Following Error Actual Value |
| Object Code | VARIABLE |
| Data type | INTEGER32 |
| Savable | no |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |
| Firmware version | FIR-v1426 |
| Change history | |

60F8h Max Slippage

Function

Defines the maximum allowed slippage error in **user-defined units** symmetrically to the **set speed** in **Profile Velocity** mode.

Object description

| | |
|----------------|----------------------------|
| Index | 60F8 _h |
| Object name | Max Slippage |
| Object Code | VARIABLE |
| Data type | INTEGER32 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000190 _h |

Firmware version FIR-v1738-B501312
Change history

Description

If the actual speed deviates so much from the set speed that the value (absolute value) of this object is exceeded, bit 13 in object **6041_h** is set. The deviation must last longer than the time in object **203F_h**.

If the value of **60F8_h** is set to "7FFFFFFF"_h, slippage error monitoring is switched off.

A reaction to the slippage error can be set in object **3700_h**. If a reaction is defined, an error is also entered in object **1003_h**.

60FAh Control Effort

Function

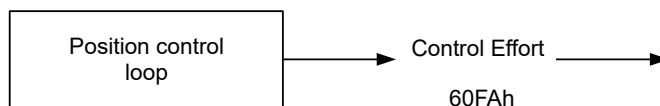
This object contains the correction speed in **user-defined units** that is fed to the speed controller by the position controller.

Object description

| | |
|------------------|-----------------------|
| Index | 60FA _h |
| Object name | Control Effort |
| Object Code | VARIABLE |
| Data type | INTEGER32 |
| Savable | no |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |
| Firmware version | FIR-v1748-B531667 |
| Change history | |

Description

The position controller calculates a correction speed from the difference between the current position and the demand position which is then passed on to the speed controller. This correction value is dependent on the proportional component (**3210_h:01_h**) and the integral component (**3210_h:02_h**) of the position controller. See also chapter **Closed Loop**.



60FCh Position Demand Internal Value

Function

Indicates the current demand position.

Object description

| | |
|------------------|--------------------------------|
| Index | 60FC _h |
| Object name | Position Demand Internal Value |
| Object Code | VARIABLE |
| Data type | INTEGER32 |
| Savable | no |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |
| Firmware version | FIR-v1738-B501312 |
| Change history | |

60FDh Digital Inputs

Function

With this object, the **digital inputs** of the motor can be read.

Object description

| | |
|------------------|-----------------------|
| Index | 60FD _h |
| Object name | Digital Inputs |
| Object Code | VARIABLE |
| Data type | UNSIGNED32 |
| Savable | no |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |
| Firmware version | FIR-v1426 |
| Change history | |

Description

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|------|------|------|------|------|------|------|------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | | | | | | | | IN 8 | IN 7 | IN 6 | IN 5 | IN 4 | IN 3 | IN 2 | IN 1 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | | | | | | | | | | | HS | PLS | NLS |

NLS (Negative Limit Switch)

Negative limit switch

PLS (Positive Limit Switch)

Positive limit switch

HS (Home Switch)

Home switch

IN n (Input n)

Input n – the number of used bits is dependent on the given controller.

60FEh Digital Outputs

Function

With this object, the **digital outputs** of the motor can be written.

Object description

| | |
|------------------|--|
| Index | 60FE _h |
| Object name | Digital Outputs |
| Object Code | ARRAY |
| Data type | UNSIGNED32 |
| Savable | yes, category: application |
| Firmware version | FIR-v1426 |
| Change history | Firmware version FIR-v1626: "Savable" entry changed from "no" to "yes, category: application". |

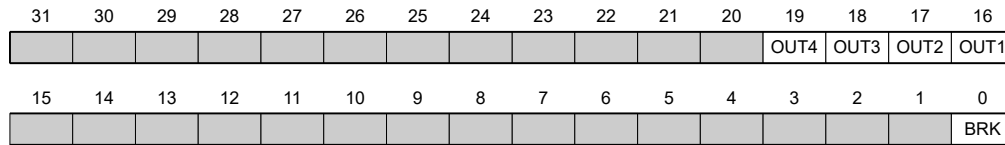
Value description

| | |
|----------------|-----------------------------|
| Subindex | 00 _h |
| Name | Highest Sub-index Supported |
| Data type | UNSIGNED8 |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 01 _h |

| | |
|----------------|-----------------------|
| Subindex | 01 _h |
| Name | Digital Outputs #1 |
| Data type | UNSIGNED32 |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |

Description

To write the outputs, the entries in object **3250_h**, subindex 02_h to 05_h, must also be taken into account.



BRK (Brake)

Bit for the brake output (if the controller supports this function).

OUT n (Output No n)

Bit for the respective digital output; the exact number of digital outputs is dependent on the controller.

60FFh Target Velocity

Function

In this object, the target speed for the **Profile Velocity** and **Cyclic Synchronous Velocity** modes is entered in **user-defined units**.

Object description

| | |
|------------------|--|
| Index | 60FF _h |
| Object name | Target Velocity |
| Object Code | VARIABLE |
| Data type | INTEGER32 |
| Savable | yes, category: application |
| Access | read / write |
| PDO mapping | RX-PDO |
| Allowed values | |
| Preset value | 00000000 _h |
| Firmware version | FIR-v1426 |
| Change history | Firmware version FIR-v1626: "Savable" entry changed from "no" to "yes, category: application". |

6502h Supported Drive Modes

Function

The object describes the supported operating modes in object **6060_h**.

Object description

| | |
|-------------|-----------------------|
| Index | 6502 _h |
| Object name | Supported Drive Modes |
| Object Code | VARIABLE |
| Data type | UNSIGNED32 |

| | |
|------------------|-----------------------|
| Savable | no |
| Access | read only |
| PDO mapping | TX-PDO |
| Allowed values | |
| Preset value | 000003EF _h |
| Firmware version | FIR-v1426 |
| Change history | |

Description

The set bit specifies whether the respective mode is supported. If the value of the bit is "0", the mode is not supported.

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|-----|-----|-----|----|----|----|----|----|----|----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | | | | CST | CSV | CSP | IP | HM | | TQ | PV | VL | PP |

PP

Profile Position Mode

VL

Velocity Mode

PV

Profile Velocity Mode

TQ

Torque Mode

HM

Homing Mode

IP

Interpolated Position Mode

CSP

Cyclic Synchronous Position Mode

CSV

Cyclic Synchronous Velocity Mode

CST

Cyclic Synchronous Torque Mode

6503h Drive Catalogue Number

Function

Contains the device name as character string.

Object description

| | |
|------------------|------------------------|
| Index | 6503 _h |
| Object name | Drive Catalogue Number |
| Object Code | VARIABLE |
| Data type | VISIBLE_STRING |
| Savable | no |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | 0 |
| Firmware version | FIR-v1426 |
| Change history | |

6505h Http Drive Catalogue Address

Function

This object contains the manufacturer's web address as a character string.

Object description

| | |
|------------------|---|
| Index | 6505 _h |
| Object name | Http Drive Catalogue Address |
| Object Code | VARIABLE |
| Data type | VISIBLE_STRING |
| Savable | no |
| Access | read only |
| PDO mapping | no |
| Allowed values | |
| Preset value | http://www.nanotec.de |
| Firmware version | FIR-v1426 |
| Change history | |

11 Copyrights

11.1 Introduction

Integrated in the Nanotec software are components from products from external software manufacturers. In this chapter, you will find the copyright information regarding the used external software sources.

11.2 AES

FIPS-197 compliant AES implementation

Based on XySSL: Copyright (C) 2006-2008 Christophe Devine

Copyright (C) 2009 Paul Bakker <polarssl_maintainer at polarssl dot org>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution; or, the application vendor's website must provide a copy of this notice.
- Neither the names of PolarSSL or XySSL nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The AES block cipher was designed by Vincent Rijmen and Joan Daemen.

<http://csrc.nist.gov/encryption/aes/rijndael/Rijndael.pdf>

<http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>

11.3 MD5

MD5C.C - RSA Data Security, Inc., MD5 message-digest algorithm

Copyright (C) 1991-2, RSA Data Security, Inc. Created 1991. All rights reserved.

License to copy and use this software is granted provided that it is identified as the "RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing this software or this function.

License is also granted to make and use derivative works provided that such works are identified as "derived from the RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing the derived work.

RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided "as is" without express or implied warranty of any kind.

These notices must be retained in any copies of any part of this documentation and/or software.

11.4 uIP

Copyright (c) 2005, Swedish Institute of Computer Science

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the Institute nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE INSTITUTE AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE INSTITUTE OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

11.5 DHCP

Copyright (c) 2005, Swedish Institute of Computer Science

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the Institute nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE INSTITUTE AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE INSTITUTE OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

11.6 CMSIS DSP Software Library

Copyright (C) 2010 ARM Limited. All rights reserved.

11.7 FatFs

FatFs - FAT file system module include file R0.08 (C)ChaN, 2010

FatFs module is a generic FAT file system module for small embedded systems.

This is a free software that opened for education, research and commercial developments under license policy of following terms.

Copyright (C) 2010, ChaN, all right reserved.

The FatFs module is a free software and there is NO WARRANTY.

No restriction on use. You can use, modify and redistribute it for personal, non-profit or commercial product UNDER YOUR RESPONSIBILITY.

Redistributions of source code must retain the above copyright notice.

11.8 Protothreads

Protothread class and macros for lightweight, stackless threads in C++.

This was "ported" to C++ from Adam Dunkels' protothreads C library at: <http://www.sics.se/~adam/pt/>

Originally ported for use by Hamilton Jet (www.hamiltonjet.co.nz) by Ben Hoyt, but stripped down for public release. See his blog entry about it for more information: <http://blog.micropledge.com/2008/07/protothreads/>

Original BSD-style license

Copyright (c) 2004-2005, Swedish Institute of Computer Science.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the Institute nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided by the Institute and contributors "as is" and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the Institute or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

11.9 lwIP

Copyright (c) 2001-2004 Swedish Institute of Computer Science.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This file is part of the lwIP TCP/IP stack.

Author: Adam Dunkels <adam@sics.se>