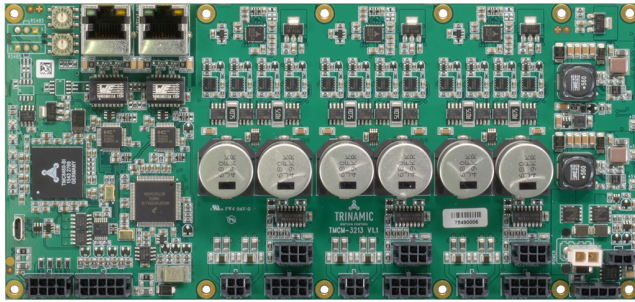


TMCM-3215 CoE Firmware Manual

Firmware Version V1.09 | Document Revision V1.08 • 2020-FEB-06

The TMCM-3215 is a three axes controller/driver module for 2-phase bipolar stepper motors with separate differential encoder and separate home and stop switch inputs for each axis. Dynamic current control, and quiet, smooth and efficient operation are combined with StealthChop™, Dc-Step™, StallGuard™ and CoolStep™ features. The module offers four analog or digital inputs as well as four digital outputs in combination with a break chopper unit.



Features

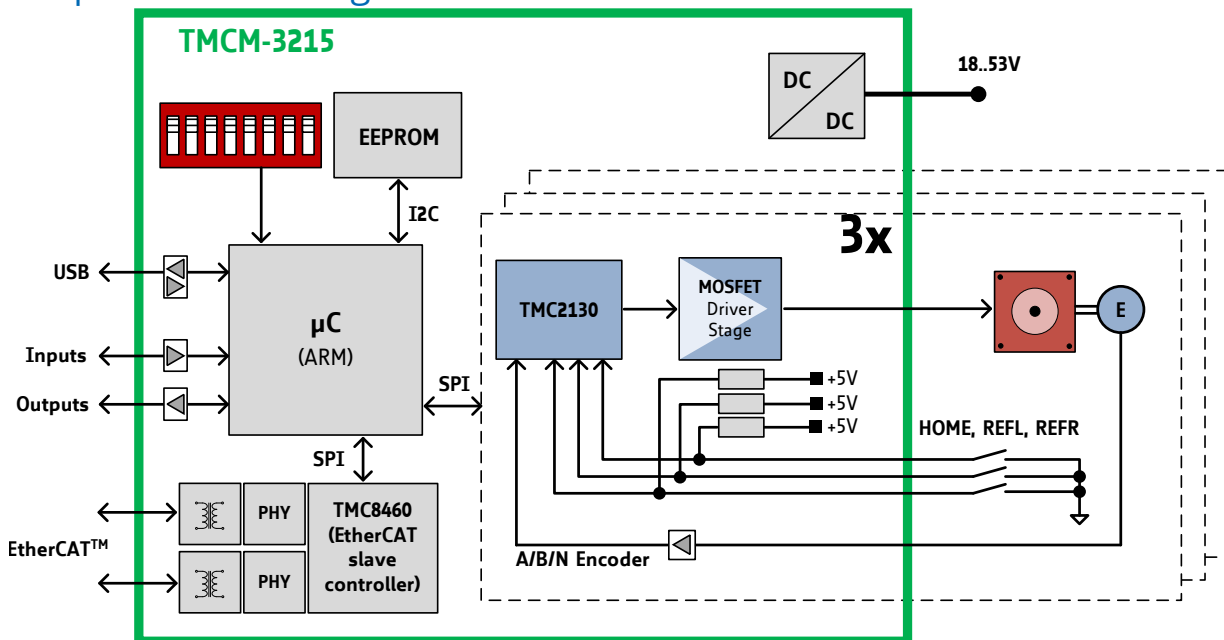
- 3-Axes Stepper Motor Control
- CoE CiA-402 Drive Profile
- Encoder Support
- SixPoint™ ramps
- CoolStep™
- StallGuard2™
- StealthChop™
- DcStep™



Applications

- Laboratory Automation
- Manufacturing
- Semiconductor Handling
- Robotics
- Factory Automation
- Test & Measurement
- Life Science
- Biotechnology
- Liquid Handling

Simplified Block Diagram



©2021 TRINAMIC Motion Control GmbH & Co. KG, Hamburg, Germany
 Terms of delivery and rights to technical change reserved.
 Download newest version at: www.trinamic.com
 EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.



Read entire documentation.

Contents

| | | |
|----------|---|-----------|
| 1 | Preface | 6 |
| 1.1 | General Features of this CoE Implementation | 6 |
| 1.2 | Abbreviations used in this Manual | 7 |
| 1.3 | Firmware Update | 7 |
| 1.4 | Trinamic's unique Features — easy to use with CoE | 8 |
| 1.4.1 | StallGuard2™ | 8 |
| 1.4.2 | CoolStep™ | 8 |
| 1.5 | SixPoint™ Motion Controller | 9 |
| 2 | Communication | 11 |
| 2.1 | Reference Model | 11 |
| 2.2 | NMT State Machine | 13 |
| 2.3 | Device Model | 14 |
| 2.4 | Object Dictionary | 15 |
| 2.4.1 | Object Indices on Multi-Axis Modules | 15 |
| 3 | Communication Area | 17 |
| 3.1 | Detailed Object Specifications | 17 |
| 3.1.1 | Object 1000 _h : Device Type | 17 |
| 3.1.2 | Object 1001 _h : Error Register | 17 |
| 3.1.3 | Object 1008 _h : Manufacturer Device Name | 18 |
| 3.1.4 | Object 1009 _h : Manufacturer Hardware Version | 18 |
| 3.1.5 | Object 100A _h : Manufacturer Software Version | 19 |
| 3.1.6 | Object 1018 _h : Identity Object | 19 |
| 3.1.7 | Object 1600 _h : Receive PDO Mapping Parameter | 20 |
| 3.1.8 | Objects 1A00 _h : Transmit PDO Mapping Parameter | 21 |
| 3.1.9 | Objects 1C00 _h : Sync Manager Communication Type | 21 |
| 3.1.10 | Objects 1C12 _h : Sync Manager 2 PDO Assignment | 22 |
| 3.1.11 | Objects 1C13 _h : Sync Manager 3 PDO Assignment | 23 |
| 4 | Manufacturer Specific Area | 24 |
| 4.1 | Objects related to CoolStep™ | 24 |
| 4.2 | Detailed Object Specifications | 27 |
| 4.2.1 | Object 2000 _h : Microstep Resolution | 27 |
| 4.2.2 | Object 2001 _h : Fullstep Resolution | 27 |
| 4.2.3 | Object 2002 _h : Brake Delay Times | 27 |
| 4.2.4 | Object 2003 _h : Maximum Current | 28 |
| 4.2.5 | Object 2004 _h : Standby Current | 29 |
| 4.2.6 | Object 2005 _h : Limit Switches | 29 |
| 4.2.7 | Object 200A _h : Enable Drive Delay Time | 30 |
| 4.2.8 | Object 200B _h : Encoder Parameters | 30 |
| 4.2.9 | Object 200C _h : Brake Current Feed | 31 |
| 4.2.10 | Object 200F _h : Encoder N Channel Latch | 31 |
| 4.2.11 | Object 2010 _h : Profile Start Velocity | 32 |
| 4.2.12 | Object 2011 _h : Profile A1 | 32 |
| 4.2.13 | Object 2012 _h : Profile V1 | 33 |
| 4.2.14 | Object 2013 _h : Profile D1 | 33 |
| 4.2.15 | Object 2015 _h : Ramp Wait Time | 33 |
| 4.2.16 | Object 2089 _h : Setting Delay | 34 |
| 4.2.17 | Object 208C _h : Velocity Dimension Index | 34 |
| 4.2.18 | Object 208E _h : Acceleration Dimension Index | 35 |
| 4.2.19 | Object 2092 _h : Chopper Blank Time | 35 |



| | | |
|--------|---|----|
| 4.2.20 | Object 2093 _h : Chopper Mode | 36 |
| 4.2.21 | Object 2094 _h : Chopper Hysteresis Decrement | 36 |
| 4.2.22 | Object 2095 _h : Chopper Hysteresis End | 37 |
| 4.2.23 | Object 2096 _h : Chopper Hysteresis Start | 37 |
| 4.2.24 | Object 2097 _h : Chopper Off Time | 37 |
| 4.2.25 | Object 2098 _h : Smart Energy Current Minimum | 38 |
| 4.2.26 | Object 2099 _h : Smart Energy Current Down Step | 38 |
| 4.2.27 | Object 209A _h : Smart Energy Hysteresis | 39 |
| 4.2.28 | Object 209B _h : Smart Energy Current Up Step | 39 |
| 4.2.29 | Object 209C _h : Smart Energy Hysteresis Start | 40 |
| 4.2.30 | Object 209D _h : Smart Energy Filter Enable | 40 |
| 4.2.31 | Object 209E _h : StallGuard2 Threshold | 41 |
| 4.2.32 | Object 20A1 _h : Short Protection Disable | 41 |
| 4.2.33 | Object 20A3 _h : Vsense | 42 |
| 4.2.34 | Object 20A4 _h : Stop on Stall | 42 |
| 4.2.35 | Object 20A5 _h : Smart Energy Threshold Speed | 43 |
| 4.2.36 | Object 20B0 _h : PWM Threshold Speed | 43 |
| 4.2.37 | Object 20B1 _h : PWM Gradient | 43 |
| 4.2.38 | Object 20B2 _h : PWM Amplitude | 44 |
| 4.2.39 | Object 20B3 _h : DcStep Minimum Speed | 44 |
| 4.2.40 | Object 20B4 _h : DcStep Time | 45 |
| 4.2.41 | Object 20B5 _h : DcStep StallGuard | 45 |
| 4.2.42 | Object 20B6 _h : Fullstep Threshold Speed | 46 |
| 4.2.43 | Object 20B7 _h : High Speed Chopper Mode | 46 |
| 4.2.44 | Object 20B8 _h : High Speed Fullstep Mode | 46 |
| 4.2.45 | Object 20B9 _h : Power Down Ramp | 47 |
| 4.2.46 | Object 2100 _h : Home Offset Display | 47 |
| 4.2.47 | Object 2101 _h : Actual Load Value | 48 |
| 4.2.48 | Object 2102 _h : Driver Error Flags | 48 |
| 4.2.49 | Object 2107 _h : Microstep Resolution Display | 49 |
| 4.2.50 | Object 210B _h : Step Counter | 49 |
| 4.2.51 | Object 2121 _h : PWM Scale Value | 50 |
| 4.2.52 | Object 2122 _h : Measured Velocity | 50 |
| 4.2.53 | Object 2702 _h : Device Digital Inputs | 50 |
| 4.2.54 | Object 2703 _h : Device Digital Outputs | 51 |
| 4.2.55 | Object 270E _h : Device Analog Inputs | 52 |

5 Profile Specific Area 53

| | | |
|--------|--|----|
| 5.1 | Detailed Object Specifications | 53 |
| 5.1.1 | Object 605A _h : Quick Stop Option Code | 53 |
| 5.1.2 | Object 605B _h : Shutdown Option Code | 54 |
| 5.1.3 | Object 605C _h : Disable Operation Option Code | 55 |
| 5.1.4 | Object 605D _h : Halt Option Code | 55 |
| 5.1.5 | Object 605E _h : Fault Reaction Option Code | 56 |
| 5.1.6 | Object 6060 _h : Modes of Operation | 56 |
| 5.1.7 | Object 6061 _h : Modes of Operation Display | 57 |
| 5.1.8 | Object 606A _h : Sensor Selection Code | 58 |
| 5.1.9 | Object 608F _h : Position Encoder Resolution | 59 |
| 5.1.10 | Object 60FD _h : Digital Inputs | 59 |
| 5.1.11 | Object 6502 _h : Supported Drive Modes | 59 |

6 Profile Position Mode 61

| | | |
|-------|---|----|
| 6.1 | Detailed Object Specifications | 61 |
| 6.1.1 | Object 6040 _h : Control Word | 62 |



| | | |
|----------|--|-----------|
| 6.1.2 | Object 6041 _h : Status Word | 63 |
| 6.1.3 | Object 6062 _h : Position Demand Value | 64 |
| 6.1.4 | Object 6063 _h : Position Actual Internal Value | 65 |
| 6.1.5 | Object 6064 _h : Position Actual Value | 65 |
| 6.1.6 | Object 6065 _h : Following Error Window | 66 |
| 6.1.7 | Object 6067 _h : Position Window | 66 |
| 6.1.8 | Object 6068 _h : Position Window Time | 67 |
| 6.1.9 | Object 606C _h : Velocity Actual Value | 67 |
| 6.1.10 | Object 607A _h : Target Position | 68 |
| 6.1.11 | Object 607D _h : Software Position Limit | 68 |
| 6.1.12 | Object 6081 _h : Profile Velocity | 69 |
| 6.1.13 | Object 6082 _h : End Velocity | 69 |
| 6.1.14 | Object 6083 _h : Profile Acceleration | 70 |
| 6.1.15 | Object 6084 _h : Profile Deceleration | 70 |
| 6.1.16 | Object 6085 _h : Quick Stop Deceleration | 70 |
| 6.1.17 | Object 60F2 _h : Positioning Option Code | 71 |
| 6.2 | How to move a Motor in pp Mode | 72 |
| 7 | Profile Velocity Mode | 73 |
| 7.1 | Detailed Object Specifications | 73 |
| 7.1.1 | Object 6040 _h : Control Word | 73 |
| 7.1.2 | Object 6041 _h : Status Word | 74 |
| 7.1.3 | Object 6062 _h : Position Demand Value | 76 |
| 7.1.4 | Object 6063 _h : Position Actual Internal Value | 76 |
| 7.1.5 | Object 6064 _h : Position Actual Value | 77 |
| 7.1.6 | Object 6065 _h : Following Error Window | 77 |
| 7.1.7 | Object 606C _h : Velocity Actual Value | 78 |
| 7.1.8 | Object 607D _h : Software Position Limit | 78 |
| 7.1.9 | Object 6083 _h : Profile Acceleration | 79 |
| 7.1.10 | Object 6085 _h : Quick Stop Deceleration | 79 |
| 7.1.11 | Object 60FF _h : Target Velocity | 79 |
| 7.2 | How to move a Motor in pv Mode | 80 |
| 8 | Homing Mode | 81 |
| 8.1 | Homing Methods | 82 |
| 8.1.1 | Homing Method 1: Homing on negative Limit Switch and Index Pulse | 82 |
| 8.1.2 | Homing Method 2: Homing on positive Limit Switch and Index Pulse | 83 |
| 8.1.3 | Homing Method 3: Homing on positive Home Switch and Index Pulse | 83 |
| 8.1.4 | Homing Method 5: Homing on negative Home Switch and Index Pulse | 83 |
| 8.1.5 | Homing Method 17: Homing on negative Limit Switch | 84 |
| 8.1.6 | Homing Method 18: Homing on positive Limit Switch | 84 |
| 8.1.7 | Homing Method 19: Homing on positive Home Switch | 85 |
| 8.1.8 | Homing Method 21: Homing on negative Home Switch | 85 |
| 8.1.9 | Homing Method 33 and 34: Homing on next Index Pulse | 85 |
| 8.1.10 | Homing Method 35: Current Position as Home Position | 86 |
| 8.2 | Detailed Object Specifications | 87 |
| 8.2.1 | Object 6040 _h : Control Word | 87 |
| 8.2.2 | Object 6041 _h : Status Word | 88 |
| 8.2.3 | Object 606C _h : Velocity Actual Value | 89 |
| 8.2.4 | Object 607C _h : Home Offset | 90 |
| 8.2.5 | Object 6098 _h : Homing Method | 91 |
| 8.2.6 | Object 6099 _h : Homing Speeds | 91 |
| 8.2.7 | Object 609A _h : Homing Acceleration | 91 |
| 8.3 | How to start a Homing in hm Mode | 92 |



| | | |
|-----------|---|------------|
| 9 | Cyclic synchronous Position Mode | 93 |
| 9.1 | Detailed Object Specifications | 93 |
| 9.1.1 | Object 6040 _h : Control Word | 93 |
| 9.1.2 | Object 6041 _h : Status Word | 94 |
| 9.1.3 | Object 6062 _h : Position Demand Value | 96 |
| 9.1.4 | Object 6063 _h : Position Actual Internal Value | 96 |
| 9.1.5 | Object 6064 _h : Position Actual Value | 97 |
| 9.1.6 | Object 606C _h : Velocity Actual Value | 97 |
| 9.1.7 | Object 607A _h : Target Position | 97 |
| 9.1.8 | Object 607D _h : Software Position Limit | 98 |
| 9.1.9 | Object 60B0 _h : Position Offset | 98 |
| 9.1.10 | Object 60C2 _h : Interpolation Time Period | 99 |
| 10 | Cyclic synchronous Velocity Mode | 100 |
| 10.1 | Detailed Object Specifications | 100 |
| 10.1.1 | Object 6040 _h : Control Word | 100 |
| 10.1.2 | Object 6041 _h : Status Word | 101 |
| 10.1.3 | Object 606C _h : Velocity Actual Value | 103 |
| 10.1.4 | Object 60FF _h : Target Velocity | 103 |
| 10.1.5 | Object 607D _h : Software Position Limit | 103 |
| 10.1.6 | Object 60B1 _h : Velocity Offset | 104 |
| 10.1.7 | Object 60C2 _h : Interpolation Time Period | 104 |
| 11 | Emergency Messages (EMCY) | 106 |
| 12 | Figures Index | 108 |
| 13 | Tables Index | 109 |
| 14 | Supplemental Directives | 113 |
| 14.1 | Producer Information | 113 |
| 14.2 | Copyright | 113 |
| 14.3 | Trademark Designations and Symbols | 113 |
| 14.4 | Target User | 113 |
| 14.5 | Disclaimer: Life Support Systems | 113 |
| 14.6 | Disclaimer: Intended Use | 113 |
| 14.7 | Collateral Documents & Tools | 114 |
| 15 | Revision History | 115 |
| 15.1 | Firmware Revision | 115 |
| 15.2 | Document Revision | 115 |



1 Preface

This document specifies objects and modes of operation of the Trinamic TMCM-3215 stepper motor control module with CANopen-over-EtherCAT (CoE) firmware. The CoE firmware is designed to fulfill the EtherCAT® version of the CANopen DS402 standards. The EtherCAT® conformance has also been tested. This manual assumes that the reader is already familiar with the basics of EtherCAT® and the CoE protocol (especially DS402).

It is also possible to connect with the TMCL-IDE through the USB interface in parallel, but only for monitoring purposes (for example monitoring velocity and position for debugging purposes) and for firmware updates. It is not possible to move the motors using the TMCL-IDE. This can only be done through EtherCAT®.

1.1 General Features of this CoE Implementation

Main Characteristics

- Communication according to EtherCAT® standards
- Protocols: CoE, FoE

SDO Communication

- 1 server
- Expedited transfer
- Segmented transfer
- No block transfer

PDO Communication

- Producer
- Consumer
- RPDOs
 - Dynamic mapping with max. 9 mapping entries.
 - Default mappings: manufacturer specific.
- TPDOs
 - Dynamic mapping with max. 9 mapping entries.
 - Default mappings: manufacturer specific.

Sync managers

- Sync manager 0: receive mailbox used for SDO communication
- Sync manager 1: send mailbox used for SDO communication
- Sync manager 2: process data output (used for TPDO)
- Sync manager 3: process data input (used for RPDO)



Further Characteristics

- Emergency: producer

1.2 Abbreviations used in this Manual

| Abbreviations | |
|---------------|-------------------------------|
| CAN | Controller area network |
| CoE | CANopen over EtherCAT |
| CHGND | chassis ground / earth ground |
| COB | Communication object |
| FoE | File transfer over EtherCAT |
| FSA | Finite state automaton |
| FSM | Finite state machine |
| NMT | Network management |
| ID | Identifier |
| LSB | Least significant bit |
| MSB | Most significant bit |
| PDO | Process data object |
| PDS | Power drive system |
| RPDO | Receive process data object |
| SDO | Service data object |
| TPDO | Transmit process data object |
| EMCY | Emergency object |
| rw | Read and write |
| ro | Read only |
| hm | Homing mode |
| pp | Profile position mode |
| pv | Profile velocity mode |
| vm | Velocity mode |

Table 1: Abbreviations used in this Manual

1.3 Firmware Update

The software running on the microprocessor consists of two parts, a boot loader and the CoE firmware itself. Whereas the boot loader is installed during production and testing at TRINAMIC and remains untouched throughout the whole lifetime, the CoE firmware can easily be updated by the user. The new firmware can be loaded into the module either via file transfer over EtherCAT (FoE) or via the firmware update function of the TMCL-IDE, using the USB interface of the module.



1.4 Trinamic’s unique Features — easy to use with CoE

1.4.1 StallGuard2™

StallGuard2™ is a high-precision sensorless load measurement using the back EMF of the coils. It can be used for stall detection as well as other uses at loads below those which stall the motor. The StallGuard2™ measurement value changes linearly over a wide range of load, velocity, and current settings. At maximum motor load, the value reaches zero or is near zero. This is the most energy-efficient point of operation for the motor.

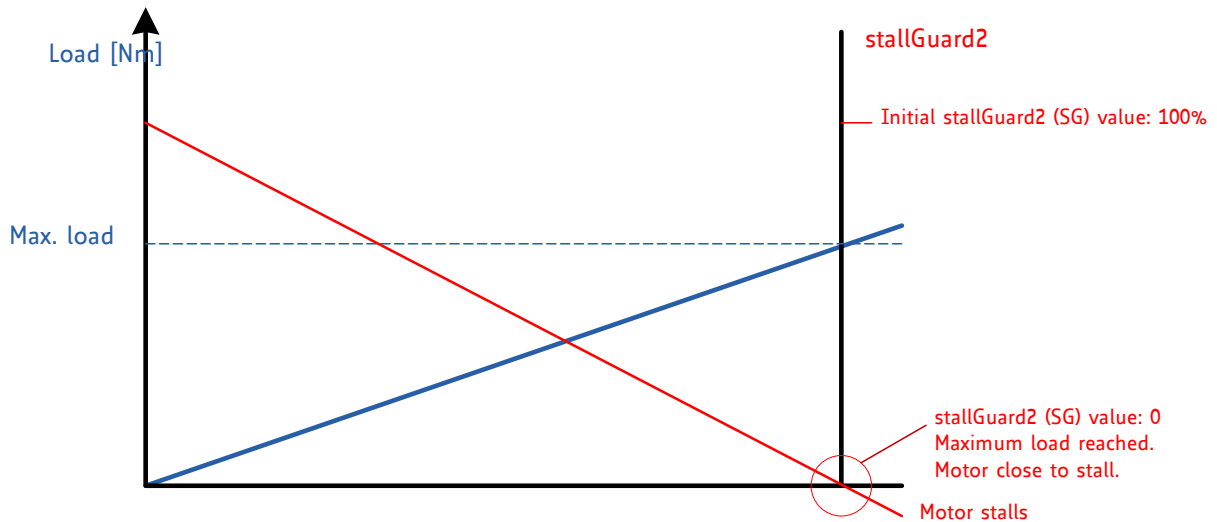


Figure 1: stallGuard2 Load Measurement as a Function of Load

1.4.2 CoolStep™

CoolStep™ is a load-adaptive automatic current scaling based on the load measurement via StallGuard2™ adapting the required current to the load. Energy consumption can be reduced by as much as 75%. CoolStep™ allows substantial energy savings, especially for motors which see varying loads or operate at a high duty cycle. Because a stepper motor application needs to work with a torque reserve of 30% to 50%, even a constant-load application allows significant energy savings because CoolStep™ automatically enables torque reserve when required. Reducing power consumption keeps the system cooler, increases motor life, and allows cost reduction.



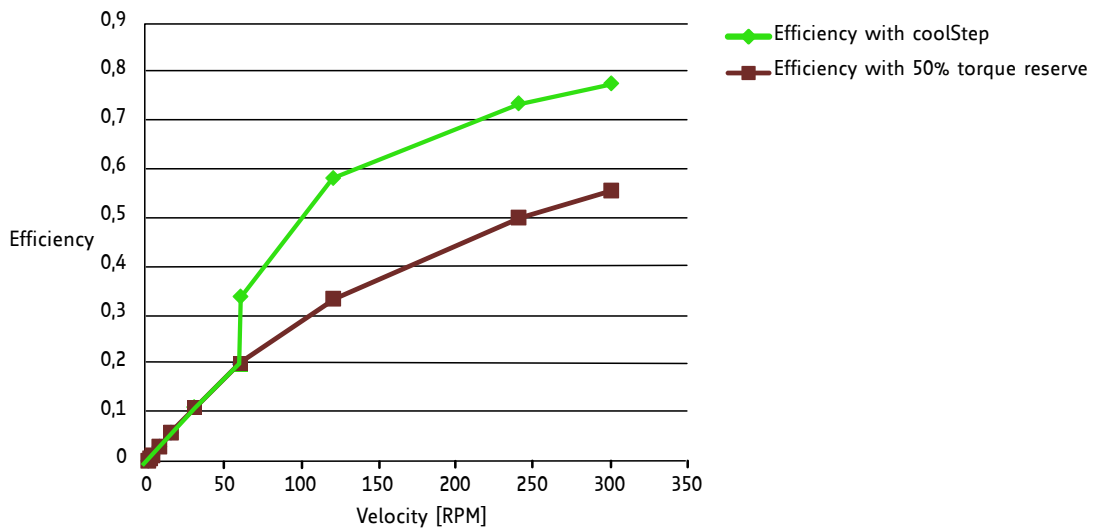


Figure 2: Energy Efficiency Example with CoolStep

1.5 SixPoint™ Motion Controller

TRINAMIC’s SixPoint™ motion controller is a new type of ramp generator which offers faster machine operation compared to the classical linear acceleration ramps. The SixPoint™ ramp generator allows adapting the acceleration ramps to the torque curves of a stepper motor. It uses two different acceleration settings for the acceleration phase and also two different deceleration settings for the deceleration phase. Start and stop speeds greater than zero can also be used.

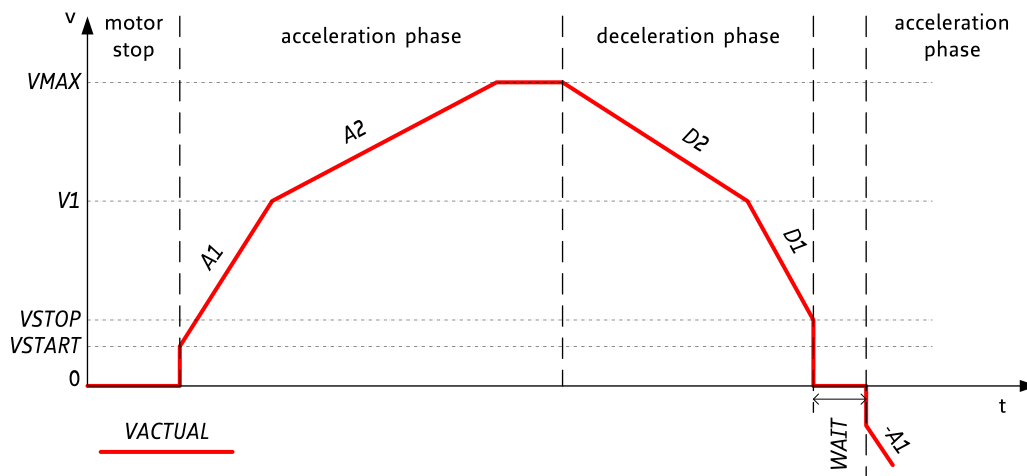


Figure 3: Typical motion profile with TRINAMIC’s SixPoint™ motion controller

A six point ramp begins using the start speed V_{START} (which can also be zero). Then, the acceleration value $A1$ will be used to accelerate the motor to the speed $V1$. When the speed $V1$ has been reached, the motor will be further accelerated using the acceleration value $A2$ until it has reached the speed V_{MAX} . The deceleration phase begins using the deceleration value $D2$. After reaching the speed $V1$ again the deceleration value $D1$ will be used to decelerate to the stop speed V_{STOP} (which can also be zero).



The SixPoint™ ramp can be configured using the following objects:

| Parameter Name | Object Index |
|--|-------------------|
| Start velocity (V_{START}) | 2010 _h |
| Acceleration A1 | 2011 _h |
| Velocity V1 | 2012 _h |
| Acceleration A2 | 6083 _h |
| Maximum positioning velocity (V_{MAX}) | 6081 _h |
| Deceleration D2 | 6084 _h |
| Deceleration D1 | 2013 _h |
| Stop velocity V_{STOP} | 6082 _h |
| Wait time WAIT | 2015 _h |

Table 2: SixPoint Ramp Parameters

Setting the velocity V1 (object 2012_h) to zero switches off the SixPoint™ ramp. In this case, a trapezoidal ramp defined by the parameters V_{START} , A2, V_{MAX} , D2 and V_{STOP} will be used.

Note

The SixPoint™ ramp will only be used in profile positioning mode (pp mode). Profile velocity mode (pv mode) will always use a trapezoidal ramp, defined just by the acceleration (object 6083_h), the speed given using object 60FF_h and the start and stop speed (objects 2010_h and 6082_h). The deceleration parameters will not be used in pv mode.



2 Communication

2.1 Reference Model

The application layer comprises a concept to configure and communicate real-time-data as well as the mechanisms for synchronization between devices. The functionality which the application layer offers to an application is logically divided over different service data objects (SDO) in the application layer. A service object offers a specific functionality and all the related services.

Applications interact by invoking services of a service object in the application layer. To realize these services this object exchanges data via the EtherCAT with peer service object(s) using a protocol.

The application and the application layer interact with service primitives.

| Service Primitives | |
|--------------------|---|
| Primitive | Definition |
| Request | Issued by the application to the application layer to request a service. |
| Indication | Issued by the application layer to the application to report an internal event detected by the application layer or indicate that a service is requested. |
| Response | Issued by the application to the application layer to respond to a previous received indication. |
| Confirmation | Issued by the application layer to the application to report the result of a previously issued request. |

Table 3: Service Primitives

A service type defines the primitives that are exchanged between the application layer and the cooperating applications for a particular service of a service object. Unconfirmed and confirmed services are collectively called remote services.



| Service Types | |
|----------------------------|--|
| Type | Definition |
| Local service | Involves only the local service object. The application issues a request to its local service object that executes the requested service without communicating with peer service object(s). |
| Unconfirmed service | Involves one or more peer service objects. The application issues a request to its local service object. This request is transferred to the peer service object(s) that each passes it to their application as an indication. The result is not confirmed back. |
| Confirmed service | Can involve only one peer service object. The application issues a request to its local service object. This request is transferred to the peer service object that passes it to the other application as an indication. The other application issues a response that is transferred to the originating service object that passes it as a confirmation to the requesting application. |
| Provider initiated service | Involves only the local service object. The service object (being the service provider) detects an event not solicited by a requested service. This event is then indicated to the application. |

Table 4: Service Types



2.2 NMT State Machine

The finite state machine (FSM) or simply state machine is a model of behavior composed of a finite number of states, transitions between those states, and actions. It shows which way the logic runs when certain conditions are met.

Starting and resetting the device is controlled via the state machine. The NMT state machine consists of the states shown in figure 4.

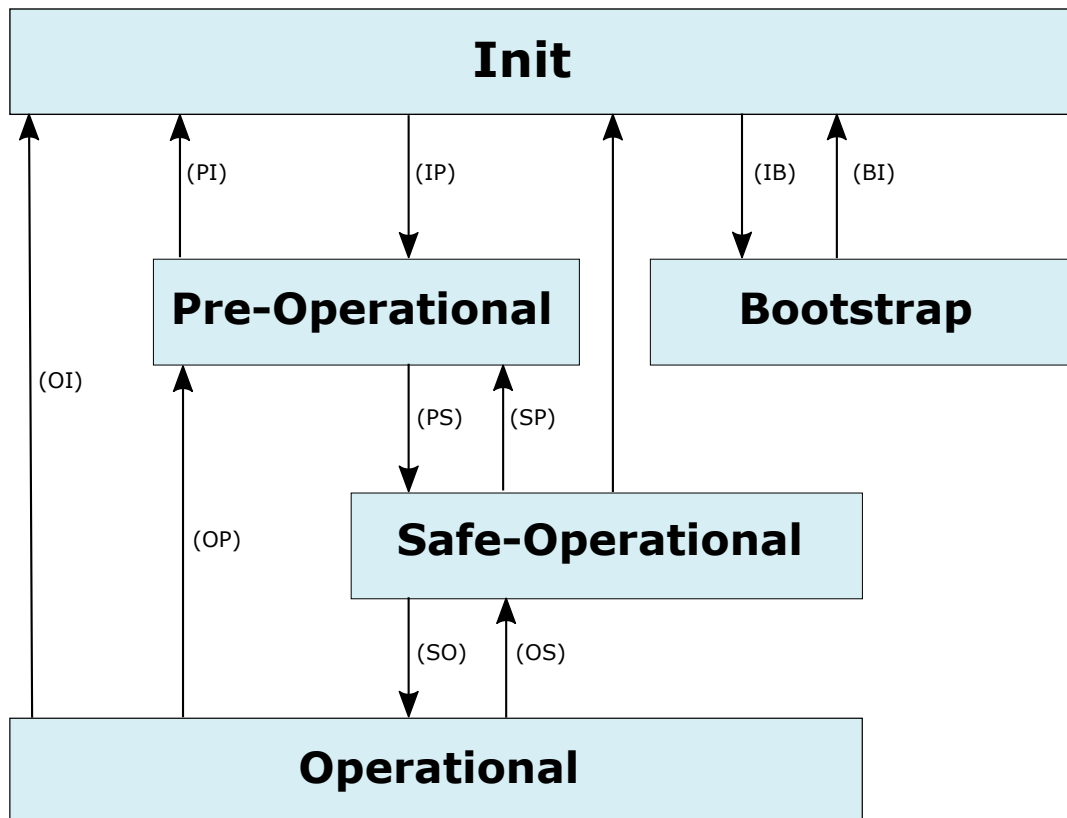


Figure 4: NMT State Machine

After power-on or reset the device enters the Initialization (**INIT**) state.

The master can then switch the device to Pre-Operational (**PRE-OP**) state. In this state, only SDO communication is possible. PDO communication is not possible.

In Safe-Operational (**SAFE-OP**) state, also PDO communication is possible. Inputs can be read, but outputs cannot be switched and the motor cannot be run.

In Operational (**OP**) state, all features of the module can be used. PDO communication is possible, outputs can be switched and the motor can be used. During Operational state the device can use all supported communication objects.

When switching from Operational to Safe-Operational state the motor will be stopped if it has been running. When the EtherCAT connection is lost during Operational state the device will also automatically



switch to Safe-Operational state.

The Bootstrap (**BOOT**) state is used for firmware updates via FoE. Before FoE can be used the device has to be switched to this state.

2.3 Device Model

A CoE device mainly consists of the following parts:

- *Communication*: This function unit provides the communication objects and the appropriate functionality to transport data items via the underlying network structure.
- *Object dictionary*: The object dictionary is a collection of all the data items which have an influence on the behavior of the application objects, the communication objects and the state machine used on this device.
- *Application*: The application comprises the functionality of the device with respect to the interaction with the process environment.

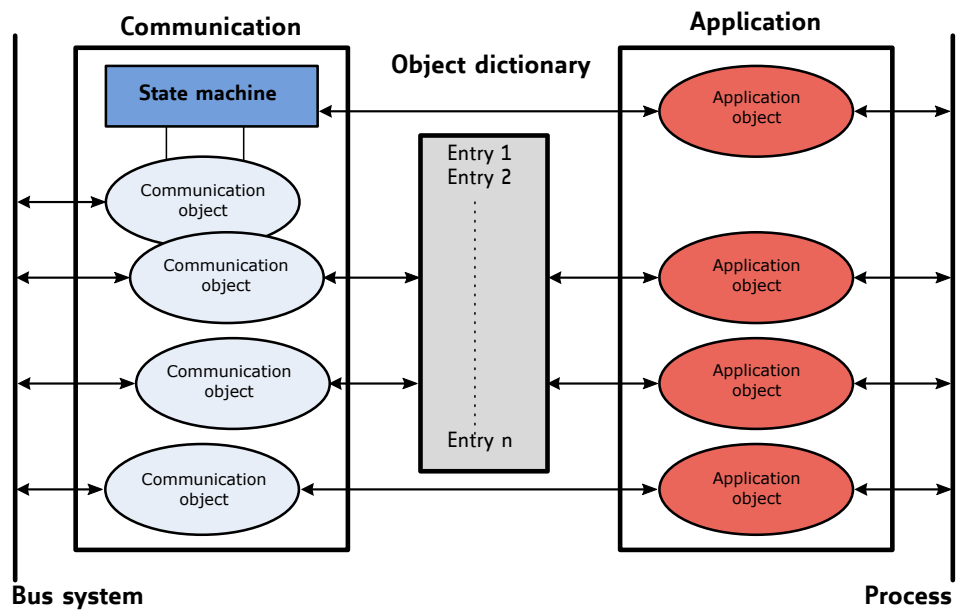


Figure 5: Device Model



2.4 Object Dictionary

The most important part of a device profile is the object dictionary description. The object dictionary is essentially a grouping of objects accessible via the network in an ordered pre-defined fashion. Each object within the dictionary is addressed using a 16-bit index. The overall layout of the standard object dictionary is shown in table 5:

| Object Dictionary | |
|---------------------------------------|---|
| Index | Object |
| 0000 _h | Not used. |
| 0001 _h – 001F _h | Static data types. |
| 0020 _h – 003F _h | Complex data types. |
| 0040 _h – 005F _h | Manufacturer specific complex data types. |
| 0060 _h – 007F _h | Device profile specific static data types. |
| 0080 _h – 009F _h | Device profile specific complex data types. |
| 00A0 _h – 0FFF _h | Reserved for further use. |
| 1000 _h – 1FFF _h | Communication profile area. |
| 2000 _h – 5FFF _h | Manufacturer specific profile area. |
| 6000 _h – 9FFF _h | Standardized device profile area. |
| A000 _h – BFFF _h | Standardized interface profile area. |
| C000 _h – FFFF _h | Reserved for further use. |

Table 5: Object Dictionary

The communication profile area at indices 1000_h through 1FFF_h contains the communication specific parameters for the CAN network. These entries are common to all devices.

The manufacturer segment at indices 2000_h through 5FFF_h contains manufacturer specific objects. These objects control the special features of the Trinamic TMCM-3215 motion control device.

The standardized device profile area at indices 6000_h through 9FFF_h contains all data objects common to a class of devices that can be read or written via the network. They describe the device parameters and the device functionality of the device profile.

2.4.1 Object Indices on Multi-Axis Modules

On a multi-axis module like the TMCM-3215 each object in the manufacturer area and each object in the profile specific area is available for each motor. In this manual, only the object indices for motor #0 are shown. The objects for the other motors can be accessed by adding offsets to the object indices:

- Add an offset of $motor_number \cdot 200_h$ to the index of a manufacturer specific object to get its index for other motors.
- Add an offset of $motor_number \cdot 800_h$ to the index of a profile specific object to get its index for other motors.



For example, the control word for motor #1 would be 6840_h (instead of 6040_h for motor #0), and the microstep resolution of motor #1 would be 2200_h for motor #1 (instead of 2000_h for motor #0).

| Multi-Axis Object Indices | | |
|---------------------------|-------------------|-------------------|
| Motor | Manufacturer area | Profile area |
| Motor #0 | $2000_h - 21FF_h$ | $6000_h - 67FF_h$ |
| Motor #1 | $2200_h - 23FF_h$ | $6800_h - 6FFF_h$ |
| Motor #2 | $2400_h - 25FF_h$ | $7000_h - 77FF_h$ |

Table 6: Multi-Axis Object Indices



3 Communication Area

The communication area contains all objects that define the communication parameters of the CoE device according to the EtherCAT standard.

3.1 Detailed Object Specifications

3.1.1 Object 1000_h: Device Type

This object contains information about the device type. The object 1000_h describes the type of device and its functionality. It is composed of a 16-bit field which describes the device profile that is used and a second 16-bit field which provides additional information about optional functionality of the device.

| Object Description | | | |
|--------------------|-------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 1000 _h | Device type | Variable | UNSIGNED32 |

Table 7: Object Description (1000_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|-----------------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | ro | no | UNSIGNED32 | FFFC0192 _h |

Table 8: Entry Description (1000_h)

3.1.2 Object 1001_h: Error Register

This object contains error information. The CANopen device maps internal errors into object 1001_h. It is part of an emergency object.

| Object Description | | | |
|--------------------|----------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 1001 _h | Error register | Variable | UNSIGNED8 |

Table 9: Object Description (1001_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | ro | no | UNSIGNED8 | 0 |

Table 10: Entry Description (1001_h)



| Error Register Bits | |
|---------------------|-------------------------|
| Bit | Definition |
| 0 | Generic error |
| 1 | Current |
| 2 | Voltage |
| 3 | Temperature |
| 4 | Communication error |
| 5 | Device profile specific |
| 6 | Reserved (always 0) |
| 7 | Manufacturer specific |

Table 11: Error Register Bits

3.1.3 Object 1008_h: Manufacturer Device Name

This object contains the name of the device as given by the manufacturer.

| Object Description | | | |
|--------------------|--------------------------|-------------|----------------|
| Index | Name | Object Type | Data Type |
| 1008 _h | Manufacturer Device Name | Variable | Visible String |

Table 12: Object Description (1008_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | ro | no | — | TMCM-3215 |

Table 13: Entry Description (1008_h)

3.1.4 Object 1009_h: Manufacturer Hardware Version

This object contains the hardware version description.

| Object Description | | | |
|--------------------|-------------------------------|-------------|----------------|
| Index | Name | Object Type | Data Type |
| 1009 _h | Manufacturer Hardware Version | Variable | Visible String |

Table 14: Object Description (1009_h)



| Entry Description | | | | |
|-------------------|--------|-------------|-------------|------------------------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | ro | no | — | Depends on device, e.g. 1.0. |

Table 15: Entry Description (1009_h)

3.1.5 Object 100A_h: Manufacturer Software Version

This object contains the software version description.

| Object Description | | | |
|--------------------|-------------------------------|-------------|----------------|
| Index | Name | Object Type | Data Type |
| 100A _h | Manufacturer Software Version | Variable | Visible String |

Table 16: Object Description (100A_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|------------------------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | ro | no | — | Depends on device, e.g. 1.0. |

Table 17: Entry Description (100A_h)

3.1.6 Object 1018_h: Identity Object

The object 1018_h contains general information about the device:

- The vendor ID (sub-index 01_h) contains a unique value allocated to each manufacturer. The vendor ID of Trinamic is 286_h.
- The manufacturer specific product code (sub-index 2_h) identifies a specific device version.
- The manufacturer specific revision number (sub-index 3_h) consists of a major revision number and a minor revision number.

| Object Description | | | |
|--------------------|-----------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 1018 _h | Identity object | Record | Identity |

Table 18: Object Description (1018_h)

| Entry Description | | | | | |
|-------------------|-------------------|--------|-------------|-------------|---|
| Sub-index | Description | Access | PDO Mapping | Value Range | Default Value |
| 00 _h | Number of entries | ro | no | 0...3 | 3 |
| 01 _h | Vendor ID | ro | no | UNSIGNED32 | 0286 _h |
| 02 _h | Product code | ro | no | UNSIGNED32 | 3215 |
| 03 _h | Revision number | ro | no | UNSIGNED32 | e.g. 20003 _h for version 2.3 |

Table 19: Entry Description (1018_h)

3.1.7 Object 1600_h: Receive PDO Mapping Parameter

This object contains the mapping parameters for the RPDO the device is able to receive. The sub-index 00_h contains the number of valid entries within the mapping record. This number of entries is also the number of the application variables which shall be received with the corresponding RPDO. The sub-indices from 01_h to the number of entries contain the information about the mapped application variables. These entries describe the PDO contents by their index, sub-index and length.

| Object Description | | | |
|--------------------|-------------------------------|-------------|-------------|
| Index | Name | Object Type | Data Type |
| 1600 _h | Receive PDO mapping parameter | RECORD | PDO Mapping |

Table 20: Object Description (1600_h)

| Entry Description | | | | |
|-------------------|---|--------|-------------|-----------------------|
| Sub-index | Description | Access | Value Range | Default Value |
| 00 _h | Number of mapped application objects in PDO | rw | 0...9 | 9 |
| 01 _h | Mapping entry 1 | rw | UNSIGNED32 | 60400010 _h |
| 02 _h | Mapping entry 2 | rw | UNSIGNED32 | 607A0020 _h |
| 03 _h | Mapping entry 3 | rw | UNSIGNED32 | 60FF0020 _h |
| 04 _h | Mapping entry 4 | rw | UNSIGNED32 | 68400010 _h |
| 05 _h | Mapping entry 5 | rw | UNSIGNED32 | 687A0020 _h |
| 06 _h | Mapping entry 6 | rw | UNSIGNED32 | 68FF0020 _h |
| 07 _h | Mapping entry 7 | rw | UNSIGNED32 | 70400010 _h |
| 08 _h | Mapping entry 8 | rw | UNSIGNED32 | 707A0020 _h |
| 09 _h | Mapping entry 9 | rw | UNSIGNED32 | 70FF0020 _h |

Table 21: Entry Description (1600_h)



3.1.8 Objects 1A00_h: Transmit PDO Mapping Parameter

This object contains the mapping parameters for the TPDO the device is able to transmit. The sub-index 00_h contains the number of valid entries within the mapping record. This number of entries is also the number of the application variables which shall be transmitted with the corresponding TPDO. The sub-indices from 01_h to the number of entries contain the information about the mapped application variables. These entries describe the PDO contents by their index, sub-index and length.

| Object Description | | | |
|--------------------|--------------------------------|-------------|-------------|
| Index | Name | Object Type | Data Type |
| 1A00 _h | Transmit PDO mapping parameter | RECORD | PDO Mapping |

Table 22: Object Description (1A00_h)

| Entry Description | | | | |
|-------------------|---|--------|-------------|-----------------------|
| Sub-index | Description | Access | Value Range | Default Value |
| 00 _h | Number of mapped application objects in PDO | rw | 0...9 | 9 |
| 01 _h | Mapping entry 1 | rw | UNSIGNED32 | 60410010 _h |
| 02 _h | Mapping entry 2 | rw | UNSIGNED32 | 60640008 _h |
| 03 _h | Mapping entry 3 | rw | UNSIGNED32 | 606C0020 _h |
| 04 _h | Mapping entry 4 | rw | UNSIGNED32 | 68410010 _h |
| 05 _h | Mapping entry 5 | rw | UNSIGNED32 | 68640020 _h |
| 06 _h | Mapping entry 6 | rw | UNSIGNED32 | 686C0020 _h |
| 07 _h | Mapping entry 7 | rw | UNSIGNED32 | 70410010 _h |
| 08 _h | Mapping entry 8 | rw | UNSIGNED32 | 70640020 _h |
| 09 _h | Mapping entry 9 | rw | UNSIGNED32 | 706C0020 _h |

Table 23: Entry Description (1A00_h)

3.1.9 Objects 1C00_h: Sync Manager Communication Type

This object describes the communication types of the EtherCAT sync managers. The types of the first four sync managers are normally fixed and should not be changed. Sync managers can have the following for communication types:



| Sync Manager Communication Types | |
|----------------------------------|---------------------|
| Type | Description |
| 1 | Mailbox receive |
| 2 | Mailbox send |
| 3 | Process data input |
| 4 | Process data output |

Table 24: Sync Manager Communication Types

| Object Description | | | |
|--------------------|---------------------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 1C00 _h | Sync manager communication type | RECORD | UNSIGNED8 |

Table 25: Object Description (1C00_h)

| Entry Description | | | | |
|-------------------|-----------------------------------|--------|-------------|---------------|
| Sub-index | Description | Access | Value Range | Default Value |
| 00 _h | Number of entries | rw | 0...3 | 4 |
| 01 _h | Communication type sync manager 1 | rw | UNSIGNED8 | 1 |
| 02 _h | Communication type sync manager 2 | rw | UNSIGNED8 | 2 |
| 03 _h | Communication type sync manager 3 | rw | UNSIGNED8 | 3 |
| 04 _h | Communication type sync manager 4 | rw | UNSIGNED8 | 4 |

Table 26: Entry Description (1C00_h)

3.1.10 Objects 1C12_h: Sync Manager 2 PDO Assignment

This object contains the index of the PDO definition object that is assigned to sync manager 2. Normally, the RPDO objects are assigned to sync manager 2. Under most circumstances there is no need to change this setting.

| Object Description | | | |
|--------------------|-------------------------------|-------------|----------------|
| Index | Name | Object Type | Data Type |
| 1C12 _h | Sync manager 2 PDO assignment | RECORD | PDO assignment |

Table 27: Object Description (1C12_h)



| Entry Description | | | | |
|-------------------|------------------------------------|--------|-------------|-------------------|
| Sub-index | Description | Access | Value Range | Default Value |
| 00 _h | Number of assigned PDOs | rw | 0...1 | 1 |
| 01 _h | PDO mapping index of assigned RPDO | rw | UNSIGNED16 | 1600 _h |

Table 28: Entry Description (1C12_h)

3.1.11 Objects 1C13_h: Sync Manager 3 PDO Assignment

This object contains the index of the PDO definition object that is assigned to sync manager 3. Normally, the TPDO objects are assigned to sync manager 3. Under most circumstances there is no need to change this setting.

| Object Description | | | |
|--------------------|-------------------------------|-------------|----------------|
| Index | Name | Object Type | Data Type |
| 1C13 _h | Sync manager 3 PDO assignment | RECORD | PDO assignment |

Table 29: Object Description (1C13_h)

| Entry Description | | | | |
|-------------------|------------------------------------|--------|-------------|-------------------|
| Sub-index | Description | Access | Value Range | Default Value |
| 00 _h | Number of assigned PDOs | rw | 0...1 | 1 |
| 01 _h | PDO mapping index of assigned TPDO | rw | UNSIGNED16 | 1A00 _h |

Table 30: Entry Description (1C13_h)



4 Manufacturer Specific Area

The manufacturer segment contains manufacturer specific objects. These objects control the special features of the Trinamic Motion Control device TMCM-3215.

Info

This section of the manual only shows the object indices for motor #0. Of course the same objects are also available for the other motors. For the other motors, add an offset of $motor_number \cdot 200_h$ to the object index. So for example the microstep resolution (object 2000_h for motor #0) can be accessed as object 2200_h for motor #1 and as object 2400_h for motor #2. Please see also section 2.4.1.

| Multi-axis Object Indices | |
|---------------------------|--------------------|
| Motor | Object Index Range |
| Motor #0 | $2000_h - 21FF_h$ |
| Motor #1 | $2200_h - 23FF_h$ |
| Motor #2 | $2400_h - 25FF_h$ |

Table 31: Multi-axis Object Indices (Manufacturer specific Area)

4.1 Objects related to CoolStep™

Figure 6 shows an overview of the CoolStep™ related objects for motor #0. Please bear in mind that the figure only shows one example for a drive. There are objects which concern the configuration of the current. Other objects are for velocity regulation and for time adjustment. The CoolStep™ feature is sometimes also called SmartEnergy.

The following adjustments have to be made:

- Thresholds for current and velocity have to be identified and set.
- The StallGuard2™ feature has to be adjusted and enabled.
- The reduction or increasing of the current in the CoolStep™ area (depending on the load) has to be configured.



coolStep™ adjustment points and thresholds

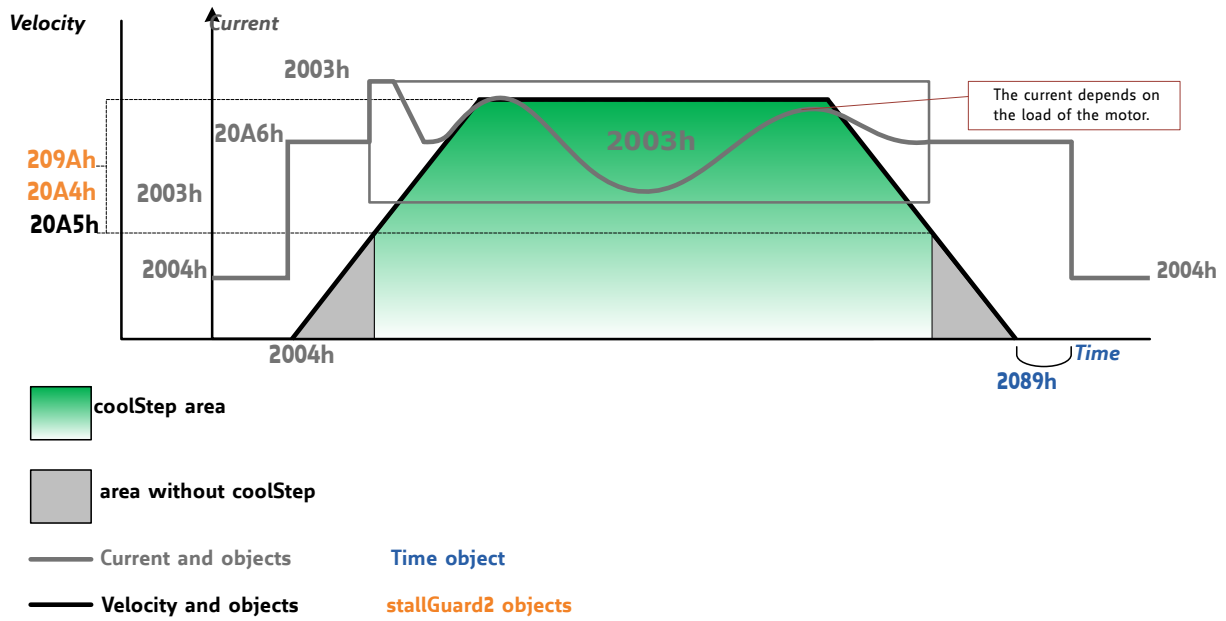


Figure 6: CoolStep Adjustment Points and Thresholds



| CoolStep Adjustment Objects | | |
|-----------------------------|-------------------------------|--|
| Object | Name | Description |
| 2003 _h | Absolute maximum current | The maximum value is 255. This value means 100% of the maximum current of the module. The current adjustment is within the range 0...255 and can be adjusted in 32 steps (0...255 divided by eight; step 0 = 0...7, step 1 = 8...15 and so on). The most important motor setting, since too high values might cause motor damage! |
| 2004 _h | Standby current | The current limit two seconds after the motor has stopped. |
| 2098 _h | SmartEnergy current minimum | Sets the lower motor current limit for CoolStep operation by scaling the run current (object 2003 _h) value. This can be: 0: for 1/2 of the run current 1: for 1/4 of the run current |
| 2099 _h | SmartEnergy current down step | Sets the speed of current decrement when the StallGuard2 reading is above the upper threshold. 0: slow decrement 3: fast decrement |
| 209B _h | SmartEnergy current up step | Sets the current increment step when the StallGuard2 below the lower threshold. 0: slow increment 3: fast increment / fast reaction to rising load |
| 209A _h | SmartEnergy hysteresis | Sets the distance between the lower and the upper threshold for StallGuard2 reading. Above the upper threshold the motor current becomes decreased. |
| 20A4 _h | Stop on stall | Below this speed the motor will not be stopped. Above this speed the motor will stop in case StallGuard2 load value reaches zero. |
| 20A5 _h | SmartEnergy threshold speed. | Above this speed CoolStep becomes enabled. |
| 2089 _h | Standby delay | Standstill period before the current is changed down to standby current. The standard value is 200 which is 2 seconds. |

Table 32: CoolStep related Objects



4.2 Detailed Object Specifications

4.2.1 Object 2000_h: Microstep Resolution

This object sets the microstep resolution of the drive. A value of 8 selects 256 (2⁸) microsteps per full step.

| Object Description | | | |
|--------------------|----------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 2000 _h | Microstep Resolution | Variable | UNSIGNED8 |

Table 33: Object Description (2000_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0...8 | 8 |

Table 34: Entry Description (2000_h)

4.2.2 Object 2001_h: Fullstep Resolution

This object sets the fullstep resolution of the motor connected to the drive. Its default value is 200 because most motors are 1.8° motors.

| Object Description | | | |
|--------------------|----------------------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 2001 _h | Motor full step resolution | Variable | UNSIGNED16 |

Table 35: Object Description (2001_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0...65535 | 200 |

Table 36: Entry Description (2001_h)

4.2.3 Object 2002_h: Brake Delay Times

With this object the delay times for applying and releasing an (optional) brake can be defined. Please see also object 200Ah for an additional delay between enabling the power stage and releasing the brake. Both times are given in ms.



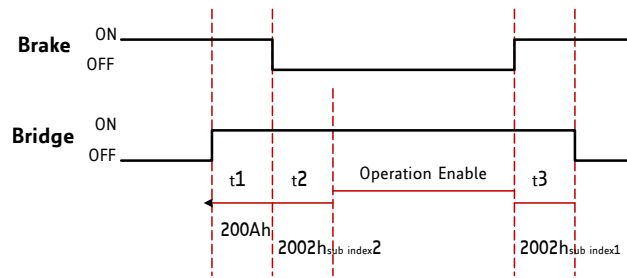


Figure 7: Brake Output Timing

| Object Description | | | |
|--------------------|-------------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 2002 _h | Brake delay times | Array | UNSIGNED16 |

Table 37: Object Description (2002_h)

| Entry Description | | | | | |
|-------------------|---|--------|-------------|-------------|---------------|
| Sub-index | Description | Access | PDO Mapping | Value Range | Default Value |
| 1 | Time between applying brake / disabling power stage | rw | no | 0...65535 | 0 |
| 2 | Time between releasing brake / switching the state machine to operational | rw | no | 0..65535 | 0 |

Table 38: Entry Description (2002_h)

4.2.4 Object 2003_h: Maximum Current

This object defines the current used when the motor is moving. A value of 255 means 100% of the maximum current of the drive.

| Object Description | | | |
|--------------------|-----------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 2003 _h | Maximum current | Variable | UNSIGNED8 |

Table 39: Object Description (2003_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0...255 | 128 |

Table 40: Entry Description (2003_h)



4.2.5 Object 2004_h: Standby Current

This object defines the current used when the motor is standing (two seconds after the last move). A value of 255 means 100% of the maximum current of the drive.

| Object Description | | | |
|--------------------|-----------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 2004 _h | Maximum current | Variable | UNSIGNED8 |

Table 41: Object Description (2004_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0...255 | 8 |

Table 42: Entry Description (2004_h)

4.2.6 Object 2005_h: Limit Switches

This object defines which limit switches are to be used. Bit 0 stands for the left and bit 1 stands for the right limit switch. If a bit is set, the corresponding limit switch will not be used. So this object has to be set to the value 3 if limit switches are not connected. The object can only be written when the drive is in the SWITCHED_ON_DISABLED state (but is always readable).

The limit switches can also be inverted using bit 2 and bit 3:

- Bit 2 inverts the left limit switch
- Bit 3 inverts the right limit switch

The polarity of the home switch can be set using bit 5.

| Object Description | | | |
|--------------------|----------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 2005 _h | Limit switches | Variable | UNSIGNED32 |

Table 43: Object Description (2005_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0...63 | 0 |

Table 44: Entry Description (2005_h)



| Bit Definitions | |
|-----------------|--|
| Bit | Definition |
| 0 | Left limit switch deactivated if set. |
| 1 | Right limit switch deactivated if set. |
| 2 | Left limit switch inverted if set. |
| 3 | Right limit switch inverted if set. |
| 4 | Home switch deactivated if set. |
| 5 | Home switch inverted if set. |

Table 45: Bit Definitions (2005_h)

4.2.7 Object 200A_h: Enable Drive Delay Time

This is an additional delay time (in milliseconds) between enabling the power stage and releasing the brake. It can be used to prevent the brake from being released too early (before the hold current in the motor has been reached). Please see also object 4.2.3.

| Object Description | | | |
|--------------------|-------------------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 200A _h | Enable drive delay time | Variable | UNSIGNED16 |

Table 46: Object Description (200A_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0...65535 | 0 |

Table 47: Entry Description (200A_h)

4.2.8 Object 200B_h: Encoder Parameters

This object defines encoder parameters. These are the polarity of the encoder null channel, the direction of rotation (set it to 1 if the direction is reversed compared to the motor) and if the position is to be initialized with the encoder position. It is only writable in SWITCHED_ON_DISABLED state.

| Object Description | | | |
|--------------------|--------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 200B _h | Encoder parameters | Array | UNSIGNED8 |

Table 48: Object Description (200B_h)



| Entry Description | | | | | |
|-------------------|-----------------------|--------|-------------|-------------|---------------|
| Sub-index | Description | Access | PDO Mapping | Value Range | Default Value |
| 1 | Null channel polarity | rw | no | 0/1 | 0 |
| 2 | Direction of rotation | rw | no | 0/1 | 0 |
| 3 | Initialize position | rw | no | 0/1 | 1 |

Table 49: Entry Description (200B_h)

4.2.9 Object 200C_h: Brake Current Feed

This object configures how much current has to be fed into the brake to apply and to release it. 0 means 0%, 255 means 100% of the maximum current (this depends on the module). In most cases it is needed to feed current into the brake to release it. Setting both values to 0 disables the automatic brake control. This object is only writable in SWITCHED_ON_DISABLED state.

| Object Description | | | |
|--------------------|--------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 200C _h | Brake current feed | Array | UNSIGNED8 |

Table 50: Object Description (200C_h)

| Entry Description | | | | | |
|-------------------|-----------------|--------|-------------|-------------|---------------|
| Sub-index | Description | Access | PDO Mapping | Value Range | Default Value |
| 1 | Apply current | rw | no | 0...255 | 0 |
| 2 | Release current | rw | no | 0...255 | 0 |

Table 51: Entry Description (200C_h)

4.2.10 Object 200F_h: Encoder N Channel Latch

With this object it is possible to make use of the encoder N channel latch capabilities of the motion controller used on the TMCM-3215 module. Write 1 to sub-index 1 to activate the latch functionality. When the encoder N channel has been passed, the internal position and the encoder position will automatically be stored to sub-index 2 and sub-index 3 of this object, and sub-index 1 will be reset to zero to show that the N channel event has occurred.

The encoder interface must have been configured properly in order to make this function work correctly.

| Object Description | | | |
|--------------------|-------------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 200F _h | Encoder N Channel Latch | Variable | RECORD |

Table 52: Object Description (200F_h)



| Entry Description | | | | | |
|-------------------|---------------------------|--------|-------------|-------------|---------------|
| Sub-index | Description | Access | PDO Mapping | Value Range | Default Value |
| 1 | Mode | rw | no | 0/1 | 0 |
| 2 | Latched internal position | ro | no | SIGNED32 | 0 |
| 3 | Latched encoder position | ro | no | SIGNED32 | 0 |

Table 53: Entry Description (200F_h)

4.2.11 Object 2010_h: Profile Start Velocity

This object contains the velocity with which a positioning ramp will be started.

| Object Description | | | |
|--------------------|------------------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 2010 _h | Profile Start Velocity | Variable | UNSIGNED32 |

Table 54: Object Description (2010_h)

| Entry Description | | | | |
|-------------------|--------|-------------|---------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0...268435455 | 0 |

Table 55: Entry Description (2010_h)

4.2.12 Object 2011_h: Profile A1

This object contains the acceleration value used for ramping up from the start velocity (object 2011_h, see section 4.2.11) to the velocity V1 (object (2012)_h, see section 4.2.13).

| Object Description | | | |
|--------------------|------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 2011 _h | Profile A1 | Variable | UNSIGNED32 |

Table 56: Object Description (2011_h)

| Entry Description | | | | |
|-------------------|--------|-------------|--------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0...16777215 | 0 |

Table 57: Entry Description (2011_h)

4.2.13 Object 2012_h: Profile V1

This object contains the velocity used for the first segment of a positioning ramp.

| Object Description | | | |
|--------------------|------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 2012 _h | Profile V1 | Variable | UNSIGNED32 |

Table 58: Object Description (2012_h)

| Entry Description | | | | |
|-------------------|--------|-------------|--------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0...16777215 | 0 |

Table 59: Entry Description (2012_h)

4.2.14 Object 2013_h: Profile D1

This object contains the deceleration value used for decelerating from the maximum positioning velocity to the velocity V1 (object 2012_h, see section 4.2.13).

| Object Description | | | |
|--------------------|------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 2013 _h | Profile D1 | Variable | UNSIGNED32 |

Table 60: Object Description (2013_h)

| Entry Description | | | | |
|-------------------|--------|-------------|--------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0...16777215 | 0 |

Table 61: Entry Description (2013_h)

4.2.15 Object 2015_h: Ramp Wait Time

This object defines the waiting time after ramping down to zero velocity before the next movement or direction inversion can start. Time range is 0 to 2 seconds. This setting avoids excess acceleration e.g. from positive stop velocity to negative start velocity.



| Object Description | | | |
|--------------------|----------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 2015 _h | Ramp Wait Time | Variable | UNSIGNED16 |

Table 62: Object Description (2015_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-----------------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0...65535 [0.000032s] | 0 |

Table 63: Entry Description (2015_h)

4.2.16 Object 2089_h: Setting Delay

This object has to be used for setting a standstill period before the current is changed down to standby current.

Unit: 10msec

| Object Description | | | |
|--------------------|---------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 2089 _h | Setting Delay | Variable | UNSIGNED16 |

Table 64: Object Description (2089_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0...400 | 0 |

Table 65: Entry Description (2089_h)

4.2.17 Object 208C_h: Velocity Dimension Index

With this object different units can be chosen:

- Writing 0 selects internal units.
- Writing 181 sets PPS for velocity and PPS/s for acceleration.

This can only be changed in SWITCHED_ON_DISABLED mode.



| Object Description | | | |
|--------------------|--------------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 208C _h | Velocity Dimension Index | Variable | UNSIGNED8 |

Table 66: Object Description (208C_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0/181 | 181 |

Table 67: Entry Description (208C_h)

4.2.18 Object 208E_h: Acceleration Dimension Index

With this object, the unit for acceleration can be read out. The unit can be set using object 208C_h. Object 208E_h reads 0 when internal units are selected and 179 when PPS/s is selected.

| Object Description | | | |
|--------------------|------------------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 208E _h | Acceleration Dimension Index | Variable | UNSIGNED8 |

Table 68: Object Description (208E_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | ro | no | 0/179 | 179 |

Table 69: Entry Description (208E_h)

4.2.19 Object 2092_h: Chopper Blank Time

This object serves for selecting the comparator blank time. This time needs to safely cover the switching event and the duration of the ringing on the sense resistor. For low current drivers, a setting of 1 or 2 is good. For higher current applications a setting of 2 or 3 will be required.

| Object Description | | | |
|--------------------|--------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 2092 _h | Chopper Blank Time | Variable | UNSIGNED8 |

Table 70: Object Description (2092_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0...3 | 2 |

Table 71: Entry Description (2092_h)

4.2.20 Object 2093_h: Chopper Mode

Select the chopper mode using this object:

- 0 – spreadCycle chopper
- 1 – classic constant off time chopper

| Object Description | | | |
|--------------------|--------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 2093 _h | Chopper Mode | Variable | UNSIGNED8 |

Table 72: Object Description (2093_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0/1 | 0 |

Table 73: Entry Description (2093_h)

4.2.21 Object 2094_h: Chopper Hysteresis Decrement

This object serves for the hysteresis decrement setting. This setting determines the slope of the hysteresis during on time and during fast decay time.

- 0 – fast decrement
- 3 – very slow decrement

| Object Description | | | |
|--------------------|------------------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 2094 _h | Chopper Hysteresis Decrement | Variable | UNSIGNED8 |

Table 74: Object Description (2094_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0...3 | 0 |

Table 75: Entry Description (2094_h)



4.2.22 Object 2095_h: Chopper Hysteresis End

This object provides the setting of the hysteresis end value after a number of decrements. The decrement interval time is controlled by object 2094_h (section 4.2.21).

Possible values are:

- -3...-1 – negative hysteresis end setting
- 0 – zero hysteresis end setting
- 1...12 – positive hysteresis end setting

| Object Description | | | |
|--------------------|------------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 2095 _h | Chopper Hysteresis End | Variable | SIGNED8 |

Table 76: Object Description (2095_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | -3...12 | 0 |

Table 77: Entry Description (2095_h)

4.2.23 Object 2096_h: Chopper Hysteresis Start

This object provides the hysteresis start setting. Please notice that this value is an offset to the hysteresis end value.

| Object Description | | | |
|--------------------|--------------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 2096 _h | Chopper Hysteresis Start | Variable | UNSIGNED8 |

Table 78: Object Description (2096_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0...8 | 3 |

Table 79: Entry Description (2096_h)

4.2.24 Object 2097_h: Chopper Off Time

The off time setting controls the minimum chopper frequency. Under normal circumstances, an off time within the range of 5µs to 20µs is used. Off time setting for constant t_{OFF} chopper: $N_{CLK} = 12 + 32 * t_{OFF}$.



Minimum is 64 clocks.

Setting this parameter to zero completely disables all driver transistors and so lets the motor free-wheel.

| Object Description | | | |
|--------------------|------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 2097 _h | Chopper Off Time | Variable | UNSIGNED8 |

Table 80: Object Description (2097_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0 / 2...15 | 5 |

Table 81: Entry Description (2097_h)

4.2.25 Object 2098_h: Smart Energy Current Minimum

This object provides the setting of the lower motor current limit for CoolStep™ operation by scaling the CS value.

Possible values are:

- 0 – 1/2 of maximum motor current setting (section 4.2.4)
- 1 – 1/4 of maximum motor current setting (section 4.2.4)

| Object Description | | | |
|--------------------|------------------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 2098 _h | Smart Energy Current Minimum | Variable | UNSIGNED8 |

Table 82: Object Description (2098_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0/1 | 0 |

Table 83: Entry Description (2098_h)

4.2.26 Object 2099_h: Smart Energy Current Down Step

This object provides the setting of the number of StallGuard2 readings above the upper threshold necessary for each current decrement of the motor current.

Possible values are:

- 0 – 32 measurements – slowest decrement
- 1 – 8 measurements



- 2 – 2 measurements
- 3 – 1 measurements – fastest decrement

| Object Description | | | |
|--------------------|--------------------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 2099 _h | Smart Energy Current Down Step | Variable | UNSIGNED8 |

Table 84: Object Description (2099_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0...3 | 0 |

Table 85: Entry Description (2099_h)

4.2.27 Object 209A_h: Smart Energy Hysteresis

This object sets the distance between the lower and the upper threshold for StallGuard2 reading. Above the upper threshold the motor current becomes decreased.

Hysteresis: (SmartEnergy hysteresis value + 1) * 32

Upper StallGuard2 threshold: (SmartEnergy hysteresis start + SmartEnergy hysteresis + 1) * 32

| Object Description | | | |
|--------------------|-------------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 209A _h | Smart Energy Hysteresis | Variable | UNSIGNED8 |

Table 86: Object Description (209A_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0...15 | 0 |

Table 87: Entry Description (209A_h)

4.2.28 Object 209B_h: Smart Energy Current Up Step

This object sets the current increment step. The current becomes incremented for each measured StallGuard2 value below the lower threshold (see smart energy hysteresis start (object 209C_h, section 4.2.29). Possible values are:

- 0 – 1 step – slowest increment
- 1 – 2 steps



- 2 – 4 steps
- 3 – 8 steps – fastest increment

| Object Description | | | |
|--------------------|------------------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 209B _h | Smart Energy Current Up Step | Variable | UNSIGNED8 |

Table 88: Object Description (209B_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0...3 | 0 |

Table 89: Entry Description (209B_h)

4.2.29 Object 209C_h: Smart Energy Hysteresis Start

This object serves to set the lower threshold for the StallGuard2 value (see smart Energy current up step (section 4.2.28)). Setting this to 0 disables the CoolStep™ function.

| Object Description | | | |
|--------------------|-------------------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 209C _h | Smart Energy Hysteresis Start | Variable | UNSIGNED8 |

Table 90: Object Description (209C_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0...15 | 0 |

Table 91: Entry Description (209C_h)

4.2.30 Object 209D_h: Smart Energy Filter Enable

This object is used to set the StallGuard2 filter for more precision of the measurement. It reduces the measurement frequency to one measurement per four fullsteps if set.

In most cases it is expedient to set the filtered mode when using CoolStep™. Use the standard mode for step loss detection.

Possible values are:

- 0 – standard mode
- 1 – filtered mode



| Object Description | | | |
|--------------------|----------------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 209D _h | Smart Energy Filter Enable | Variable | UNSIGNED8 |

Table 92: Object Description (209D_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0/1 | 0 |

Table 93: Entry Description (209D_h)

4.2.31 Object 209E_h: StallGuard2 Threshold

This signed value controls the StallGuard2 threshold level for stall output and sets the optimum measurement range for readout. A lower value gives a higher sensitivity. Zero is the starting value. A higher value makes StallGuard2 less sensitive and requires more torque to indicate a stall.

| Object Description | | | |
|--------------------|-----------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 209E _h | StallGuard2 Threshold | Variable | SIGNED8 |

Table 94: Object Description (209E_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | -63...63 | 0 |

Table 95: Entry Description (209E_h)

4.2.32 Object 20A1_h: Short Protection Disable

This object is used to enable or to disable the short to ground protection. Normally there is no need to change this. Use the default value.

Possible values are:

- 0 – Short to GND protection enabled
- 1 – Short to GND protection disabled

| Object Description | | | |
|--------------------|--------------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 20A1 _h | Short Protection Disable | Variable | UNSIGNED8 |

Table 96: Object Description (20A1_h)



| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0/1 | 0 |

Table 97: Entry Description (20A1_h)

4.2.33 Object 20A3_h: Vsense

This object is used for setting the sense resistor voltage based current scaling. Use the default value and change only when recommended by Trinamic.

Possible settings are:

- 0 - Full scale sense resistor voltage is 1/18 VDD
- 1 - Full scale sense resistor voltage is 1/36 VDD

| Object Description | | | |
|--------------------|--------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 20A3 _h | Vsense | Variable | UNSIGNED8 |

Table 98: Object Description (20A3_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0/1 | 0 |

Table 99: Entry Description (20A3_h)

4.2.34 Object 20A4_h: Stop on Stall

Below this speed the motor will not be stopped. Above this speed the motor will be stopped in case the StallGuard2 load value reaches zero.

| Object Description | | | |
|--------------------|---------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 20A4 _h | Stop on Stall | Variable | UNSIGNED32 |

Table 100: Object Description (20A4_h)

| Entry Description | | | | |
|-------------------|--------|-------------|----------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0...2147483647 | 0 |

Table 101: Entry Description (20A4_h)



4.2.35 Object 20A5_n: Smart Energy Threshold Speed

The CoolStep™ functionality will be enabled when the actual speed is above this speed. It will be disabled again when the actual speed drops below this value.

| Object Description | | | |
|--------------------|------------------------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 20A5 _n | Smart Energy Threshold Speed | Variable | UNSIGNED32 |

Table 102: Object Description (20A5_n)

| Entry Description | | | | |
|-------------------|--------|-------------|----------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0...2147483647 | 0 |

Table 103: Entry Description (20A5_n)

4.2.36 Object 20B0_n: PWM Threshold Speed

The StealthChop feature will be switched on when the value of this object is greater than zero and the actual velocity is lower than the value set by this object.

| Object Description | | | |
|--------------------|---------------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 20B0 _n | PWM Threshold Speed | Variable | UNSIGNED32 |

Table 104: Object Description (20B0_n)

| Entry Description | | | | |
|-------------------|--------|-------------|--------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0...16777215 | 0 |

Table 105: Entry Description (20B0_n)

4.2.37 Object 20B1_n: PWM Gradient

Velocity dependent gradient for the PWM amplitude (StealthChop). Setting this value to 0 turns off StealthChop.



| Object Description | | | |
|--------------------|--------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 20B1 _h | PWM Gradient | Variable | UNSIGNED8 |

Table 106: Object Description (20B1_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0...255 | 0 |

Table 107: Entry Description (20B1_h)

4.2.38 Object 20B2_h: PWM Amplitude

Maximum PWM amplitude when switching to StealthChop mode. Do not set too low. Values above 64 are recommended.

| Object Description | | | |
|--------------------|---------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 20B2 _h | PWM Amplitude | Variable | UNSIGNED8 |

Table 108: Object Description (20B2_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0...255 | 0 |

Table 109: Entry Description (20B2_h)

4.2.39 Object 20B3_h: DcStep Minimum Speed

Minimum speed for switching to DcStep. The motor driver will not switch to DcStep mode when the actual velocity is below this value. Setting this object to zero completely switches off DcStep.

| Object Description | | | |
|--------------------|----------------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 20B3 _h | DcStep Minimum Speed | Variable | UNSIGNED32 |

Table 110: Object Description (20B3_h)

| Entry Description | | | | |
|-------------------|--------|-------------|--------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0...16777215 | 0 |

Table 111: Entry Description (20B3_h)

4.2.40 Object 20B4_h: DcStep Time

This setting controls the reference pulse width for DcStep load measurement. It must be optimized for robust operation with maximum motor torque. A higher value allows higher torque and higher velocity, a lower value allows operation down to a lower velocity as set by the minimum DcStep speed (object 20B3_h, see section 4.2.39).

| Object Description | | | |
|--------------------|-------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 20B4 _h | DcStep Time | Variable | UNSIGNED16 |

Table 112: Object Description (20B4_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0...1023 | 0 |

Table 113: Entry Description (20B4_h)

4.2.41 Object 20B5_h: DcStep StallGuard

This setting controls stall detection in DcStep mode. Increase this value for higher sensitivity.

| Object Description | | | |
|--------------------|-------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 20B5 _h | DcStep StallGuard | Variable | UNSIGNED8 |

Table 114: Object Description (20B5_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0...255 | 0 |

Table 115: Entry Description (20B5_h)



4.2.42 Object 20B6_h: Fullstep Threshold Speed

Depending on the settings of objects 20B7_h and 20B8_h (see sections 4.2.43 and 4.2.44) the driver switches to fullstep mode and/or to a different chopper mode when the speed set by this object is exceeded.

| Object Description | | | |
|--------------------|--------------------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 20B6 _h | Fullstep Threshold Speed | Variable | UNSIGNED32 |

Table 116: Object Description (20B6_h)

| Entry Description | | | | |
|-------------------|--------|-------------|--------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0...16777215 | 0 |

Table 117: Entry Description (20B6_h)

4.2.43 Object 20B7_h: High Speed Chopper Mode

The motor driver will switch to a different chopper mode when this object is set to 1 and the measured speed is greater than the threshold speed set by object 20B6_h (see section 4.2.42).

| Object Description | | | |
|--------------------|-------------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 20B7 _h | High Speed Chopper Mode | Variable | UNSIGNED8 |

Table 118: Object Description (20B7_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0/1 | 0 |

Table 119: Entry Description (20B7_h)

4.2.44 Object 20B8_h: High Speed Fullstep Mode

The motor driver will switch to fullstep mode when this object is set to 1 and the measured speed is greater than the threshold speed set by object 20B6_h (see section 4.2.42).

| Object Description | | | |
|--------------------|-------------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 20B8 _h | High Speed Chopper Mode | Variable | UNSIGNED8 |

Table 120: Object Description (20B8_h)



| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0/1 | 0 |

Table 121: Entry Description (20B8_h)

4.2.45 Object 20B9_h: Power Down Ramp

The value set by this object controls the number of clock cycles for motor power down after a motion as soon as the motor has stopped and the setting time (set by object 2089_h, please see section 4.2.16) has expired. The smooth transition avoids a motor jerk upon power down.

- 0=instant power down.
- 15=longest possible power down ramp.

| Object Description | | | |
|--------------------|-----------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 20B9 _h | Power Down Ramp | Variable | UNSIGNED8 |

Table 122: Object Description (20B9_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0...15 | 7 |

Table 123: Entry Description (20B9_h)

4.2.46 Object 2100_h: Home Offset Display

This object shows the home offset. The value is given in microsteps.

| Object Description | | | |
|--------------------|---------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 2100 _h | Home Offset Display | Variable | SIGNED32 |

Table 124: Object Description (2100_h)

| Entry Description | | | | |
|-------------------|--------|-------------|--------------------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | ro | no | -2147483648...2147483647 | 0 |

Table 125: Entry Description (2100_h)



4.2.47 Object 2101_h: Actual Load Value

This object shows the actual load value used for stall detection (StallGuard2™).

| Object Description | | | |
|--------------------|-------------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 2101 _h | Actual Load Value | Variable | UNSIGNED16 |

Table 126: Object Description (2101_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | ro | no | 0...1023 | 0 |

Table 127: Entry Description (2101_h)

4.2.48 Object 2102_h: Driver Error Flags

This object shows the error flags of the motor driver IC.

| Error Flags | | |
|-------------|------|-------------------------|
| Bit | Name | Meaning |
| 7 | OT | Overtemperature |
| 6 | OTPW | Temperature pre-warning |
| 5 | UV | Undervoltage |
| 4 | OCHS | Overcurrent high side |
| 3 | OLB | Open load on bridge B |
| 2 | OLA | Open load on bridge A |
| 1 | OCB | Overcurrent on bridge B |
| 0 | OCA | Overcurrent on bridge A |

Table 128: Driver Error Flags (2102_h)

| Object Description | | | |
|--------------------|--------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 2102 _h | Driver Error Flags | Variable | UNSIGNED8 |

Table 129: Object Description (2102_h)



| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | ro | no | 0...255 | 0 |

Table 130: Entry Description (2102_h)

4.2.49 Object 2107_h: Microstep Resolution Display

This object shows the microstep resolution, set by object 2000_h (please see section 4.2.1).

| Object Description | | | |
|--------------------|------------------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 2107 _h | Microstep resolution display | Variable | UNSIGNED8 |

Table 131: Object Description (2107_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | ro | no | 0...8 | 8 |

Table 132: Entry Description (2107_h)

4.2.50 Object 210B_h: Step Counter

This object shows the overall number of microsteps done by this motor so far. The value can be read as a 64 bit value (sub-index 3) or split into two 32 bit values (sub-index 1 and sub-index 2).

| Object Description | | | |
|--------------------|--------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 210B _h | Step Counter | Variable | RECORD |

Table 133: Object Description (210B_h)

| Entry Description | | | | | |
|-------------------|----------------|--------|-------------|-----------------------------------|---------------|
| Sub-index | Description | Access | PDO Mapping | Value Range | Default Value |
| 1 | Lower 32 Bits | ro | no | 0...ffffffff _h | 0 |
| 2 | Higher 32 Bits | ro | no | 0...ffffffff _h | 0 |
| 3 | 64 Bit Value | ro | no | 0...ffffffffffffffff _h | 0 |

Table 134: Entry Description (210B_h)



4.2.51 Object 2121_h: PWM Scale Value

Actual PWM scale value used when the motor driver is operating in StealthChop mode.

| Object Description | | | |
|--------------------|-----------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 2121 _h | PWM Scale Value | Variable | UNSIGNED8 |

Table 135: Object Description (2121_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | ro | no | 0...255 | — |

Table 136: Entry Description (2121_h)

4.2.52 Object 2122_h: Measured Velocity

This object contains the velocity measured by the motor driver. This value is important only when the motor driver is operating in DcStep mode.

| Object Description | | | |
|--------------------|-------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 2122 _h | Measured Velocity | Variable | SIGNED32 |

Table 137: Object Description (2122_h)

| Entry Description | | | | |
|-------------------|--------|-------------|----------------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | ro | no | -16777215...16777215 | — |

Table 138: Entry Description (2122_h)

4.2.53 Object 2702_h: Device Digital Inputs

Bits 23...16 of this object reflect the states of the general purpose inputs of the module. The number of available inputs depends on the module type.



| Bit Definitions | |
|-----------------|-------------|
| Bit | Description |
| 16 | IN0 |
| 17 | IN1 |
| 18 | IN2 |
| 19 | IN3 |

Table 139: Bit Definitions (2702_h)

| Object Description | | | |
|--------------------|-----------------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 2702 _h | Device Digital Inputs | Variable | UNSIGNED32 |

Table 140: Object Description (2702_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | ro | yes | - | 0 |

Table 141: Entry Description (2702_h)

4.2.54 Object 2703_h: Device Digital Outputs

With this object the digital outputs (general purpose outputs) can be set. Bits 23... 16 of sub index 1 switch the outputs of the module. Bits 23... 16 of sub index 2 determine which outputs can be switched. The number of available digital outputs depends on the module type.

| Bit Definitions | |
|-----------------|-------------|
| Bit | Description |
| 16 | OUT0 |
| 17 | OUT1 |
| 18 | OUT2 |
| 19 | OUT3 |

Table 142: Bit Definitions (2703_h)



| Object Description | | | |
|--------------------|------------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 2703 _h | Device Digital Outputs | Variable | ARRAY |

Table 143: Object Description (2703_h)

| Entry Description | | | | | |
|-------------------|------------------|--------|-------------|-------------|---------------|
| Sub-index | Description | Access | PDO Mapping | Value Range | Default Value |
| 1 | Physical outputs | rw | yes | UNSIGNED32 | 0 |
| 2 | Output mask | rw | yes | UNSIGNED32 | 0 |

Table 144: Entry Description (2703_h)

Note Some outputs can also be used for automatically controlling a brake that can be connected to the module via these outputs. In order to be able to control these outputs via this object the automatic brake control function has to be disabled. Do this by writing 0 to sub-index 1 and sub-index 2 of object 200C_h.

4.2.55 Object 270E_h: Device Analog Inputs

This object shows the values of the analog inputs of the device.

| Object Description | | | |
|--------------------|----------------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 270E _h | Device Analog Inputs | Array | UNSIGNED32 |

Table 145: Object Description (270E_h)

| Entry Description | | | | | |
|-------------------|----------------|--------|-------------|-------------|---------------|
| Sub-index | Description | Access | PDO Mapping | Value Range | Default Value |
| 1 | Analog input 0 | ro | yes | 0...65535 | - |
| 2 | Analog input 1 | ro | yes | 0...65535 | - |
| 3 | Analog input 2 | ro | yes | 0...65535 | - |
| 4 | Analog input 3 | ro | yes | 0...65535 | - |

Table 146: Entry Description (270E_h)



5 Profile Specific Area

The profile segment contains CiA-402 standard motion control objects. These objects control the motion control functions of the TMCM-3215. Since it is not possible to operate the modes in parallel, the user is able to activate the required function by selecting a mode of operation. The control device writes to the modes of operation object in order to select the operation mode. The drive device provides the modes of operation display object to indicate the actual activated operation mode. Controlword, statusword, and set-points are used mode-specific. This implies the responsibility of the control device to avoid inconsistencies and erroneous behavior.

The following operating modes (selectable via object 6060_h, please see 5.1.6) are implemented on the TMCM-3215:

- Profile position mode (pp)
- Profile velocity mode (pv)
- Homing mode (hm)
- Cyclic position mode (csp)
- Cyclic velocity mode (csv)

i Info

This section of the manual only shows the object indices for motor #0. Of course the same objects are also available for the other motors. For the other motors, add an offset of $motor_number \cdot 800_h$ to the object index. So for example the control word (object 6040_h for motor #0) can be accessed as object 6840_h for motor #1 and as object 7040_h for motor #2. Please see also section 2.4.1.

| Multi-axis Object Indices | |
|---------------------------|---------------------------------------|
| Motor | Object Index Range |
| Motor #0 | 6000 _h – 67FF _h |
| Motor #1 | 6800 _h – 6FFF _h |
| Motor #2 | 7000 _h – 77FF _h |

Table 147: Multi-axis Object Indices (Profile specific Area)

5.1 Detailed Object Specifications

5.1.1 Object 605A_h: Quick Stop Option Code

This object indicates what action is performed when the quick stop function is executed. The slow down ramp is the deceleration value of the used mode of operation. The following quick stop option codes are supported in the current version of the CANopen firmware:



| Value Definition | |
|------------------|--|
| Value | Definition |
| 1 | Slow down on <i>slow down ramp</i> and transit into <i>switch on disabled</i> |
| 2 | Slow down on <i>quick stop ramp</i> and transit into <i>switch on disabled</i> |
| 5 | Slow down on <i>slow down ramp</i> and stay in <i>quick stop active</i> |
| 6 | Slow down on <i>quick stop ramp</i> and stay in <i>quick stop active</i> |

Table 148: Value Description (605A_h)

| Object Description | | | |
|--------------------|------------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 605A _h | Quick stop option code | Variable | SIGNED16 |

Table 149: Object Description (605A_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 1/2/5/6 | 2 |

Table 150: Entry Description (605A_h)

5.1.2 Object 605B_h: Shutdown Option Code

This object indicates what action is performed if there is a transition from *operation enabled* state to *ready to switch on state*. The shutdown option code always has the value 0 as only this is supported.

| Value Definition | |
|------------------|---|
| Value | Definition |
| 0 | Disable drive function (switch off the power stage) |

Table 151: Value Description (605B_h)

| Object Description | | | |
|--------------------|----------------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 605B _h | Shutdown option code | Variable | UNSIGNED16 |

Table 152: Object Description (605B_h)



| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0 | 0 |

Table 153: Entry Description (605B_h)

5.1.3 Object 605C_h: Disable Operation Option Code

This object indicates what action is performed if there is a transition from *operation enabled* state to *switched on* state. The disable operation option code always has the value 1 as only this is supported. The slow down ramp is the deceleration value of the used mode of operation.

| Value Definition | |
|------------------|-----------------------------|
| Value | Definition |
| 1 | Slow down on slow down ramp |

Table 154: Value Description (605C_h)

| Object Description | | | |
|--------------------|-------------------------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 605C _h | Disable operation option code | Variable | UNSIGNED16 |

Table 155: Object Description (605C_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 1 | 1 |

Table 156: Entry Description (605C_h)

5.1.4 Object 605D_h: Halt Option Code

This object indicates what action is performed when the halt function is executed. The slow down ramp is the deceleration value of the used mode of operation.

| Value Definition | |
|------------------|--|
| Value | Definition |
| 1 | Slow down on slow down ramp and stay in <i>operation enabled</i> |

Table 157: Value Description (605D_h)

| Object Description | | | |
|--------------------|------------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 605D _h | Halt option code | Variable | UNSIGNED16 |

Table 158: Object Description (605D_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 1 | 1 |

Table 159: Entry Description (605D_h)

5.1.5 Object 605E_h: Fault Reaction Option Code

This object indicates what action is performed when fault is detected in the power drive system. The slow down ramp is the deceleration value of the used mode of operation. The fault reaction option code always has the value 2 as only this is supported.

| Value Definition | |
|------------------|------------------------------|
| Value | Definition |
| 2 | Slow down on quick stop ramp |

Table 160: Value Description (605E_h)

| Object Description | | | |
|--------------------|----------------------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 605E _h | Fault reaction option code | Variable | UNSIGNED16 |

Table 161: Object Description (605E_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 2 | 2 |

Table 162: Entry Description (605E_h)

5.1.6 Object 6060_h: Modes of Operation

This object indicates the requested operation mode. Supported operating modes are:



| Value Definition | |
|------------------|--|
| Value | Mode |
| 0 | No mode |
| 1 | Profile position mode (pp) |
| 3 | Profile velocity mode (pv) |
| 6 | Homing mode (hm) |
| 8 | Cyclic synchronous position mode (csp) |
| 9 | Cyclic synchronous velocity mode (csv) |

Table 163: Value Description (6060_h)

The motor will not run when the operating mode is set to 0. It will be stopped when the motor is running in one of the supported operating modes and the operating mode is then switched to 0.

| Object Description | | | |
|--------------------|--------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 6060 _h | Modes of operation | Variable | SIGNED8 |

Table 164: Object Description (6060_h)

| Entry Description | | | | |
|-------------------|--------|------------------|---------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | refer to CiA-402 | see table 163 | 0 |

Table 165: Entry Description (6060_h)

5.1.7 Object 6061_h: Modes of Operation Display

This object shows the operating mode that is currently set.

| Value Definition | |
|------------------|--|
| Value | Mode |
| 0 | No mode |
| 1 | Profile position mode (pp) |
| 3 | Profile velocity mode (pv) |
| 6 | Homing mode (hm) |
| 8 | Cyclic synchronous position mode (csp) |
| 9 | Cyclic synchronous velocity mode (csv) |

Table 166: Value Description (6061_h)



The motor will not run when the operating mode is set to 0. It will be stopped when the motor is running in one of the supported operating modes and the operating mode is then switched to 0.

| Object Description | | | |
|--------------------|----------------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 6061 _h | Modes of operation display | Variable | SIGNED8 |

Table 167: Object Description (6061_h)

| Entry Description | | | | |
|-------------------|--------|------------------|---------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | refer to CiA-402 | see table 166 | 0 |

Table 168: Entry Description (6061_h)

5.1.8 Object 606A_h: Sensor Selection Code

This object provides the source of the position sensor actual value. It selects whether an encoder is to be used or not.

| Value Definition | |
|------------------|--------------|
| Value | Mode |
| 0 | Encoder used |
| -1 | No encoder |

Table 169: Value Description (606A_h)

| Object Description | | | |
|--------------------|-----------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 606A _h | Sensor selection code | Variable | SIGNED16 |

Table 170: Object Description (606A_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0/-1 | -1 |

Table 171: Entry Description (606A_h)



5.1.9 Object 608F_h: Position Encoder Resolution

This object defines the resolution of the encoder. The position encoder resolution is calculated by the following formula:

$$position\ encoder\ resolution = \frac{encoder\ increments}{motor\ revolutions}$$

All values are dimensionless.

| Object Description | | | |
|--------------------|-----------------------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 608F _h | Position Encoder Resolution | Array | UNSIGNED32 |

Table 172: Object Description (608F_h)

| Entry Description | | | | | |
|-------------------|-----------------------------|--------|-------------|----------------|---------------|
| Sub-index | Description | Access | PDO Mapping | Value Range | Default Value |
| 0 | Highest sub-index supported | ro | no | 2 | 2 |
| 1 | Encoder increments | rw | no | 0...2147483647 | 1 |
| 2 | Motor revolutions | ro | no | 1 | 1 |

Table 173: Entry Description (608F_h)

5.1.10 Object 60FD_h: Digital Inputs

This object contains the states of the digital inputs of the module. Starting from bit 0, every bit reflects the state of one digital input. The number of valid bits depends on the number of digital inputs on the module used.

| Object Description | | | |
|--------------------|----------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 60FD _h | Digital inputs | Variable | UNSIGNED32 |

Table 174: Object Description (60FD_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | mappable | UNSIGNED32 | 0 |

Table 175: Entry Description (60FD_h)

5.1.11 Object 6502_h: Supported Drive Modes

This object provides information on the supported drive modes. A bit that is set means that the mode is supported, a bit that is not set means that the mode is not supported by the drive.



| Value Definition | |
|------------------|--|
| Bit | Mode |
| 0 | Profile position mode (pp) |
| 1 | Velocity mode (vl) |
| 2 | Profile velocity mode (pv) |
| 3 | Torque mode (tq) |
| 4 | Reserved |
| 5 | Homing mode (hm) |
| 6 | Interpolated position mode (ip) |
| 7 | Cyclic synchronous position mode (csp) |
| 8 | Cyclic synchronous velocity mode (csv) |
| 9 | Cyclic synchronous torque mode (cst) |

Table 176: Value Definition (6502_h)

| Object Description | | | |
|--------------------|-----------------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 6502 _h | Supported drive modes | Variable | UNSIGNED32 |

Table 177: Object Description (6502_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|-----------------------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | ro | no | UNSIGNED32 | Depends on supported modes. |

Table 178: Entry Description (6502_h)



6 Profile Position Mode

A target position is applied to the trajectory generator. It is generating a position demand value for the position control loop described in the position control function.

Please refer to object 6060_h (section 5.1.6) for information about how to choose an operation mode. Object 6061_h (section 5.1.7) shows the operation mode that is set.

6.1 Detailed Object Specifications

The following text offers detailed object specifications. For a better understanding, it is necessary to see how the state machine works.

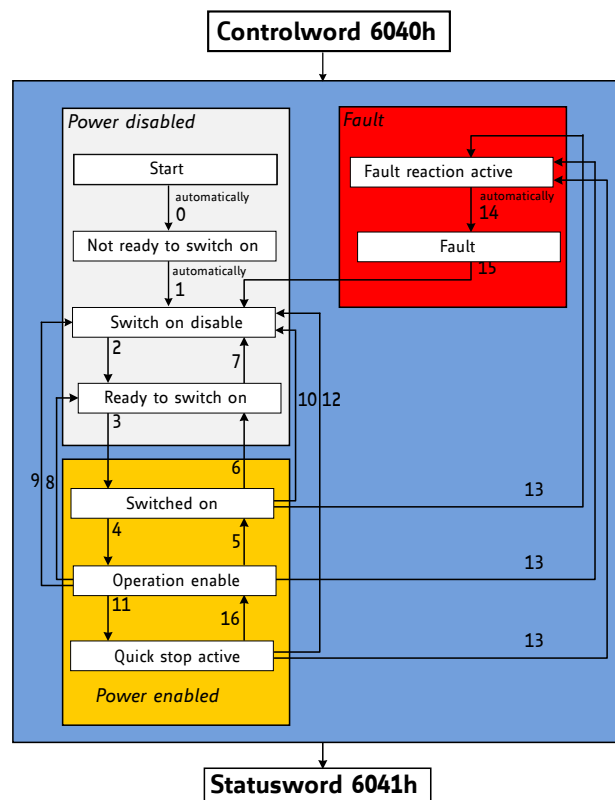


Figure 8: DS402 Finite State Machine

Notes on state transitions:

- Commands directing a change in state are processed completely and the new state achieved before additional state change commands are processed.
- Transitions 0 and 1 occur automatically at drive power-on or reset. Transition 14 occurs automatically, too. All other state changes must be directed by the host.
- Drive function disabled indicates that no current is being supplied to the motor.
- Drive function enabled indicates that current is available for the motor and profile position and profile velocity reference values may be processed.



6.1.1 Object 6040_n: Control Word

This object indicates the received command controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to figure 8 for detailed information.

| Structure of the Control Word | | | | | | | | | | | |
|-------------------------------|----|-----|---|----|-----|----|----|----|----|-----|---|
| 15 | 11 | 10 | 9 | 8 | 7 | 6 | 4 | 3 | 2 | 1 | 0 |
| nu | r | oms | h | fr | oms | eo | qs | ev | so | | |
| MSB | | | | | | | | | | LSB | |

Legend: nu=not used; r=reserved; oms=operation mode specific; h=halt; fr=fault reset; eo=enable operation; qs=quick stop; ev=enable voltage; so=switch on.

Table 179: Structure of the Control Word in pp Mode

| Operation Mode specific Bits in pp Mode | | |
|---|---------------------|--|
| Bit | Name | Definition |
| 4 | New set point | 0-to-1: the next positioning will be started. |
| 5 | Change immediately | Not supported. |
| 6 | Absolute / relative | 0: New position is absolute. 1: New position is relative. |
| 9 | Change set point | Not supported. |

Table 180: Operation Mode specific Bits in pp Mode

| Command Coding | | | | | | |
|------------------------------|----------------------|-------|-------|-------|-------|-------------|
| Command | Bits of Control Word | | | | | Transitions |
| | Bit 7 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
| Shutdown | 0 | x | 1 | 1 | 0 | 2,6,8 |
| Switch on | 0 | 0 | 1 | 1 | 1 | 3 |
| Switch on & enable operation | 0 | 1 | 1 | 1 | 1 | 3, 4 |
| Disable voltage | 0 | x | x | 0 | x | 7,9,10,12 |
| Quick stop | 0 | x | 0 | 1 | x | 7,10,11 |
| Disable operation | 0 | 0 | 1 | 1 | 1 | 5 |
| Enable operation | 0 | 1 | 1 | 1 | 1 | 4, 16 |
| Fault reset | 0-to-1 | x | x | x | x | 15 |

Table 181: Command Coding



| Object Description | | | |
|--------------------|-------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 6040 _h | Controlword | Variable | UNSIGNED16 |

Table 182: Object Description (6040_h in pp Mode)

| Entry Description | | | | |
|-------------------|--------|--------------|---------------------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | see CiA402-3 | See command coding above. | |

Table 183: Entry Description (6040_h in pp Mode)

6.1.2 Object 6041_h: Status Word

This object provides the status of the PDS FSA. It reflects the status of the CiA-402 state machine. Please refer to figure 8 for detailed information. The object is structured as defined below. For more information about the coding please refer to the CANopen Drives and motion control device profile, part 2.

| Structure of the Status Word | | | | | | | | | | | | | | | |
|------------------------------|-----|-----|-----|----|----|----|---|-----|----|----|---|----|----|------|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| dir | mot | oms | ila | tr | rm | ms | w | sod | qs | ve | f | oe | so | rtso | |
| MSB | | | | | | | | | | | | | | LSB | |

Legend: nu=not used; r=reserved; oms=operation mode specific; h=halt; fr=fault reset; oe=operation enable; qs=quick stop; ve=voltage enable; so=switch on.

Table 184: Structure of the Status Word in pp Mode

| Trinamic Specific Bits | | |
|------------------------|-----------------------|---|
| Bit | Name | Definition |
| 14 | Motor activity | 0: Motor stands still. 1: Motor rotates. |
| 15 | Direction of rotation | This bit shows the direction of rotation. |

Table 185: Trinamic Specific Bits



| Operation Mode specific Bits in pp Mode | | |
|---|------------------------|---|
| Bit | Name | Definition |
| 10 | Target reached | Set when the motor is within the position window. |
| 12 | Set point acknowledged | 0: Set point processed. 1: Set point still in process. |
| 13 | Following error | Not supported. |

Table 186: Operation Mode specific Bits in pp Mode

| State Coding | |
|----------------------------------|------------------------|
| Status word | FSA state |
| xxxx xxxx x0xx 0000 _h | Not ready to switch on |
| xxxx xxxx x1xx 0000 _h | Switch on disabled |
| xxxx xxxx x01x 0001 _h | Ready to switch on |
| xxxx xxxx x01x 0011 _h | Switched on |
| xxxx xxxx x01x 0111 _h | Operation enabled |
| xxxx xxxx x00x 0111 _h | Quick stop active |
| xxxx xxxx x0xx 1111 _h | Fault reaction active |
| xxxx xxxx x0xx 1000 _h | Fault |

Table 187: State Coding

| Object Description | | | |
|--------------------|-------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 6041 _h | Controlword | Variable | UNSIGNED16 |

Table 188: Object Description (6041_h in pp Mode)

| Entry Description | | | | |
|-------------------|--------|--------------|-------------------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | see CiA402-3 | See state coding above. | |

Table 189: Entry Description (6041_h in pp Mode)

6.1.3 Object 6062_h: Position Demand Value

This object provides the demanded position value. The value is given in microsteps. Object 6062_h indicates the actual position that the motor should have. It is not to be confused with objects 6063_h and 6064_h.



| Object Description | | | |
|--------------------|-----------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 6062 _h | Position Demand Value | Variable | SIGNED32 |

Table 190: Object Description (6062_h)

| Entry Description | | | | |
|-------------------|--------|-------------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | ro | Refer to CiA402-3 | SIGNED32 | no |

Table 191: Entry Description (6062_h)

6.1.4 Object 6063_h: Position Actual Internal Value

This object provides the actual value of the encoder or the motor. Please use the sensor selection object 606A_h (see section 5.1.8) for selecting the motor or the encoder first. Object 6063_h indicates the actual position of the encoder or the motor, re-scaled to the microstep resolution. The value is given in microsteps.

| Object Description | | | |
|--------------------|--------------------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 6063 _h | Position Actual Internal Value | Variable | SIGNED32 |

Table 192: Object Description (6063_h)

| Entry Description | | | | |
|-------------------|--------|-------------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | ro | Refer to CiA402-3 | SIGNED32 | no |

Table 193: Entry Description (6063_h)

6.1.5 Object 6064_h: Position Actual Value

This object provides the actual value of the position measurement device. It always contains the same value as object 6063_h.

| Object Description | | | |
|--------------------|-----------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 6064 _h | Position Actual Value | Variable | SIGNED32 |

Table 194: Object Description (6064_h)



| Entry Description | | | | |
|-------------------|--------|-------------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | ro | Refer to CiA402-3 | SIGNED32 | no |

Table 195: Entry Description (6064_h)

6.1.6 Object 6065_h: Following Error Window

This object indicates the configured range of tolerated position values symmetrically to the position demand value. If the position actual value is out of the following error window, a following error occurs. A following error may occur when a drive is blocked, unreachable profile velocity occurs, or at wrong closed-loop coefficients. The value shall be given in microsteps.

When the difference between motor position (object 6062_h) and encoder position (object 6063_h or 6064_h) is greater than the value set here, the motor will be stopped and an emergency message will be sent. Setting this object to zero will turn off this feature completely.

Note Setting this object to a too low value will lead to false alarms.

| Object Description | | | |
|--------------------|------------------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 6065 _h | Following Error Window | Variable | UNSIGNED32 |

Table 196: Object Description (6065_h)

| Entry Description | | | | |
|-------------------|--------|-------------|----------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0...2147483647 | 0 |

Table 197: Entry Description (6065_h)

6.1.7 Object 6067_h: Position Window

This object indicates the configured symmetrical range of accepted positions relative to the target position. If the actual value of the position encoder is within the position window, this target position is regarded as having been reached. The value is given in increments. If the value of the position window is FFFFFFFF_h, the position window control is switched off. If this object is set to zero, the target reached event will be signaled when the demand position (6062_h) has reached the target position (6064_h). When the position window is set to a value greater than zero, the target reached event will be signaled when the actual encoder position value (6064_h) is within $(target_position - position_window)$ and $(target_position + position_window)$.



| Object Description | | | |
|--------------------|-----------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 6067 _h | Position Window | Variable | UNSIGNED32 |

Table 198: Object Description (6067_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|----------------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | UNSIGNED32 | FFFFFFF _h |

Table 199: Entry Description (6067_h)

6.1.8 Object 6068_h: Position Window Time

This object indicates the configured time, during which the actual position within the position window is measured. The value is given in ms. If this object is set to a value greater than zero and also the position window (6067_h) is set to a value greater than zero the target reached event will not be signaled until the actual position (6064_h) is at least as many milliseconds within the position window as defined by this object.

| Object Description | | | |
|--------------------|----------------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 6068 _h | Position Window Time | Variable | UNSIGNED16 |

Table 200: Object Description (6068_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | UNSIGNED16 | 0 |

Table 201: Entry Description (6068_h)

6.1.9 Object 606C_h: Velocity Actual Value

This object shows the actual velocity value of the motor. The value is given in internal or user-defined velocity units (depending on object 208C_h, described in section 4.2.17).

| Object Description | | | |
|--------------------|-----------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 606C _h | Velocity Actual Value | Variable | SIGNED32 |

Table 202: Object Description (606C_h)

| Entry Description | | | | |
|-------------------|--------|-------------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | ro | Refer to CiA402-3 | SIGNED32 | no |

Table 203: Entry Description (606C_h)

6.1.10 Object 607A_h: Target Position

The target position is the position that the drive should move to in profile position mode using the current settings of motion control parameters (such as velocity, acceleration, deceleration, motion profile type etc.). The value of this object is interpreted as absolute or relative depending on the abs/rel flag in the controlword. It is given in microsteps.

| Object Description | | | |
|--------------------|-----------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 607A _h | Target Position | Variable | SIGNED32 |

Table 204: Object Description (607A_h in pp Mode)

| Entry Description | | | | |
|-------------------|--------|-------------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | Refer to CiA402-3 | SIGNED32 | 0 |

Table 205: Entry Description (607A_h in pp Mode)

6.1.11 Object 607D_h: Software Position Limit

This object indicates the configured maximal and minimal software position limits. These parameters define the absolute position limits for the position demand value and the position actual value. Every new target position is checked against these limits. The limit positions are always relative to the machine home position. Before being compared with the target position, they are corrected internally by the home offset as follows:

$$\text{Corrected_min_position_limit} = \text{min_position_limit} - \text{home_offset}$$

$$\text{Corrected_max_position_limit} = \text{max_position_limit} - \text{home_offset}$$

| Object Description | | | |
|--------------------|-------------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 607D _h | Software Position Limit | Array | SIGNED32 |

Table 206: Object Description (607D_h)

| Entry Description | | | | | |
|-------------------|------------------------|--------|-------------|-------------|---------------|
| Sub-index | Description | Access | PDO Mapping | Value Range | Default Value |
| 1 | Minimum Position Limit | rw | no | SIGNED32 | -2147483648 |
| 2 | Maximum Position Limit | rw | no | SIGNED32 | 2147483647 |

Table 207: Entry Description (607D_h)

6.1.12 Object 6081_h: Profile Velocity

This object indicates the configured velocity normally attained at the end of the acceleration ramp during a profiled motion and is valid for both directions of motion. The profile velocity is the maximum velocity used when driving to a new position. It is given in internal or user specific units (depending on object 208C_h, section 4.2.17).

| Object Description | | | |
|--------------------|------------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 6081 _h | Profile Velocity | Variable | UNSIGNED32 |

Table 208: Object Description (6081_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | UNSIGNED32 | 0 |

Table 209: Entry Description (6081_h)

6.1.13 Object 6082_h: End Velocity

This object indicates the configured velocity normally attained at the end of the deceleration ramp during a profiled motion and is valid for both directions of motion. The end velocity is the velocity used when reaching the new position. It is given in internal or user specific units (depending on object 208C_h, section 4.2.17).

| Object Description | | | |
|--------------------|--------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 6082 _h | End Velocity | Variable | UNSIGNED32 |

Table 210: Object Description (6082_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | UNSIGNED32 | 0 |

Table 211: Entry Description (6082_h)



6.1.14 Object 6083_h: Profile Acceleration

This object indicates the configured acceleration. Object 6083_h sets the maximum acceleration to be used in profile position and profile velocity mode.

The units for object 6083_h can be chosen with object 208E_h, described in section 4.2.18.

In profile velocity mode, this object also sets the deceleration to be used (the deceleration ramp is always the same as the acceleration ramp in pv mode).

| Object Description | | | |
|--------------------|----------------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 6083 _h | Profile Acceleration | Variable | UNSIGNED32 |

Table 212: Object Description (6083_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | UNSIGNED32 | 0 |

Table 213: Entry Description (6083_h)

6.1.15 Object 6084_h: Profile Deceleration

This object indicates the configured deceleration. Object 6084_h sets the maximum deceleration to be used in profile positioning mode.

The units for object 6084_h can be chosen with object 208E_h, described in section 4.2.18.

| Object Description | | | |
|--------------------|----------------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 6084 _h | Profile Deceleration | Variable | UNSIGNED32 |

Table 214: Object Description (6084_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | UNSIGNED32 | 0 |

Table 215: Entry Description (6084_h)

6.1.16 Object 6085_h: Quick Stop Deceleration

This object indicates the configured deceleration used to stop the motor when the quick stop function is activated and the quick stop code object 605A_h is set to 2 (or 6). The value is given in the same unit as profile acceleration object 6083_h.



| Object Description | | | |
|--------------------|-------------------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 6085 _h | Quick stop deceleration | Variable | UNSIGNED32 |

Table 216: Object Description (6085_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | UNSIGNED32 | 51200 |

Table 217: Entry Description (6085_h)

6.1.17 Object 60F2_h: Positioning Option Code

This object indicates the positioning behaviour in profile position mode. Only bits 0 and 1 (relative option) are supported.

| Bit Definitions | | |
|-----------------|-------|--|
| Bit 1 | Bit 0 | Definition |
| 0 | 0 | Positioning moves shall be performed relative to the preceding (internal absolute) target position. |
| 0 | 1 | Positioning moves shall be performed relative to the actual position demand value (object 6063 _h). |
| 1 | 0 | Positioning moves shall be performed relative to the position actual value (object 6064 _h). |
| 1 | 1 | reserved |

Table 218: Bit Definitions of Object 60F2_h

| Object Description | | | |
|--------------------|-------------------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 60F2 _h | Positioning option code | Variable | UNSIGNED16 |

Table 219: Object Description (60F2_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | UNSIGNED16 | 0 |

Table 220: Entry Description (60F2_h)

6.2 How to move a Motor in pp Mode

Here is a little example that shows how to get a motor running in pp mode. In this little example we assume that the module has been reset (and then switched to pre-operational or operational) by NMT commands before. Please note that the values are decimal.

- If you do not have any limit switches connected, first disable the limit switch inputs by writing 3 to object 2005_h.
- Select pp mode by writing 1 to object 6060_h.
- Write 6 to object 6040_h to switch to READY_TO_SWITCH_ON state.
- Write 7 to object 6040_h to switch to SWITCHED_ON state.
- Write 15 to object 6040_h to switch to OPERATION_ENABLED state.
- Write the desired target position (e.g. 500000) to object 607A_h.
- Mark the new target position as active by writing 31 to object 6040_h. The motor starts moving now.
- Reset the activation by writing 15 to object 6040_h (this can be done while the motor is still moving).



7 Profile Velocity Mode

The profile velocity mode is used to control the velocity of the drive without a special regard of the position. It contains limit functions and trajectory generation.

The profile velocity mode covers the following sub-functions:

- Demand value input via trajectory generator.
- Monitoring of the profile velocity using a window-function.
- Monitoring of velocity actual value using a threshold.

The operation of the reference value generator and its input parameters include:

- Profile velocity
- Profile acceleration
- Profile deceleration
- Emergency stop
- Motion profile type

7.1 Detailed Object Specifications

7.1.1 Object 6040_n: Control Word

This object indicates the received command controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to figure 8 for detailed information.

In pv mode the control word does not contain any operation mode specific bits.

| Structure of the Control Word | | | | | | | | | | | |
|-------------------------------|----|----|---|----|---|-----|----|----|----|---|---|
| 15 | 11 | 10 | 9 | 8 | 7 | 6 | 4 | 3 | 2 | 1 | 0 |
| nu | r | r | h | fr | r | eo | qs | ev | so | | |
| MSB | | | | | | LSB | | | | | |

Legend: nu=not used; r=reserved; h=halt; fr=fault reset; eo=enable operation; qs=quick stop; ev=enable voltage; so=switch on.

Table 221: Structure of the Control Word in pv Mode



| Command Coding | | | | | | |
|------------------------------|----------------------|-------|-------|-------|-------|-------------|
| Command | Bits of Control Word | | | | | Transitions |
| | Bit 7 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
| Shutdown | 0 | x | 1 | 1 | 0 | 2,6,8 |
| Switch on | 0 | 0 | 1 | 1 | 1 | 3 |
| Switch on & enable operation | 0 | 1 | 1 | 1 | 1 | 3, 4 |
| Disable voltage | 0 | x | x | 0 | x | 7,9,10,12 |
| Quick stop | 0 | x | 0 | 1 | x | 7,10,11 |
| Disable operation | 0 | 0 | 1 | 1 | 1 | 5 |
| Enable operation | 0 | 1 | 1 | 1 | 1 | 4, 16 |
| Fault reset | 0-to-1 | x | x | x | x | 15 |

Table 222: Command Coding

| Object Description | | | |
|--------------------|-------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 6040 _h | Controlword | Variable | UNSIGNED16 |

Table 223: Object Description (6040_h in pv Mode)

| Entry Description | | | | |
|-------------------|--------|--------------|---------------------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | see CiA402-3 | See command coding above. | |

Table 224: Entry Description (6040_h in pv Mode)

7.1.2 Object 6041_h: Status Word

This object provides the status of the PDS FSA. It reflects the status of the CiA-402 state machine. Please refer to figure 8 for detailed information. The object is structured as defined below. For more information about the coding please refer to the CANopen Drives and motion control device profile, part 2.

| Structure of the Status Word | | | | | | | | | | | | | | | |
|------------------------------|-----|-----|-----|----|----|----|---|-----|----|----|---|----|----|------|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| dir | mot | oms | ila | tr | rm | ms | w | sod | qs | ve | f | oe | so | rtso | |
| MSB | | | | | | | | | | | | | | LSB | |

Legend: nu=not used; r=reserved; oms=operation mode specific; h=halt; fr=fault reset; oe=operation enable; qs=quick stop; ve=voltage enable; so=switch on.

Table 225: Structure of the Status Word in pv Mode



| Trinamic Specific Bits | | |
|------------------------|-----------------------|---|
| Bit | Name | Definition |
| 14 | Motor activity | 0: Motor stands still. 1: Motor rotates. |
| 15 | Direction of rotation | This bit shows the direction of rotation. |

Table 226: Trinamic Specific Bits

| Operation Mode specific Bits in pv Mode | | |
|---|---------------------|---|
| Bit | Name | Definition |
| 10 | Target reached | Indicates that the target speed has been reached. |
| 12 | Speed | Not supported. |
| 13 | Max. slippage error | Not supported. |

Table 227: Operation Mode specific Bits in pv Mode

| State Coding | |
|----------------------------------|------------------------|
| Status word | FSA state |
| xxxx xxxx x0xx 0000 _h | Not ready to switch on |
| xxxx xxxx x1xx 0000 _h | Switch on disabled |
| xxxx xxxx x01x 0001 _h | Ready to switch on |
| xxxx xxxx x01x 0011 _h | Switched on |
| xxxx xxxx x01x 0111 _h | Operation enabled |
| xxxx xxxx x00x 0111 _h | Quick stop active |
| xxxx xxxx x0xx 1111 _h | Fault reaction active |
| xxxx xxxx x0xx 1000 _h | Fault |

Table 228: State Coding

| Object Description | | | |
|--------------------|-------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 6041 _h | Controlword | Variable | UNSIGNED16 |

Table 229: Object Description (6041_h in pv Mode)



| Entry Description | | | | |
|-------------------|--------|--------------|------------------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | see CiA402-3 | See state coding above | |

Table 230: Entry Description (6041_h in pv Mode)

7.1.3 Object 6062_h: Position Demand Value

This object provides the demanded position value. The value is given in microsteps. Object 6062_h indicates the actual position that the motor should have. It is not to be confused with objects 6063_h and 6064_h.

| Object Description | | | |
|--------------------|-----------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 6062 _h | Position Demand Value | Variable | SIGNED32 |

Table 231: Object Description (6062_h)

| Entry Description | | | | |
|-------------------|--------|-------------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | ro | Refer to CiA402-3 | SIGNED32 | no |

Table 232: Entry Description (6062_h)

7.1.4 Object 6063_h: Position Actual Internal Value

This object provides the actual value of the encoder or the motor. Please use the sensor selection object 606A_h (see section 5.1.8) for selecting the motor or the encoder first. Object 6063_h indicates the actual position of the encoder or the motor, re-scaled to the microstep resolution. The value is given in microsteps.

| Object Description | | | |
|--------------------|--------------------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 6063 _h | Position Actual Internal Value | Variable | SIGNED32 |

Table 233: Object Description (6063_h)

| Entry Description | | | | |
|-------------------|--------|-------------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | ro | Refer to CiA402-3 | SIGNED32 | no |

Table 234: Entry Description (6063_h)



7.1.5 Object 6064_h: Position Actual Value

This object provides the actual value of the position measurement device. It always contains the same value as object 6063_h.

| Object Description | | | |
|--------------------|-----------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 6064 _h | Position Actual Value | Variable | SIGNED32 |

Table 235: Object Description (6064_h)

| Entry Description | | | | |
|-------------------|--------|-------------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | ro | Refer to CiA402-3 | SIGNED32 | no |

Table 236: Entry Description (6064_h)

7.1.6 Object 6065_h: Following Error Window

This object indicates the configured range of tolerated position values symmetrically to the position demand value. If the position actual value is out of the following error window, a following error occurs. A following error may occur when a drive is blocked, unreachable profile velocity occurs, or at wrong closed-loop coefficients. The value shall be given in microsteps.

When the difference between motor position (object 6062_h) and encoder position (object 6063_h or 6064_h) is greater than the value set here, the motor will be stopped and an emergency message will be sent. Setting this object to zero will turn off this feature completely.

Note Setting this object to a too low value will lead to false alarms.

| Object Description | | | |
|--------------------|------------------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 6065 _h | Following Error Window | Variable | UNSIGNED32 |

Table 237: Object Description (6065_h)

| Entry Description | | | | |
|-------------------|--------|-------------|----------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | 0...2147483647 | 0 |

Table 238: Entry Description (6065_h)



7.1.7 Object 606C_h: Velocity Actual Value

This object shows the actual velocity value of the motor. The value is given in internal or user-defined velocity units (depending on object 208C_h, described in section 4.2.17).

| Object Description | | | |
|--------------------|-----------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 606C _h | Velocity Actual Value | Variable | SIGNED32 |

Table 239: Object Description (606C_h)

| Entry Description | | | | |
|-------------------|--------|-------------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | ro | Refer to CiA402-3 | SIGNED32 | no |

Table 240: Entry Description (606C_h)

7.1.8 Object 607D_h: Software Position Limit

This object indicates the configured maximal and minimal software position limits. These parameters define the absolute position limits for the position demand value and the position actual value. Every new target position is checked against these limits. The limit positions are always relative to the machine home position. Before being compared with the target position, they are corrected internally by the home offset as follows:

$$Corrected_min_position_limit = min_position_limit - home_offset$$

$$Corrected_max_position_limit = max_position_limit - home_offset$$

| Object Description | | | |
|--------------------|-------------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 607D _h | Software Position Limit | Array | SIGNED32 |

Table 241: Object Description (607D_h)

| Entry Description | | | | | |
|-------------------|------------------------|--------|-------------|-------------|---------------|
| Sub-index | Description | Access | PDO Mapping | Value Range | Default Value |
| 1 | Minimum Position Limit | rw | no | SIGNED32 | -2147483648 |
| 2 | Maximum Position Limit | rw | no | SIGNED32 | 2147483647 |

Table 242: Entry Description (607D_h)



7.1.9 Object 6083_h: Profile Acceleration

This object indicates the configured acceleration. Object 6083_h sets the maximum acceleration to be used in profile position and profile velocity mode.

The units for object 6083_h can be chosen with object 208E_h, described in section 4.2.18.

In profile velocity mode, this object also sets the deceleration to be used (the deceleration ramp is always the same as the acceleration ramp in pv mode).

| Object Description | | | |
|--------------------|----------------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 6083 _h | Profile Acceleration | Variable | UNSIGNED32 |

Table 243: Object Description (6083_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | UNSIGNED32 | 0 |

Table 244: Entry Description (6083_h)

7.1.10 Object 6085_h: Quick Stop Deceleration

This object indicates the configured deceleration used to stop the motor when the quick stop function is activated and the quick stop code object 605A_h is set to 2 (or 6). The value is given in the same unit as profile acceleration object 6083_h.

| Object Description | | | |
|--------------------|-------------------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 6085 _h | Quick stop deceleration | Variable | UNSIGNED32 |

Table 245: Object Description (6085_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | UNSIGNED32 | 51200 |

Table 246: Entry Description (6085_h)

7.1.11 Object 60FF_h: Target Velocity

This object indicates the configured target velocity and is used as input for the trajectory generator. Object 60FF_h sets the target velocity when using profile velocity mode. The drive then accelerates or decelerates to that velocity using the acceleration and deceleration set by objects 6083_h and 6084_h. The values are given in units which can be selected with object 208C_h, described in section 4.2.17.



| Object Description | | | |
|--------------------|-----------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 60FF _h | Target Velocity | Variable | SIGNED32 |

Table 247: Object Description (60FF_h)

| Entry Description | | | | |
|-------------------|--------|--------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | see CiA402-3 | SIGNED32 | 0 |

Table 248: Entry Description (60FF_h)

7.2 How to move a Motor in pv Mode

Here is a little example that shows how to get a motor running in pv mode. In this little example we assume that the module has been reset (and then switched to pre-operational or operational) by NMT commands before.

- If you do not have any limit switches connected, first disable the limit switch inputs by writing 3 to object 2005_h.
- Select pv mode by writing 3 to object 6060_h.
- Write 6 to object 6040_h to switch to READY_TO_SWITCH_ON state.
- Write 7 to object 6040_h to switch to SWITCHED_ON state.
- Write 15 to object 6040_h to switch to OPERATION_ENABLED state.
- Write the desired target speed (e.g. 100000) to object 60FF_h. The motor now accelerates to that speed.
- Stop the motor by writing 0 to object 60FF_h.



8 Homing Mode

This chapter describes the method by which a drive seeks the home position (reference point). There are various methods of achieving this using limit switches at the ends of travel or a home switch in mid-travel. Some methods also use the index (zero) pulse train from an incremental encoder. The user may specify the speeds, acceleration and the method of homing.

There is no output data except for those bits in the statusword which return the status or result of the homing process and the demand to the position control loops.

There are four sources of the homing signal available: these are positive and negative limit switches, the home switch and the index pulse from an encoder.

Figure 9 shows the defined input objects as well as the output objects. The user can specify the speeds, acceleration and method of homing. The home offset object 607C_h allows displacing the zero in point the coordinate system for the home position.

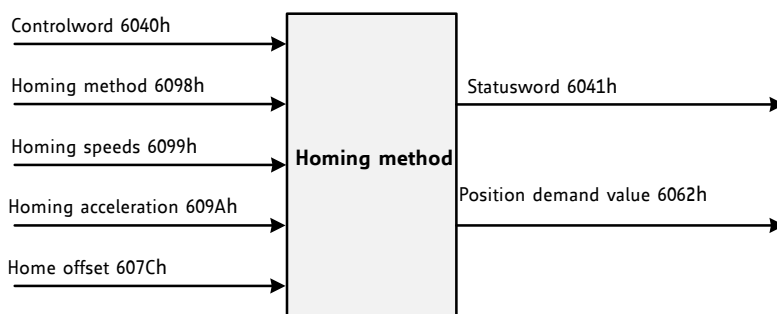


Figure 9: Homing Mode Function

Choosing a homing mode determines the following things:

- The homing signal (positive limit switch, negative limit switch, and home switch).
- The direction of actuation where appropriate.
- The position of the index pulse.

The home position and the zero position are offset by the home offset (see object 607C_h, section 8.2.4).

Depending on the module there are different sources of homing methods available:

- Negative and positive limit switches.
- Home switch.
- Index pulse of an encoder.

For the operation of positioning drives, an exact knowledge of the absolute position is normally required. Since for cost reasons drives often do not have an absolute encoder, a homing operation is necessary.



8.1 Homing Methods

The TMCM-3215 supports a subset of different standard CANopen homing methods. The homing method that is to be used can be chosen via object 6098_h (section 8.2.5).

| Supported Homing Methods | |
|--------------------------|---|
| Method | Description |
| 0 | No homing (default value for object 6098 _h). |
| 1 | Search the left end switch, then search the next encoder index pulse. |
| 2 | Search the right end switch, then search the next encoder index pulse. |
| 3 | Search the positive edge of the home switch, then search the next encoder index pulse. |
| 5 | Search the negative edge of the home switch, then search the next encoder index pulse. |
| 17 | Search the left end switch. |
| 18 | Search the right end switch. |
| 19 | Search the positive edge of the home switch. |
| 21 | Search the negative edge of the home switch. |
| 33 | Search next index pulse in negative direction. |
| 34 | Search next index pulse in positive direction. |
| 35 | The actual position is used as home position. All position values (objects 6062h, 6063h, and 6064h) are set to zero, but the motor will not move. |

Table 249: Supported CANopen Homing Methods

When using homing methods that need end switch inputs or home switch inputs please take care of their configuration (object 2005_h, section 4.2.6).

8.1.1 Homing Method 1: Homing on negative Limit Switch and Index Pulse

Using this method, the initial direction of movement shall be leftward if the negative limit switch is inactive (here: low). The home position shall be at the first index pulse to the right of the position where the negative limit switch becomes inactive.

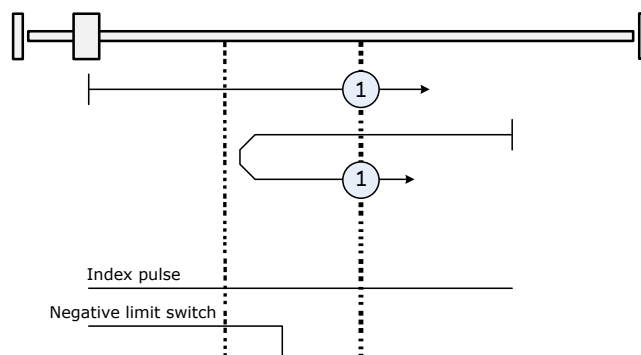


Figure 10: Homing Method 1



8.1.2 Homing Method 2: Homing on positive Limit Switch and Index Pulse

Using this method, the initial direction of movement shall be rightward if the positive limit switch is inactive (here: low). The position of home shall be at the first index pulse to the left of the position where the positive limit switch becomes inactive.

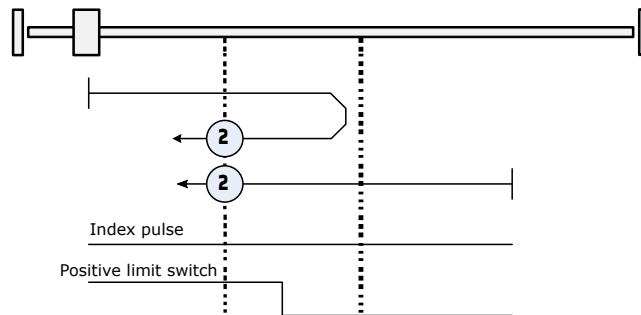


Figure 11: Homing Method 2

8.1.3 Homing Method 3: Homing on positive Home Switch and Index Pulse

Using this method, the initial direction of movement shall be dependent on the state of the home switch. The home position shall be at the index pulse to either to the left or the right of the point where the home switch changes state. If the initial position is situated so that the direction of movement shall reverse during homing, the point at which the reversal takes place is anywhere after a change of state of the home switch.

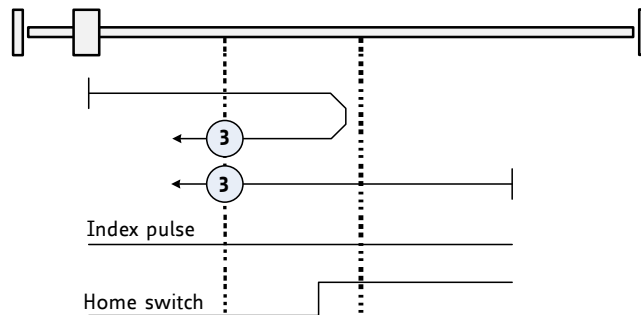


Figure 12: Homing Method 3

8.1.4 Homing Method 5: Homing on negative Home Switch and Index Pulse

Using this method, the initial direction of movement shall be dependent on the state of the home switch. The home position shall be at the index pulse to either to the left or the right of the point where the home switch changes state. If the initial position is situated so that the direction of movement shall reverse during homing, the point at which the reversal takes place is anywhere after a change of state of the home switch.



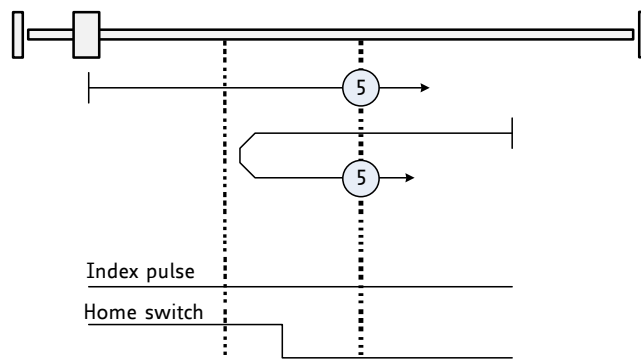


Figure 13: Homing Method 5

8.1.5 Homing Method 17: Homing on negative Limit Switch

Using this method, the initial direction of movement shall be leftward if the negative limit switch is inactive (here: low). The home position shall at the point where the negative limit switch becomes inactive.

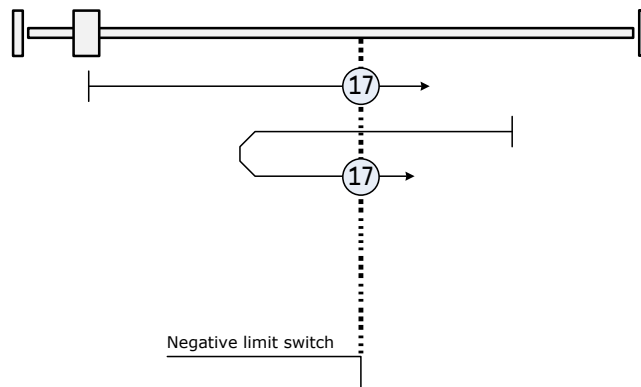


Figure 14: Homing Method 17

8.1.6 Homing Method 18: Homing on positive Limit Switch

Using this method, the initial direction of movement shall be rightward if the positive limit switch is inactive (here: low). The home position shall be at point the where the positive limit switch becomes inactive.

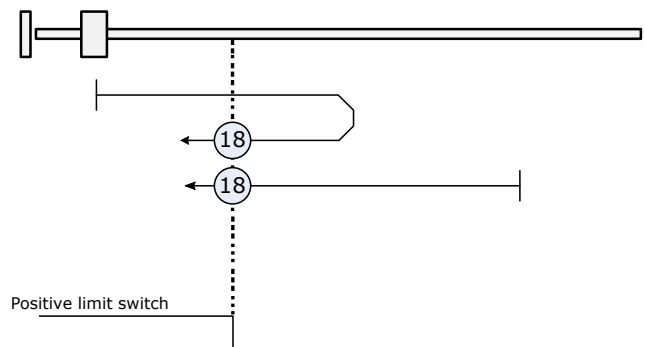


Figure 15: Homing Method 18



8.1.7 Homing Method 19: Homing on positive Home Switch

Using this method, the initial direction of movement shall be dependent on the state of the home switch. The home position shall be at the point where the home switch changes state. If the initial direction of movement leads away from the home switch, the drive shall reverse on encountering the relevant limit switch.

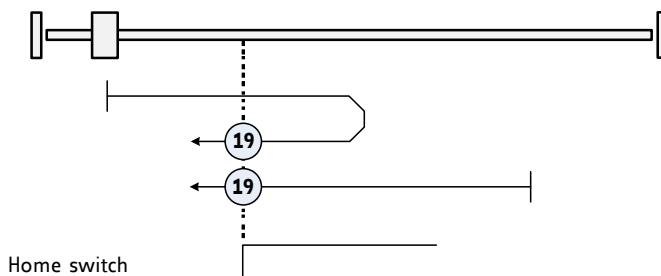


Figure 16: Homing Method 19

8.1.8 Homing Method 21: Homing on negative Home Switch

Using this method, the initial direction of movement shall be dependent on the state of the home switch. The home position shall be at the point where the home switch changes state. If the initial direction of movement leads away from the home switch, the drive shall reverse on encountering the relevant limit switch.

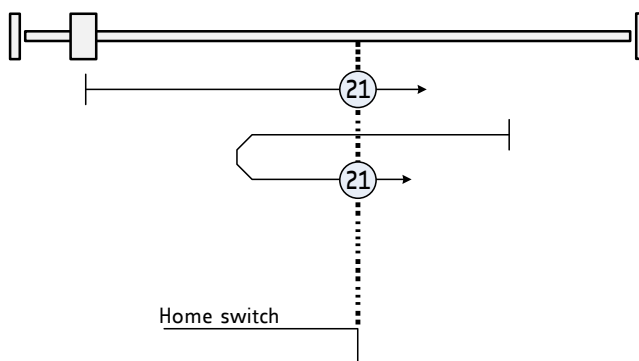


Figure 17: Homing Method 21

8.1.9 Homing Method 33 and 34: Homing on next Index Pulse

Using these methods, the direction of homing is negative or positive respectively. The home position shall be at the index pulse found in the selected direction.



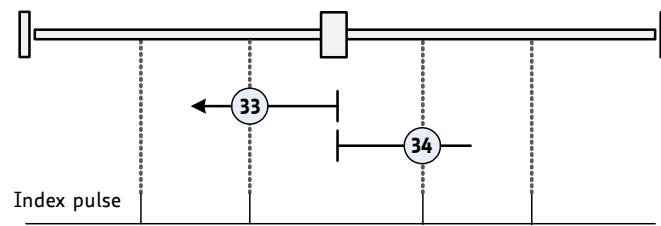


Figure 18: Homing Methods 33 and 34

8.1.10 Homing Method 35: Current Position as Home Position

In this method, the current position shall be taken to be the home position. This method does not require the drive device to be in operation enabled state.



8.2 Detailed Object Specifications

8.2.1 Object 6040_h: Control Word

This object indicates the received command controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to figure 8 for detailed information.

| Structure of the Control Word | | | | | | | | | | | |
|-------------------------------|----|-----|---|----|-----|-----|----|----|----|---|---|
| 15 | 11 | 10 | 9 | 8 | 7 | 6 | 4 | 3 | 2 | 1 | 0 |
| nu | r | oms | h | fr | oms | eo | qs | ev | so | | |
| MSB | | | | | | LSB | | | | | |

Legend: nu=not used; r=reserved; oms=operation mode specific; h=halt; fr=fault reset; eo=enable operation; qs=quick stop; ev=enable voltage; so=switch on.

Table 250: Structure of the Control Word in hm Mode

| Operation Mode specific Bits in hm Mode | | |
|---|------------------------|---------------------------------|
| Bit | Name | Definition |
| 4 | Homing operation start | 1: start homing; 0: stop homing |
| 8 | Halt | Not supported. |

Table 251: Operation Mode specific Bits in hm Mode

| Command Coding | | | | | | |
|------------------------------|----------------------|-------|-------|-------|-------|-------------|
| Command | Bits of Control Word | | | | | Transitions |
| | Bit 7 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
| Shutdown | 0 | x | 1 | 1 | 0 | 2,6,8 |
| Switch on | 0 | 0 | 1 | 1 | 1 | 3 |
| Switch on & enable operation | 0 | 1 | 1 | 1 | 1 | 3, 4 |
| Disable voltage | 0 | x | x | 0 | x | 7,9,10,12 |
| Quick stop | 0 | x | 0 | 1 | x | 7,10,11 |
| Disable operation | 0 | 0 | 1 | 1 | 1 | 5 |
| Enable operation | 0 | 1 | 1 | 1 | 1 | 4, 16 |
| Fault reset | 0-to-1 | x | x | x | x | 15 |

Table 252: Command Coding



| Object Description | | | |
|--------------------|-------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 6040 _h | Controlword | Variable | UNSIGNED16 |

Table 253: Object Description (6040_h in hm Mode)

| Entry Description | | | | |
|-------------------|--------|--------------|---------------------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | see CiA402-3 | See command coding above. | |

Table 254: Entry Description (6040_h in hm Mode)

8.2.2 Object 6041_h: Status Word

This object provides the status of the PDS FSA. It reflects the status of the CiA-402 state machine. Please refer to figure 8 for detailed information. The object is structured as defined below. For more information about the coding please refer to the CANopen Drives and motion control device profile, part 2.

| Structure of the Status Word | | | | | | | | | | | | | | | |
|------------------------------|-----|-----|-----|----|----|----|---|-----|----|----|---|----|----|------|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| dir | mot | oms | ila | tr | rm | ms | w | sod | qs | ve | f | oe | so | rtso | |
| MSB | | | | | | | | | | | | | | LSB | |

Legend: nu=not used; r=reserved; oms=operation mode specific; h=halt; fr=fault reset; oe=operation enable; qs=quick stop; ve=voltage enable; so=switch on.

Table 255: Structure of the Status Word in hm Mode

| Trinamic Specific Bits | | |
|------------------------|-----------------------|---|
| Bit | Name | Definition |
| 14 | Motor activity | 0: Motor stands still. 1: Motor rotates. |
| 15 | Direction of rotation | This bit shows the direction of rotation. |

Table 256: Trinamic Specific Bits



| Operation Mode specific Bits in hm Mode | | |
|---|----------------|--|
| Bit | Name | Definition |
| 10 | Target reached | Set when the zero position has been found or homing has been stopped by setting controlword bit 4 to zero. |
| 12 | Home attained | Set when zero position has been found. |
| 13 | Homing error | Not supported. |

Table 257: Operation Mode specific Bits in hm Mode

| State Coding | |
|----------------------------------|------------------------|
| Status word | FSA state |
| xxxx xxxx x0xx 0000 _h | Not ready to switch on |
| xxxx xxxx x1xx 0000 _h | Switch on disabled |
| xxxx xxxx x01x 0001 _h | Ready to switch on |
| xxxx xxxx x01x 0011 _h | Switched on |
| xxxx xxxx x01x 0111 _h | Operation enabled |
| xxxx xxxx x00x 0111 _h | Quick stop active |
| xxxx xxxx x0xx 1111 _h | Fault reaction active |
| xxxx xxxx x0xx 1000 _h | Fault |

Table 258: State Coding

| Object Description | | | |
|--------------------|-------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 6041 _h | Controlword | Variable | UNSIGNED16 |

Table 259: Object Description (6041_h in hm Mode)

| Entry Description | | | | |
|-------------------|--------|--------------|-------------------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | see CiA402-3 | See state coding above. | |

Table 260: Entry Description (6041_h in hm Mode)

8.2.3 Object 606C_h: Velocity Actual Value

This object shows the actual velocity value of the motor. The value is given in internal or user-defined velocity units (depending on object 208C_h, described in section 4.2.17).



| Object Description | | | |
|--------------------|-----------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 606C _h | Velocity Actual Value | Variable | SIGNED32 |

Table 261: Object Description (606C_h)

| Entry Description | | | | |
|-------------------|--------|-------------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | ro | Refer to CiA402-3 | SIGNED32 | no |

Table 262: Entry Description (606C_h)

8.2.4 Object 607C_h: Home Offset

This object indicates the configured difference between the zero position for the application and the machine home position/home switch (found during homing). While homing, the machine home position is found and once the homing is completed, the zero position is offset from the home position by adding the home offset to the home position. The effect of setting the home position to a non-zero value depends on the selected homing method. The value of this object is given in microsteps. Negative values indicate the opposite direction.

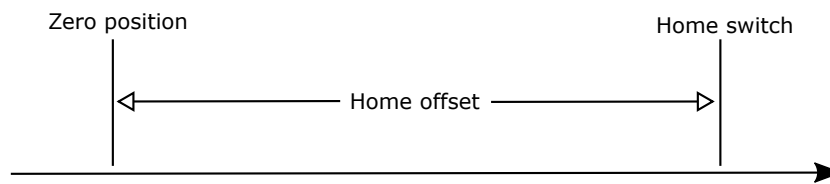


Figure 19: Home Offset

| Object Description | | | |
|--------------------|-------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 607C _h | Home offset | Variable | SIGNED32 |

Table 263: Object Description (607C_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | SIGNED32 | 0 |

Table 264: Entry Description (607C_h)



8.2.5 Object 6098_h: Homing Method

The homing method to be used can be selected by writing to this object. Please see table 249 for a list of homing methods supported by the current version of the TMCM-3215 CANopen firmware.

| Object Description | | | |
|--------------------|---------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 6098 _h | Homing method | Variable | SIGNED8 |

Table 265: Object Description (6098_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | SIGNED8 | 0 |

Table 266: Entry Description (6098_h)

8.2.6 Object 6099_h: Homing Speeds

This object indicates the configured speeds used during homing procedure. The values are given in pps units or internal units selectable with object 208C_h (section 4.2.17). Using object 6099_h a fast and a slow homing speed can be set. In most homing modes, the home switch is searched with the fast speed first. When the home switch has been found, the motor will be decelerated to the slow speed (using the homing acceleration, object 609A_h) to search for the exact switch point. When the switch point has been found the motor will be stopped at that point.

| Object Description | | | |
|--------------------|---------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 6099 _h | Homing speeds | Array | UNSIGNED32 |

Table 267: Object Description (6099_h)

| Entry Description | | | | | |
|-------------------|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Description | Access | PDO Mapping | Value Range | Default Value |
| 1 | Fast homing speed | rw | no | UNSIGNED32 | 0 |
| 2 | Slow homing speed | rw | no | UNSIGNED32 | 0 |

Table 268: Entry Description (6099_h)

8.2.7 Object 609A_h: Homing Acceleration

This object indicates the configured acceleration and deceleration to be used during homing operation. The value is given in units selected by object 208E_h (section 4.2.18).



| Object Description | | | |
|--------------------|---------------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 609A _h | Homing acceleration | Variable | UNSIGNED32 |

Table 269: Object Description (609A_h)

| Entry Description | | | | |
|-------------------|--------|-------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | no | UNSIGNED32 | 0 |

Table 270: Entry Description (609A_h)

8.3 How to start a Homing in hm Mode

Here is a little example that shows how to home the motor in hm mode. In this little example we assume that the module has been reset (and then switched to pre-operational or operational) by NMT commands before. The home switch must be connected to the home switch input. It can be operated manually.

- Select hm mode by writing 6 to object 6060_h.
- Write 6 to object 6040_h to switch to READY_TO_SWITCH_ON state.
- Write 7 to object 6040_h to switch to SWITCHED_ON state.
- Write 15 to object 6040_h to switch to OPERATION_ENABLED state.
- Select homing method 19 by writing 19 to object 6098_h.
- Set the homing speeds by writing e.g. 50000 to object 6099_h sub index 1 and e.g. 10000 to object 6099_h sub index 2.
- Write 31 to object 6040_h to start the homing process.
- Press and release the home switch.
- When homing has finished, write 15 to object 6040_h again.



9 Cyclic synchronous Position Mode

The cyclic synchronous position mode is used to directly control the position of the motor. It contains limit functions, but not a trajectory generator. The trajectory generator is located in the control device (the master), not in the drive device. In cyclic synchronous manner, the control device provides a target position to the drive device, which performs position control, velocity control and torque control.

The main control parameters are the target position (object 607A_h, see section 9.1.7) and the interpolation time period (object 60C2_h, see section 9.1.10). The drive automatically sets the velocity in such a manner that the next target position is reached within the interpolation time period. Acceleration and deceleration ramps are not used in this mode.

The cyclic synchronous position mode covers the following sub-functions:

- Position demand value input directly via an object.
- Monitoring of the position.
- Limiting the position using the software limits or the hardware limit switches.

9.1 Detailed Object Specifications

9.1.1 Object 6040_h: Control Word

This object indicates the received command controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to figure 8 for detailed information. The cyclic synchronous position mode does not use any mode specific bits of the control word.

| Structure of the Control Word | | | | | | | | | |
|-------------------------------|---|----|----|----|-----|----|----|---|---|
| 15 | 9 | 8 | 7 | 6 | 4 | 3 | 2 | 1 | 0 |
| nu | h | fr | nu | eo | qs | ev | so | | |
| MSB | | | | | LSB | | | | |

Legend: nu=not used; h=halt; fr=fault reset; eo=enable operation; qs=quick stop; ev=enable voltage; so=switch on.

Table 271: Structure of the Control Word in csp Mode



| Command Coding | | | | | | |
|------------------------------|----------------------|-------|-------|-------|-------|-------------|
| Command | Bits of Control Word | | | | | Transitions |
| | Bit 7 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
| Shutdown | 0 | x | 1 | 1 | 0 | 2,6,8 |
| Switch on | 0 | 0 | 1 | 1 | 1 | 3 |
| Switch on & enable operation | 0 | 1 | 1 | 1 | 1 | 3, 4 |
| Disable voltage | 0 | x | x | 0 | x | 7,9,10,12 |
| Quick stop | 0 | x | 0 | 1 | x | 7,10,11 |
| Disable operation | 0 | 0 | 1 | 1 | 1 | 5 |
| Enable operation | 0 | 1 | 1 | 1 | 1 | 4, 16 |
| Fault reset | 0-to-1 | x | x | x | x | 15 |

Table 272: Command Coding

| Object Description | | | |
|--------------------|-------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 6040 _h | Controlword | Variable | UNSIGNED16 |

Table 273: Object Description (6040_h in csp Mode)

| Entry Description | | | | |
|-------------------|--------|--------------|---------------------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | see CiA402-3 | See command coding above. | |

Table 274: Entry Description (6040_h in csp Mode)

9.1.2 Object 6041_h: Status Word

This object provides the status of the PDS FSA. It reflects the status of the CiA-402 state machine. Please refer to figure 8 for detailed information. The object is structured as defined below. For more information about the coding please refer to the CANopen Drives and motion control device profile, part 2.

| Structure of the Status Word | | | | | | | | | | | | | | | |
|------------------------------|-----|-----|-----|----|----|----|---|-----|----|----|---|----|----|------|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| dir | mot | oms | ila | r | rm | ms | w | sod | qs | ve | f | oe | so | rtso | |
| MSB | | | | | | | | | | | | | | LSB | |

Legend: nu=not used; r=reserved; oms=operation mode specific; h=halt; fr=fault reset; oe=operation enable; qs=quick stop; ve=voltage enable; so=switch on.

Table 275: Structure of the Status Word in csp Mode



| Trinamic Specific Bits | | |
|------------------------|-----------------------|---|
| Bit | Name | Definition |
| 14 | Motor activity | 0: Motor stands still. 1: Motor rotates. |
| 15 | Direction of rotation | This bit shows the direction of rotation. |

Table 276: Trinamic Specific Bits

| Operation Mode specific Bits in csp Mode | | |
|--|-------------------------|---|
| Bit | Name | Definition |
| 10 | Reserved | Not used. |
| 12 | Target position ignored | 0: Target position ignored. 1: Target position used as input to position controller. |
| 13 | Following error | 0: No following error. 1: Following error. |

Table 277: Operation Mode specific Bits in csp Mode

| State Coding | |
|----------------------------------|------------------------|
| Status word | FSA state |
| xxxx xxxx x0xx 0000 _h | Not ready to switch on |
| xxxx xxxx x1xx 0000 _h | Switch on disabled |
| xxxx xxxx x01x 0001 _h | Ready to switch on |
| xxxx xxxx x01x 0011 _h | Switched on |
| xxxx xxxx x01x 0111 _h | Operation enabled |
| xxxx xxxx x00x 0111 _h | Quick stop active |
| xxxx xxxx x0xx 1111 _h | Fault reaction active |
| xxxx xxxx x0xx 1000 _h | Fault |

Table 278: State Coding

| Object Description | | | |
|--------------------|-------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 6041 _h | Controlword | Variable | UNSIGNED16 |

Table 279: Object Description (6041_h in csp Mode)



| Entry Description | | | | |
|-------------------|--------|--------------|------------------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | see CiA402-3 | See state coding above | |

Table 280: Entry Description (6041_h in csp Mode)

9.1.3 Object 6062_h: Position Demand Value

This object provides the demanded position value. The value is given in microsteps. Object 6062_h indicates the actual position that the motor should have. It is not to be confused with objects 6063_h and 6064_h.

| Object Description | | | |
|--------------------|-----------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 6062 _h | Position Demand Value | Variable | SIGNED32 |

Table 281: Object Description (6062_h)

| Entry Description | | | | |
|-------------------|--------|-------------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | ro | Refer to CiA402-3 | SIGNED32 | no |

Table 282: Entry Description (6062_h)

9.1.4 Object 6063_h: Position Actual Internal Value

This object provides the actual value of the encoder or the motor. Please use the sensor selection object 606A_h (see section 5.1.8) for selecting the motor or the encoder first. Object 6063_h indicates the actual position of the encoder or the motor, re-scaled to the microstep resolution. The value is given in microsteps.

| Object Description | | | |
|--------------------|--------------------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 6063 _h | Position Actual Internal Value | Variable | SIGNED32 |

Table 283: Object Description (6063_h)

| Entry Description | | | | |
|-------------------|--------|-------------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | ro | Refer to CiA402-3 | SIGNED32 | no |

Table 284: Entry Description (6063_h)



9.1.5 Object 6064_h: Position Actual Value

This object provides the actual value of the position measurement device. It always contains the same value as object 6063_h.

| Object Description | | | |
|--------------------|-----------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 6064 _h | Position Actual Value | Variable | SIGNED32 |

Table 285: Object Description (6064_h)

| Entry Description | | | | |
|-------------------|--------|-------------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | ro | Refer to CiA402-3 | SIGNED32 | no |

Table 286: Entry Description (6064_h)

9.1.6 Object 606C_h: Velocity Actual Value

This object shows the actual velocity value of the motor. The value is given in internal or user-defined velocity units (depending on object 208C_h, described in section 4.2.17).

| Object Description | | | |
|--------------------|-----------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 606C _h | Velocity Actual Value | Variable | SIGNED32 |

Table 287: Object Description (606C_h)

| Entry Description | | | | |
|-------------------|--------|-------------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | ro | Refer to CiA402-3 | SIGNED32 | no |

Table 288: Entry Description (606C_h)

9.1.7 Object 607A_h: Target Position

The target position is the position that the drive should move to in cyclic synchronous position mode using the current interpolation time period. In csp mode this value is always interpreted as an absolute value.

| Object Description | | | |
|--------------------|-----------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 607A _h | Target Position | Variable | SIGNED32 |

Table 289: Object Description (607A_h in csp Mode)



| Entry Description | | | | |
|-------------------|--------|-------------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | Refer to CiA402-3 | SIGNED32 | 0 |

Table 290: Entry Description (607A_h in csp Mode)

9.1.8 Object 607D_h: Software Position Limit

This object indicates the configured maximal and minimal software position limits. These parameters define the absolute position limits for the position demand value and the position actual value. Every new target position is checked against these limits. The limit positions are always relative to the machine home position. Before being compared with the target position, they are corrected internally by the home offset as follows:

$$\begin{aligned} \text{Corrected_min_position_limit} &= \text{min_position_limit} - \text{home_offset} \\ \text{Corrected_max_position_limit} &= \text{max_position_limit} - \text{home_offset} \end{aligned}$$

| Object Description | | | |
|--------------------|-------------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 607D _h | Software Position Limit | Array | SIGNED32 |

Table 291: Object Description (607D_h)

| Entry Description | | | | | |
|-------------------|------------------------|--------|-------------|-------------|---------------|
| Sub-index | Description | Access | PDO Mapping | Value Range | Default Value |
| 1 | Minimum Position Limit | rw | no | SIGNED32 | -2147483648 |
| 2 | Maximum Position Limit | rw | no | SIGNED32 | 2147483647 |

Table 292: Entry Description (607D_h)

9.1.9 Object 60B0_h: Position Offset

This object provides an offset to the target position (object 607A_h, see section 9.1.7)). The value is given in microsteps and will be added to the target position.

| Object Description | | | |
|--------------------|---------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 60B0 _h | Offset Torque | Variable | SIGNED32 |

Table 293: Object Description (60B0_h)



| Entry Description | | | | |
|-------------------|--------|-------------|--------------------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | yes | -2147483648...2147483647 | 0 |

Table 294: Entry Description (60B0_h)

9.1.10 Object 60C2_h: Interpolation Time Period

This object indicates the interpolation cycle time. The interpolation time period (sub-index 01_h) is given in $10^{\text{interpolation_time_index}}$ s. The interpolation time index (sub-index 02_h) is dimensionless.

| Object Description | | | |
|--------------------|---------------|-------------|---|
| Index | Name | Object Type | Data Type |
| 60C2 _h | Offset Torque | Vecord | Interpolation time period record (0080 _h) |

Table 295: Object Description (60C2_h)

| Entry Description | | | | | |
|-------------------|---------------------------------|--------|-------------|-------------|---------------|
| Sub-index | Description | Access | PDO Mapping | Value Range | Default Value |
| 0 | Highest sub-index supported | ro | no | UNSIGNED8 | 2 |
| 1 | Interpolation time period value | rw | no | UNSIGNED8 | 1 |
| 2 | Interpolation time index | rw | no | -3...3 | -3 |

Table 296: Entry Description (60C2_h)

10 Cyclic synchronous Velocity Mode

The cyclic synchronous velocity mode is used to directly control the velocity of the motor. It contains limit functions, but not a trajectory generator. The trajectory generator is located in the control device (the master), not in the drive device. In cyclic synchronous manner, the control device provides a target velocity to the drive device, which performs position control, velocity control and torque control.

The main control parameters are the target velocity (object 60FF_h, see section 10.1.4) and the interpolation time period (object 60C2_h, see section 10.1.7). The drive automatically sets the acceleration in such a manner that the next target velocity is reached within the interpolation time period. Acceleration and deceleration ramps are not used in this mode.

The cyclic synchronous velocity mode covers the following sub-functions:

- Velocity demand value input directly via an object.
- Monitoring of the position.
- Limiting the position using the software limits or the hardware limit switches.

10.1 Detailed Object Specifications

10.1.1 Object 6040_h: Control Word

This object indicates the received command controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to figure 8 for detailed information. The cyclic synchronous velocity mode does not use any mode specific bits of the control word.

| Structure of the Control Word | | | | | | | | | |
|-------------------------------|---|----|----|----|-----|----|----|---|---|
| 15 | 9 | 8 | 7 | 6 | 4 | 3 | 2 | 1 | 0 |
| nu | h | fr | nu | eo | qs | ev | so | | |
| MSB | | | | | LSB | | | | |

Legend: nu=not used; h=halt; fr=fault reset; eo=enable operation; qs=quick stop; ev=enable voltage; so=switch on.

Table 297: Structure of the Control Word in csv Mode



| Command Coding | | | | | | |
|------------------------------|----------------------|-------|-------|-------|-------|-------------|
| Command | Bits of Control Word | | | | | Transitions |
| | Bit 7 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
| Shutdown | 0 | x | 1 | 1 | 0 | 2,6,8 |
| Switch on | 0 | 0 | 1 | 1 | 1 | 3 |
| Switch on & enable operation | 0 | 1 | 1 | 1 | 1 | 3, 4 |
| Disable voltage | 0 | x | x | 0 | x | 7,9,10,12 |
| Quick stop | 0 | x | 0 | 1 | x | 7,10,11 |
| Disable operation | 0 | 0 | 1 | 1 | 1 | 5 |
| Enable operation | 0 | 1 | 1 | 1 | 1 | 4, 16 |
| Fault reset | 0-to-1 | x | x | x | x | 15 |

Table 298: Command Coding

| Object Description | | | |
|--------------------|-------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 6040 _h | Controlword | Variable | UNSIGNED16 |

Table 299: Object Description (6040_h in csv Mode)

| Entry Description | | | | |
|-------------------|--------|--------------|---------------------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | see CiA402-3 | See command coding above. | |

Table 300: Entry Description (6040_h in csv Mode)

10.1.2 Object 6041_h: Status Word

This object provides the status of the PDS FSA. It reflects the status of the CiA-402 state machine. Please refer to figure 8 for detailed information. The object is structured as defined below. For more information about the coding please refer to the CANopen Drives and motion control device profile, part 2.

| Structure of the Status Word | | | | | | | | | | | | | | | |
|------------------------------|-----|-----|-----|----|----|----|---|-----|----|----|---|----|-----|------|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| dir | mot | oms | ila | r | rm | ms | w | sod | qs | ve | f | oe | so | rtso | |
| MSB | | | | | | | | | | | | | LSB | | |

Legend: nu=not used; r=reserved; oms=operation mode specific; h=halt; fr=fault reset; oe=operation enable; qs=quick stop; ve=voltage enable; so=switch on.

Table 301: Structure of the Status Word in csv Mode



| Trinamic Specific Bits | | |
|------------------------|-----------------------|---|
| Bit | Name | Definition |
| 14 | Motor activity | 0: Motor stands still. 1: Motor rotates. |
| 15 | Direction of rotation | This bit shows the direction of rotation. |

Table 302: Trinamic Specific Bits

| Operation Mode specific Bits in csv Mode | | |
|--|-------------------------|---|
| Bit | Name | Definition |
| 10 | Reserved | Not used. |
| 12 | Target position ignored | 0: Target velocity ignored. 1: Target velocity used as input to velocity controller. |
| 13 | Reserved | Not used. |

Table 303: Operation Mode specific Bits in csv Mode

| State Coding | |
|----------------------------------|------------------------|
| Status word | FSA state |
| xxxx xxxx x0xx 0000 _h | Not ready to switch on |
| xxxx xxxx x1xx 0000 _h | Switch on disabled |
| xxxx xxxx x01x 0001 _h | Ready to switch on |
| xxxx xxxx x01x 0011 _h | Switched on |
| xxxx xxxx x01x 0111 _h | Operation enabled |
| xxxx xxxx x00x 0111 _h | Quick stop active |
| xxxx xxxx x0xx 1111 _h | Fault reaction active |
| xxxx xxxx x0xx 1000 _h | Fault |

Table 304: State Coding

| Object Description | | | |
|--------------------|-------------|-------------|------------|
| Index | Name | Object Type | Data Type |
| 6041 _h | Controlword | Variable | UNSIGNED16 |

Table 305: Object Description (6041_h in csv Mode)



| Entry Description | | | | |
|-------------------|--------|--------------|------------------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | see CiA402-3 | See state coding above | |

Table 306: Entry Description (6041_h in csv Mode)

10.1.3 Object 606C_h: Velocity Actual Value

This object shows the actual velocity value of the motor. The value is given in internal or user-defined velocity units (depending on object 208C_h, described in section 4.2.17).

| Object Description | | | |
|--------------------|-----------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 606C _h | Velocity Actual Value | Variable | SIGNED32 |

Table 307: Object Description (606C_h)

| Entry Description | | | | |
|-------------------|--------|-------------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | ro | Refer to CiA402-3 | SIGNED32 | no |

Table 308: Entry Description (606C_h)

10.1.4 Object 60FF_h: Target Velocity

In csv mode the target velocity specifies the velocity that is to be reached within the interpolation time period. The values are given in units which can be selected with object 208C_h, described in section 4.2.17.

| Object Description | | | |
|--------------------|-----------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 60FF _h | Target Velocity | Variable | SIGNED32 |

Table 309: Object Description (60FF_h)

| Entry Description | | | | |
|-------------------|--------|--------------|-------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | see CiA402-3 | SIGNED32 | 0 |

Table 310: Entry Description (60FF_h)

10.1.5 Object 607D_h: Software Position Limit

This object indicates the configured maximal and minimal software position limits. These parameters define the absolute position limits for the position demand value and the position actual value. Every



new target position is checked against these limits. The limit positions are always relative to the machine home position. Before being compared with the target position, they are corrected internally by the home offset as follows:

$$\begin{aligned} \text{Corrected_min_position_limit} &= \text{min_position_limit} - \text{home_offset} \\ \text{Corrected_max_position_limit} &= \text{max_position_limit} - \text{home_offset} \end{aligned}$$

| Object Description | | | |
|--------------------|-------------------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 607D _h | Software Position Limit | Array | SIGNED32 |

Table 311: Object Description (607D_h)

| Entry Description | | | | | |
|-------------------|------------------------|--------|-------------|-------------|---------------|
| Sub-index | Description | Access | PDO Mapping | Value Range | Default Value |
| 1 | Minimum Position Limit | rw | no | SIGNED32 | -2147483648 |
| 2 | Maximum Position Limit | rw | no | SIGNED32 | 2147483647 |

Table 312: Entry Description (607D_h)

10.1.6 Object 60B1_h: Velocity Offset

This object provides an offset to the target velocity (object 60FF_h, see section 10.1.4)). The value will be added to the target velocity.

| Object Description | | | |
|--------------------|-----------------|-------------|-----------|
| Index | Name | Object Type | Data Type |
| 60B1 _h | Velocity Offset | Variable | SIGNED32 |

Table 313: Object Description (60B1_h)

| Entry Description | | | | |
|-------------------|--------|-------------|--------------------------|---------------|
| Sub-index | Access | PDO Mapping | Value Range | Default Value |
| 0 | rw | yes | -2147483648...2147483647 | 0 |

Table 314: Entry Description (60B1_h)

10.1.7 Object 60C2_h: Interpolation Time Period

This object indicates the interpolation cycle time. The interpolation time period (sub-index 01_h) is given in 10^{*interpolation_time_index*} s. The interpolation time index (sub-index 02_h) is dimensionless.



| Object Description | | | |
|--------------------|---------------|-------------|---|
| Index | Name | Object Type | Data Type |
| 60C2 _h | Offset Torque | Vecord | Interpolation time period record (0080 _h) |

Table 315: Object Description (60C2_h)

| Entry Description | | | | | |
|-------------------|---------------------------------|--------|-------------|-------------|---------------|
| Sub-index | Description | Access | PDO Mapping | Value Range | Default Value |
| 0 | Highest sub-index supported | ro | no | UNSIGNED8 | 2 |
| 1 | Interpolation time period value | rw | no | UNSIGNED8 | 1 |
| 2 | Interpolation time index | rw | no | -3...3 | -3 |

Table 316: Entry Description (60C2_h)



11 Emergency Messages (EMCY)

The module sends an emergency message if an error occurs. The message contains information about the error type. The module can map internal errors and object 1001_h (error register) is part of every emergency object.

Please note that the additional byte #2 shows which motor is affected.

| Emergency Messages (EMCY) of the TMCM-3215 | | | | | | |
|--|-----------------|-------|---|---|---|--|
| Error code | Additional byte | | | | | Description |
| | 1 | 2 | 3 | 4 | 5 | |
| 0000 _h | 0 | 0...2 | 0 | 0 | 0 | Fault reset The fault reset command has been executed. |
| 1000 _h | 1 | 0...2 | 0 | 0 | 0 | Generic error: open load bridge A The motor driver indicates open load on bridge A. It is possible that the motor cable is broken or that there is an error in the power amplifier itself. |
| 1000 _h | 2 | 0...2 | 0 | 0 | 0 | Generic error: open load bridge B The motor driver indicates open load on bridge B. It is possible that the motor cable is broken or that there is an error in the power amplifier itself. |
| 2310 _h | 0 | 0...2 | 0 | 0 | 0 | Overcurrent high side The motor driver indicates an overcurrent on the high side. This can be caused by a short circuit in the driver stage. |
| 2311 _h | 0 | 0...2 | 0 | 0 | 0 | Overcurrent bridge B The motor driver indicates that there is overcurrent on bridge B. This can be caused by a short circuit in the motor itself or in the motor driver stage. |
| 2312 _h | 0 | 0...2 | 0 | 0 | 0 | Overcurrent bridge A The motor driver indicates that there is overcurrent on bridge A. This can be caused by a short circuit in the motor itself or in the motor driver stage. |
| 3230 _h | 0 | 0...2 | 0 | 0 | 0 | stallGuard2 error The actual load value exceeds the stallGuard2 limit. |
| 4310 _h | 1 | 0...2 | 0 | 0 | 0 | Overtemperature pre-warning The temperature in the motor driver exceeds the pre-warning limit. |
| 4310 _h | 2 | 0...2 | 0 | 0 | 0 | Overtemperature error The motor driver has been switched off because the temperature limit has been exceeded. |
| 5441 _h | 0 | 255 | 0 | 0 | 0 | Shutdown switch active The enable signal is missing (due to the shutdown switch) and the motor driver has been switched off. |
| 6320 _h | 0 | 255 | 0 | 0 | 0 | Parameter error The data in the received PDO is either wrong or cannot be accepted due to the internal state of the drive. |



| Error code | Additional byte | | | | | Description |
|-------------------|-----------------|-------|---|---|---|--|
| | 1 | 2 | 3 | 4 | 5 | |
| 8611 _h | 0 | 0...2 | 0 | 0 | 0 | Following error The deviation between motor position counter and encoder position counter has exceeded the following error window. |
| ff00 _h | 0 | 0...2 | 0 | 0 | 0 | Undervoltage The supply voltage is too low to drive a motor. |
| ff01 _h | 1 | 0...2 | 0 | 0 | 0 | Positive software limit The actual position is outside the range defined by object 607d _h . |
| ff01 _h | 2 | 0...2 | 0 | 0 | 0 | Negative software limit The actual position is outside the range defined by object 607d _h . |
| ff01 _h | 3 | 0...2 | 0 | 0 | 0 | Positive limit switch The positive limit switch has been touched outside of the homing function. |
| ff01 _h | 4 | 0...2 | 0 | 0 | 0 | Negative limit switch The negative limit switch has been touched outside of the homing function. |

Table 317: Emergency Messages (EMCY)



12 Figures Index

| | | | | | |
|---|--|----|----|------------------------------------|----|
| 1 | stallGuard2 Load Measurement as a Function of Load | 8 | 9 | Homing Mode Function | 81 |
| 2 | Energy Efficiency Example with CoolStep | 9 | 10 | Homing Method 1 | 82 |
| 3 | Typical motion profile with TRINAMIC's SixPoint™ motion controller | 9 | 11 | Homing Method 2 | 83 |
| 4 | NMT State Machine | 13 | 12 | Homing Method 3 | 83 |
| 5 | Device Model | 14 | 13 | Homing Method 5 | 84 |
| 6 | CoolStep Adjustment Points and Thresholds | 25 | 14 | Homing Method 17 | 84 |
| 7 | Brake Output Timing | 28 | 15 | Homing Method 18 | 84 |
| 8 | DS402 Finite State Machine | 61 | 16 | Homing Method 19 | 85 |
| | | | 17 | Homing Method 21 | 85 |
| | | | 18 | Homing Methods 33 and 34 | 86 |
| | | | 19 | Home Offset | 90 |



13 Tables Index

| | | | | | |
|----|--|----|-----|---|----|
| 1 | Abbreviations used in this Manual . . . | 7 | 52 | Object Description (200F _h) | 31 |
| 2 | SixPoint Ramp Parameters | 10 | 53 | Entry Description (200F _h) | 32 |
| 3 | Service Primitives | 11 | 54 | Object Description (2010 _h) | 32 |
| 4 | Service Types | 12 | 55 | Entry Description (2010 _h) | 32 |
| 5 | Object Dictionary | 15 | 56 | Object Description (2011 _h) | 32 |
| 6 | Multi-Axis Object Indices | 16 | 57 | Entry Description (2011 _h) | 32 |
| 7 | Object Description (1000 _h) | 17 | 58 | Object Description (2012 _h) | 33 |
| 8 | Entry Description (1000 _h) | 17 | 59 | Entry Description (2012 _h) | 33 |
| 9 | Object Description (1001 _h) | 17 | 60 | Object Description (2013 _h) | 33 |
| 10 | Entry Description (1001 _h) | 17 | 61 | Entry Description (2013 _h) | 33 |
| 11 | Error Register Bits | 18 | 62 | Object Description (2015 _h) | 34 |
| 12 | Object Description (1008 _h) | 18 | 63 | Entry Description (2015 _h) | 34 |
| 13 | Entry Description (1008 _h) | 18 | 64 | Object Description (2089 _h) | 34 |
| 14 | Object Description (1009 _h) | 18 | 65 | Entry Description (2089 _h) | 34 |
| 15 | Entry Description (1009 _h) | 19 | 66 | Object Description (208C _h) | 35 |
| 16 | Object Description (100A _h) | 19 | 67 | Entry Description (208C _h) | 35 |
| 17 | Entry Description (100A _h) | 19 | 68 | Object Description (208E _h) | 35 |
| 18 | Object Description (1018 _h) | 19 | 69 | Entry Description (208E _h) | 35 |
| 19 | Entry Description (1018 _h) | 20 | 70 | Object Description (2092 _h) | 35 |
| 20 | Object Description (1600 _h) | 20 | 71 | Entry Description (2092 _h) | 36 |
| 21 | Entry Description (1600 _h) | 20 | 72 | Object Description (2093 _h) | 36 |
| 22 | Object Description (1A00 _h) | 21 | 73 | Entry Description (2093 _h) | 36 |
| 23 | Entry Description (1A00 _h) | 21 | 74 | Object Description (2094 _h) | 36 |
| 24 | Sync Manager Communication Types | 22 | 75 | Entry Description (2094 _h) | 36 |
| 25 | Object Description (1C00 _h) | 22 | 76 | Object Description (2095 _h) | 37 |
| 26 | Entry Description (1C00 _h) | 22 | 77 | Entry Description (2095 _h) | 37 |
| 27 | Object Description (1C12 _h) | 22 | 78 | Object Description (2096 _h) | 37 |
| 28 | Entry Description (1C12 _h) | 23 | 79 | Entry Description (2096 _h) | 37 |
| 29 | Object Description (1C13 _h) | 23 | 80 | Object Description (2097 _h) | 38 |
| 30 | Entry Description (1C13 _h) | 23 | 81 | Entry Description (2097 _h) | 38 |
| 31 | Multi-axis Object Indices (Manufacturer specific Area) | 24 | 82 | Object Description (2098 _h) | 38 |
| 32 | CoolStep related Objects | 26 | 83 | Entry Description (2098 _h) | 38 |
| 33 | Object Description (2000 _h) | 27 | 84 | Object Description (2099 _h) | 39 |
| 34 | Entry Description (2000 _h) | 27 | 85 | Entry Description (2099 _h) | 39 |
| 35 | Object Description (2001 _h) | 27 | 86 | Object Description (209A _h) | 39 |
| 36 | Entry Description (2001 _h) | 27 | 87 | Entry Description (209A _h) | 39 |
| 37 | Object Description (2002 _h) | 28 | 88 | Object Description (209B _h) | 40 |
| 38 | Entry Description (2002 _h) | 28 | 89 | Entry Description (209B _h) | 40 |
| 39 | Object Description (2003 _h) | 28 | 90 | Object Description (209C _h) | 40 |
| 40 | Entry Description (2003 _h) | 28 | 91 | Entry Description (209C _h) | 40 |
| 41 | Object Description (2004 _h) | 29 | 92 | Object Description (209D _h) | 41 |
| 42 | Entry Description (2004 _h) | 29 | 93 | Entry Description (209D _h) | 41 |
| 43 | Object Description (2005 _h) | 29 | 94 | Object Description (209E _h) | 41 |
| 44 | Entry Description (2005 _h) | 29 | 95 | Entry Description (209E _h) | 41 |
| 45 | Bit Definitions (2005 _h) | 30 | 96 | Object Description (20A1 _h) | 41 |
| 46 | Object Description (200A _h) | 30 | 97 | Entry Description (20A1 _h) | 42 |
| 47 | Entry Description (200A _h) | 30 | 98 | Object Description (20A3 _h) | 42 |
| 48 | Object Description (200B _h) | 30 | 99 | Entry Description (20A3 _h) | 42 |
| 49 | Entry Description (200B _h) | 31 | 100 | Object Description (20A4 _h) | 42 |
| 50 | Object Description (200C _h) | 31 | 101 | Entry Description (20A4 _h) | 42 |
| 51 | Entry Description (200C _h) | 31 | 102 | Object Description (20A5 _h) | 43 |
| | | | 103 | Entry Description (20A5 _h) | 43 |



| | | | | | |
|-----|---|----|-----|---|----|
| 104 | Object Description (20B0 _h) | 43 | 157 | Value Description (605D _h) | 55 |
| 105 | Entry Description (20B0 _h) | 43 | 158 | Object Description (605D _h) | 56 |
| 106 | Object Description (20B1 _h) | 44 | 159 | Entry Description (605D _h) | 56 |
| 107 | Entry Description (20B1 _h) | 44 | 160 | Value Description (605E _h) | 56 |
| 108 | Object Description (20B2 _h) | 44 | 161 | Object Description (605E _h) | 56 |
| 109 | Entry Description (20B2 _h) | 44 | 162 | Entry Description (605E _h) | 56 |
| 110 | Object Description (20B3 _h) | 44 | 163 | Value Description (6060 _h) | 57 |
| 111 | Entry Description (20B3 _h) | 45 | 164 | Object Description (6060 _h) | 57 |
| 112 | Object Description (20B4 _h) | 45 | 165 | Entry Description (6060 _h) | 57 |
| 113 | Entry Description (20B4 _h) | 45 | 166 | Value Description (6061 _h) | 57 |
| 114 | Object Description (20B5 _h) | 45 | 167 | Object Description (6061 _h) | 58 |
| 115 | Entry Description (20B5 _h) | 45 | 168 | Entry Description (6061 _h) | 58 |
| 116 | Object Description (20B6 _h) | 46 | 169 | Value Description (606A _h) | 58 |
| 117 | Entry Description (20B6 _h) | 46 | 170 | Object Description (606A _h) | 58 |
| 118 | Object Description (20B7 _h) | 46 | 171 | Entry Description (606A _h) | 58 |
| 119 | Entry Description (20B7 _h) | 46 | 172 | Object Description (608F _h) | 59 |
| 120 | Object Description (20B8 _h) | 46 | 173 | Entry Description (608F _h) | 59 |
| 121 | Entry Description (20B8 _h) | 47 | 174 | Object Description (60FD _h) | 59 |
| 122 | Object Description (20B9 _h) | 47 | 175 | Entry Description (60FD _h) | 59 |
| 123 | Entry Description (20B9 _h) | 47 | 176 | Value Definition (6502 _h) | 60 |
| 124 | Object Description (2100 _h) | 47 | 177 | Object Description (6502 _h) | 60 |
| 125 | Entry Description (2100 _h) | 47 | 178 | Entry Description (6502 _h) | 60 |
| 126 | Object Description (2101 _h) | 48 | 179 | Structure of the Control Word in pp Mode | 62 |
| 127 | Entry Description (2101 _h) | 48 | 180 | Operation Mode specific Bits in pp Mode | 62 |
| 128 | Driver Error Flags (2102 _h) | 48 | 181 | Command Coding | 62 |
| 129 | Object Description (2102 _h) | 48 | 182 | Object Description (6040 _h in pp Mode) | 63 |
| 130 | Entry Description (2102 _h) | 49 | 183 | Entry Description (6040 _h in pp Mode) | 63 |
| 131 | Object Description (2107 _h) | 49 | 184 | Structure of the Staus Word in pp Mode | 63 |
| 132 | Entry Description (2107 _h) | 49 | 185 | Trinamic Specific Bits | 63 |
| 133 | Object Description (210B _h) | 49 | 186 | Operation Mode specific Bits in pp Mode | 64 |
| 134 | Entry Description (210B _h) | 49 | 187 | State Coding | 64 |
| 135 | Object Description (2121 _h) | 50 | 188 | Object Description (6041 _h in pp Mode) | 64 |
| 136 | Entry Description (2121 _h) | 50 | 189 | Entry Description (6041 _h in pp Mode) | 64 |
| 137 | Object Description (2122 _h) | 50 | 190 | Object Description (6062 _h) | 65 |
| 138 | Entry Description (2122 _h) | 50 | 191 | Entry Description (6062 _h) | 65 |
| 139 | Bit Definitions (2702 _h) | 51 | 192 | Object Description (6063 _h) | 65 |
| 140 | Object Description (2702 _h) | 51 | 193 | Entry Description (6063 _h) | 65 |
| 141 | Entry Description (2702 _h) | 51 | 194 | Object Description (6064 _h) | 65 |
| 142 | Bit Definitions (2703 _h) | 51 | 195 | Entry Description (6064 _h) | 66 |
| 143 | Object Description (2703 _h) | 52 | 196 | Object Description (6065 _h) | 66 |
| 144 | Entry Description (2703 _h) | 52 | 197 | Entry Description (6065 _h) | 66 |
| 145 | Object Description (270E _h) | 52 | 198 | Object Description (6067 _h) | 67 |
| 146 | Entry Description (270E _h) | 52 | 199 | Entry Description (6067 _h) | 67 |
| 147 | Multi-axis Object Indices (Profile specific Area) | 53 | 200 | Object Description (6068 _h) | 67 |
| 148 | Value Description (605A _h) | 54 | 201 | Entry Description (6068 _h) | 67 |
| 149 | Object Description (605A _h) | 54 | 202 | Object Description (606C _h) | 67 |
| 150 | Entry Description (605A _h) | 54 | 203 | Entry Description (606C _h) | 68 |
| 151 | Value Description (605B _h) | 54 | 204 | Object Description (607A _h in pp Mode) | 68 |
| 152 | Object Description (605B _h) | 54 | 205 | Entry Description (607A _h in pp Mode) | 68 |
| 153 | Entry Description (605B _h) | 55 | 206 | Object Description (607D _h) | 68 |
| 154 | Value Description (605C _h) | 55 | 207 | Entry Description (607D _h) | 69 |
| 155 | Object Description (605C _h) | 55 | | | |
| 156 | Entry Description (605C _h) | 55 | | | |



| | | | | | |
|-----|---|----|-----|---|-----|
| 208 | Object Description (6081 _h) | 69 | 257 | Operation Mode specific Bits in hm Mode | 89 |
| 209 | Entry Description (6081 _h) | 69 | 258 | State Coding | 89 |
| 210 | Object Description (6082 _h) | 69 | 259 | Object Description (6041 _h in hm Mode) | 89 |
| 211 | Entry Description (6082 _h) | 69 | 260 | Entry Description (6041 _h in hm Mode) | 89 |
| 212 | Object Description (6083 _h) | 70 | 261 | Object Description (606C _h) | 90 |
| 213 | Entry Description (6083 _h) | 70 | 262 | Entry Description (606C _h) | 90 |
| 214 | Object Description (6084 _h) | 70 | 263 | Object Description (607C _h) | 90 |
| 215 | Entry Description (6084 _h) | 70 | 264 | Entry Description (607C _h) | 90 |
| 216 | Object Description (6085 _h) | 71 | 265 | Object Description (6098 _h) | 91 |
| 217 | Entry Description (6085 _h) | 71 | 266 | Entry Description (6098 _h) | 91 |
| 218 | Bit Definitions of Object 60F2 _h | 71 | 267 | Object Description (6099 _h) | 91 |
| 219 | Object Description (60F2 _h) | 71 | 268 | Entry Description (6099 _h) | 91 |
| 220 | Entry Description (60F2 _h) | 71 | 269 | Object Description (609A _h) | 92 |
| 221 | Structure of the Control Word in pv Mode | 73 | 270 | Entry Description (609A _h) | 92 |
| 222 | Command Coding | 74 | 271 | Structure of the Control Word in csp Mode | 93 |
| 223 | Object Description (6040 _h in pv Mode) | 74 | 272 | Command Coding | 94 |
| 224 | Entry Description (6040 _h in pv Mode) | 74 | 273 | Object Description (6040 _h in csp Mode) | 94 |
| 225 | Structure of the Status Word in pv Mode | 74 | 274 | Entry Description (6040 _h in csp Mode) | 94 |
| 226 | Trinamic Specific Bits | 75 | 275 | Structure of the Status Word in csp Mode | 94 |
| 227 | Operation Mode specific Bits in pv Mode | 75 | 276 | Trinamic Specific Bits | 95 |
| 228 | State Coding | 75 | 277 | Operation Mode specific Bits in csp Mode | 95 |
| 229 | Object Description (6041 _h in pv Mode) | 75 | 278 | State Coding | 95 |
| 230 | Entry Description (6041 _h in pv Mode) | 76 | 279 | Object Description (6041 _h in csp Mode) | 95 |
| 231 | Object Description (6062 _h) | 76 | 280 | Entry Description (6041 _h in csp Mode) | 96 |
| 232 | Entry Description (6062 _h) | 76 | 281 | Object Description (6062 _h) | 96 |
| 233 | Object Description (6063 _h) | 76 | 282 | Entry Description (6062 _h) | 96 |
| 234 | Entry Description (6063 _h) | 76 | 283 | Object Description (6063 _h) | 96 |
| 235 | Object Description (6064 _h) | 77 | 284 | Entry Description (6063 _h) | 96 |
| 236 | Entry Description (6064 _h) | 77 | 285 | Object Description (6064 _h) | 97 |
| 237 | Object Description (6065 _h) | 77 | 286 | Entry Description (6064 _h) | 97 |
| 238 | Entry Description (6065 _h) | 77 | 287 | Object Description (606C _h) | 97 |
| 239 | Object Description (606C _h) | 78 | 288 | Entry Description (606C _h) | 97 |
| 240 | Entry Description (606C _h) | 78 | 289 | Object Description (607A _h in csp Mode) | 97 |
| 241 | Object Description (607D _h) | 78 | 290 | Entry Description (607A _h in csp Mode) | 98 |
| 242 | Entry Description (607D _h) | 78 | 291 | Object Description (607D _h) | 98 |
| 243 | Object Description (6083 _h) | 79 | 292 | Entry Description (607D _h) | 98 |
| 244 | Entry Description (6083 _h) | 79 | 293 | Object Description (60B0 _h) | 98 |
| 245 | Object Description (6085 _h) | 79 | 294 | Entry Description (60B0 _h) | 99 |
| 246 | Entry Description (6085 _h) | 79 | 295 | Object Description (60C2 _h) | 99 |
| 247 | Object Description (60FF _h) | 80 | 296 | Entry Description (60C2 _h) | 99 |
| 248 | Entry Description (60FF _h) | 80 | 297 | Structure of the Control Word in csv Mode | 100 |
| 249 | Supported CANopen Homing Methods | 82 | 298 | Command Coding | 101 |
| 250 | Structure of the Control Word in hm Mode | 87 | 299 | Object Description (6040 _h in csv Mode) | 101 |
| 251 | Operation Mode specific Bits in hm Mode | 87 | 300 | Entry Description (6040 _h in csv Mode) | 101 |
| 252 | Command Coding | 87 | 301 | Structure of the Status Word in csv Mode | 101 |
| 253 | Object Description (6040 _h in hm Mode) | 88 | 302 | Trinamic Specific Bits | 102 |
| 254 | Entry Description (6040 _h in hm Mode) | 88 | 303 | Operation Mode specific Bits in csv Mode | 102 |
| 255 | Structure of the Status Word in hm Mode | 88 | | | |
| 256 | Trinamic Specific Bits | 88 | | | |



| | | | | | |
|-----|--|-----|-----|---|-----|
| 304 | State Coding | 102 | 312 | Entry Description (607D _h) | 104 |
| 305 | Object Description (6041 _h in csv Mode) | 102 | 313 | Object Description (60B1 _h) | 104 |
| 306 | Entry Description (6041 _h in csv Mode) | 103 | 314 | Entry Description (60B1 _h) | 104 |
| 307 | Object Description (606C _h) | 103 | 315 | Object Description (60C2 _h) | 105 |
| 308 | Entry Description (606C _h) | 103 | 316 | Entry Description (60C2 _h) | 105 |
| 309 | Object Description (60FF _h) | 103 | 317 | Emergency Messages (EMCY) | 107 |
| 310 | Entry Description (60FF _h) | 103 | 318 | Firmware Revision | 115 |
| 311 | Object Description (607D _h) | 104 | 319 | Document Revision | 115 |



14 Supplemental Directives

14.1 Producer Information

14.2 Copyright

TRINAMIC owns the content of this user manual in its entirety, including but not limited to pictures, logos, trademarks, and resources. © Copyright 2021 TRINAMIC. All rights reserved. Electronically published by TRINAMIC, Germany.

Redistributions of source or derived format (for example, Portable Document Format or Hypertext Markup Language) must retain the above copyright notice, and the complete Datasheet User Manual documentation of this product including associated Application Notes; and a reference to other available product-related documentation.

14.3 Trademark Designations and Symbols

Trademark designations and symbols used in this documentation indicate that a product or feature is owned and registered as trademark and/or patent either by TRINAMIC or by other manufacturers, whose products are used or referred to in combination with TRINAMIC's products and TRINAMIC's product documentation.

This CoE Firmware Manual is a non-commercial publication that seeks to provide concise scientific and technical user information to the target user. Thus, trademark designations and symbols are only entered in the Short Spec of this document that introduces the product at a quick glance. The trademark designation /symbol is also entered when the product or feature name occurs for the first time in the document. All trademarks and brand names used are property of their respective owners.

14.4 Target User

The documentation provided here, is for programmers and engineers only, who are equipped with the necessary skills and have been trained to work with this type of product.

The Target User knows how to responsibly make use of this product without causing harm to himself or others, and without causing damage to systems or devices, in which the user incorporates the product.

14.5 Disclaimer: Life Support Systems

TRINAMIC Motion Control GmbH & Co. KG does not authorize or warrant any of its products for use in life support systems, without the specific written consent of TRINAMIC Motion Control GmbH & Co. KG.

Life support systems are equipment intended to support or sustain life, and whose failure to perform, when properly used in accordance with instructions provided, can be reasonably expected to result in personal injury or death.

Information given in this document is believed to be accurate and reliable. However, no responsibility is assumed for the consequences of its use nor for any infringement of patents or other rights of third parties which may result from its use. Specifications are subject to change without notice.

14.6 Disclaimer: Intended Use

The data specified in this user manual is intended solely for the purpose of product description. No representations or warranties, either express or implied, of merchantability, fitness for a particular purpose



or of any other nature are made hereunder with respect to information/specification or the products to which information refers and no guarantee with respect to compliance to the intended use is given.

In particular, this also applies to the stated possible applications or areas of applications of the product. TRINAMIC products are not designed for and must not be used in connection with any applications where the failure of such products would reasonably be expected to result in significant personal injury or death (safety-Critical Applications) without TRINAMIC's specific written consent.

TRINAMIC products are not designed nor intended for use in military or aerospace applications or environments or in automotive applications unless specifically designated for such use by TRINAMIC. TRINAMIC conveys no patent, copyright, mask work right or other trade mark right to this product. TRINAMIC assumes no liability for any patent and/or other trade mark rights of a third party resulting from processing or handling of the product and/or any other use of the product.

14.7 Collateral Documents & Tools

This product documentation is related and/or associated with additional tool kits, firmware and other items, as provided on the product page at: www.trinamic.com.



15 Revision History

15.1 Firmware Revision

| Version | Date | Author | Description |
|--------------|-------------|--------|---|
| 1.01 | 2016-JUL-20 | OK | First release. |
| 1.02... 1.06 | | OK | Not deployed. |
| 1.07 | 2018-JAN-12 | OK | CSP mode included. |
| 1.08 | 2019-JAN-11 | OK | CSV mode included. Distributed Clocks fixed. |
| 1.09 | 2019-MAY-09 | OK | Fully compliant with latest conformance test. |

Table 318: Firmware Revision

15.2 Document Revision

| Version | Date | Author | Description |
|---------|-------------|--------|--|
| V1.01 | 2016-JUL-20 | OK | First release. |
| V1.02 | 2016-NOV-29 | OK | Block diagrams included. |
| V1.03 | 2017-JAN-18 | OK | CSP mode included. |
| V1.04 | 2017-FEB-27 | OK | Mapping tables (objects 1600 _h and 1A00 _h) fixed. |
| V1.05 | 2018-JAN-12 | OK | Firmware V1.07 included. |
| V1.06 | 2018-AUG-02 | OK | Firmware V1.08 included. |
| V1.07 | 2019-MAY-09 | OK | Firmware V1.09 included. |
| V1.08 | 2020-FEB-06 | OK | Description of USB interface revised. |

Table 319: Document Revision

