

CANopen Firmware Version V3.16

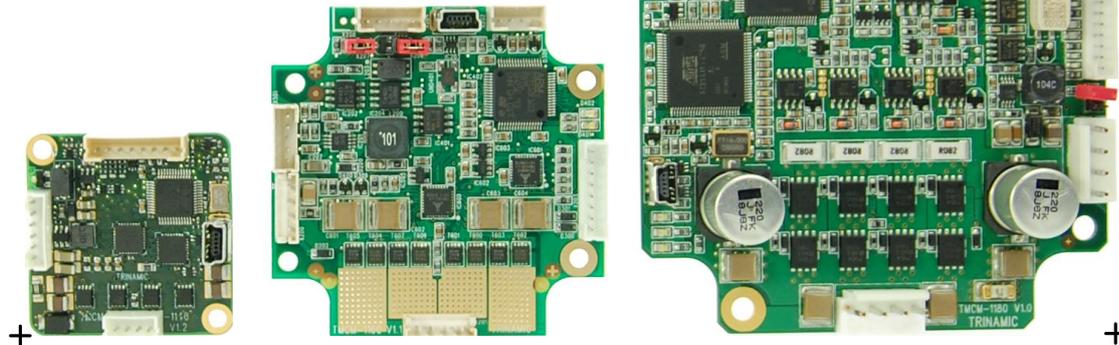
# CANopen MANUAL

## TMCM-1140 TMCM-1160 TMCM-1180

1-Axis Controller/Driver Modules  
CANopen  
Motor Mountable (NEMA17, NEMA23/24, NEMA34)  
Integrated sens0step™ Encoder

+

CANopen



### UNIQUE FEATURES



coolStep™

stallGuard<sup>2</sup>™

TRINAMIC Motion Control GmbH & Co. KG  
Hamburg, Germany

[www.trinamic.com](http://www.trinamic.com)



**TRINAMIC**  
MOTION CONTROL

# Table of Contents

1	Preface .....	6
1.1	General Features of the CANopen Implementation .....	6
1.2	Abbreviations .....	7
1.3	Firmware Update .....	7
2	Communication .....	9
2.1	Reference Model .....	9
2.2	NMT State Machine .....	10
2.3	Device Model .....	12
2.4	Object Dictionary .....	13
3	Communication Objects .....	14
3.1	Detailed Object Specifications .....	14
3.1.1	Object 1000 <sub>h</sub> : Device Type .....	14
3.1.2	Object 1001 <sub>h</sub> : Error Register .....	14
3.1.3	Object 1005 <sub>h</sub> : COB-ID SYNC Message .....	15
3.1.4	Object 1008 <sub>h</sub> : Manufacturer Device Name .....	15
3.1.5	Object 1009 <sub>h</sub> : Manufacturer Hardware Version .....	15
3.1.6	Object 100A <sub>h</sub> : Manufacturer Software Version .....	15
3.1.7	Object 100C <sub>h</sub> : Guard Time .....	16
3.1.8	Object 100D <sub>h</sub> : Life Time Factor .....	16
3.1.9	Object 1010 <sub>h</sub> : Store Parameters .....	16
3.1.10	Object 1011 <sub>h</sub> : Restore Default Parameters .....	17
3.1.11	Object 1014 <sub>h</sub> : COB-ID Emergency Object .....	19
3.1.12	Object 1015 <sub>h</sub> : Inhibit Time EMCY .....	19
3.1.13	Object 1016 <sub>h</sub> : Consumer Heartbeat Time .....	19
3.1.14	Object 1017 <sub>h</sub> : Producer Heartbeat Time .....	19
3.1.15	Object 1018 <sub>h</sub> : Identity Object .....	20
3.1.16	Object 1023 <sub>h</sub> : OS Command .....	20
3.1.17	Object 1029 <sub>h</sub> : Error Behavior .....	20
3.1.18	Objects 1400 <sub>h</sub> -1405 <sub>h</sub> : Receive PDO Communication Parameter .....	21
3.1.19	Objects 1600 <sub>h</sub> -1605 <sub>h</sub> : Receive PDO Mapping .....	21
3.1.20	Objects 1800 <sub>h</sub> -1805 <sub>h</sub> : Transmit PDO Communication Parameter .....	22
3.1.21	Objects 1A00 <sub>h</sub> -1A05 <sub>h</sub> : Transmit PDO Mapping Parameter .....	23
4	Device Profile Objects (CiA402) and Modes of Operation .....	24
4.1	Detailed Object Specifications .....	24
4.1.1	Object 605A <sub>h</sub> : Quick Stop Option Code .....	24
4.1.2	Object 605B <sub>h</sub> : Shutdown Option Code .....	25
4.1.3	Object 605C <sub>h</sub> : Disable Operation Option Code .....	25
4.1.4	Object 605D <sub>h</sub> : Halt Option Code .....	25
4.1.5	Object 605E <sub>h</sub> : Fault Reaction Option Code .....	26
4.1.6	Object 6060 <sub>h</sub> : Modes of Operation .....	26
4.1.7	Object 6061 <sub>h</sub> : Modes of Operation Display .....	26
4.1.8	Object 606A <sub>h</sub> : Sensor Selection Code .....	27
4.1.9	Object 6085 <sub>h</sub> : Quick Stop Deceleration .....	27
4.1.10	Object 608F <sub>h</sub> : Position Encoder Resolution .....	27
4.1.11	Object 60C5 <sub>h</sub> : Max Acceleration .....	28
4.1.12	Object 60C6 <sub>h</sub> : Max Deceleration .....	28
4.1.13	Object 60FD <sub>h</sub> : Digital Inputs .....	28
4.1.14	Object 6502 <sub>h</sub> : Supported Drive Modes .....	29
4.2	Profile Position Mode .....	30
4.2.1	Detailed Object Specifications .....	30
4.2.1.1	Object 6040 <sub>h</sub> : Controlword .....	31
4.2.1.2	Object 6041 <sub>h</sub> : Statusword .....	32
4.2.1.3	Object 6062 <sub>h</sub> : Position Demand Value .....	33
4.2.1.4	Object 6063 <sub>h</sub> : Position Actual Internal Value .....	33
4.2.1.5	Object 6064 <sub>h</sub> : Position Actual Value .....	33
4.2.1.6	Object 6065 <sub>h</sub> : Following Error Window .....	34

4.2.1.7	Object 6067 <sub>h</sub> : Position Window .....	34
4.2.1.8	Object 6068 <sub>h</sub> : Position Window Time.....	34
4.2.1.9	Object 606C <sub>h</sub> : Velocity Actual Value .....	35
4.2.1.10	Object 607A <sub>h</sub> : Target Position .....	35
4.2.1.11	Object 607D <sub>h</sub> : Software Position Limit.....	35
4.2.1.12	Object 607F <sub>h</sub> : Maximum Profile Velocity.....	36
4.2.1.13	Object 6081 <sub>h</sub> : Profile Velocity.....	36
4.2.1.14	Object 6083 <sub>h</sub> : Profile Acceleration .....	36
4.2.1.15	Object 6084 <sub>h</sub> : Profile Deceleration .....	37
4.2.2	How to Move a Motor in <i>pp</i> Mode .....	37
4.3	Homing Mode .....	38
4.3.1	Connecting Home Switch, Left Switch, and Right Switch.....	39
4.3.1.1	TMC1140.....	39
4.3.1.2	TMC1160.....	40
4.3.1.2.1	Digital Inputs STOP_L, STOP_R, and HOME.....	41
4.3.1.3	TMC1180.....	42
4.3.1.3.1	Connect a Brake .....	43
4.3.2	Homing Methods .....	44
4.3.3	Detailed Object Definitions .....	47
4.3.3.1	Object 6040 <sub>h</sub> : Controlword.....	47
4.3.3.2	Object 6041 <sub>h</sub> : Statusword.....	48
4.3.3.3	Object 606C <sub>h</sub> : Velocity Actual Value .....	48
4.3.3.4	Object 607C <sub>h</sub> : Home Offset .....	49
4.3.3.5	Object 6098 <sub>h</sub> : Homing Method.....	49
4.3.3.6	Object 6099 <sub>h</sub> : Homing Speeds .....	50
4.3.3.7	Object 609A <sub>h</sub> : Homing Acceleration .....	50
4.3.3.8	Object 2100 <sub>h</sub> : Home Offset Display .....	50
4.3.4	How to Start a Homing in <i>hm</i> Mode .....	51
4.4	Velocity Mode .....	52
4.4.1	Detailed Object Definitions .....	52
4.4.1.1	Object 6040 <sub>h</sub> : Controlword.....	52
4.4.1.2	Object 6041 <sub>h</sub> : Statusword.....	53
4.4.1.3	Object 6042 <sub>h</sub> : vI Target Velocity .....	53
4.4.1.4	Object 6043 <sub>h</sub> : vI Velocity Demand .....	54
4.4.1.5	Object 6044 <sub>h</sub> : vI Velocity Actual Value .....	54
4.4.1.6	Object 6046 <sub>h</sub> : vI Velocity Min Max Amount.....	54
4.4.1.7	Object 6048 <sub>h</sub> : vI Velocity Acceleration .....	55
4.4.1.8	Object 6049 <sub>h</sub> : vI Velocity Deceleration .....	55
4.4.2	How to Move a Motor in <i>vI</i> Mode.....	55
4.5	Profile Velocity Mode .....	56
4.5.1.1	Object 6040 <sub>h</sub> : Controlword.....	56
4.5.1.2	Object 6041 <sub>h</sub> : Statusword.....	57
4.5.1.3	Object 6062 <sub>h</sub> : Position Demand Value.....	57
4.5.1.4	Object 6063 <sub>h</sub> : Position Actual Internal Value .....	57
4.5.1.5	Object 6064 <sub>h</sub> : Position Actual Value.....	58
4.5.1.6	Object 6065 <sub>h</sub> : Following Error Window .....	58
4.5.1.7	Object 606C <sub>h</sub> : Velocity Actual Value .....	58
4.5.1.8	Object 607D <sub>h</sub> : Software Position Limit.....	59
4.5.1.9	Object 6083 <sub>h</sub> : Profile Acceleration .....	59
4.5.1.10	Object 6084 <sub>h</sub> : Profile Deceleration .....	59
4.5.1.11	Object 60FF <sub>h</sub> : Target Velocity .....	60
4.5.2	How to Move a Motor in <i>pV</i> Mode.....	60
5	Manufacturer Specific Area .....	61
5.1	coolStep™ related Objects (Overview).....	61
5.2	Detailed Object Specifications .....	63
5.2.1	Object 2000 <sub>h</sub> : Microstep Resolution .....	63
5.2.2	Object 2001 <sub>h</sub> : Fullstep Resolution.....	63

5.2.3	Object 2002 <sub>h</sub> : Brake Delay Times .....	63
5.2.4	Object 2003 <sub>h</sub> : Maximum Current.....	64
5.2.5	Object 2004 <sub>h</sub> : Standby Current.....	64
5.2.6	Object 2005 <sub>h</sub> : Limit Switches.....	64
5.2.7	Object 200A <sub>h</sub> : Enable Drive Delay Time .....	65
5.2.8	Object 200B <sub>h</sub> : Encoder Parameters .....	65
5.2.9	Object 200C <sub>h</sub> : Brake Current Feed.....	65
5.2.10	Object 204E <sub>h</sub> : Boost Current .....	66
5.2.11	Object 2085 <sub>h</sub> : Ramp Divisor .....	66
5.2.12	Object 2086 <sub>h</sub> : Pulse Divisor.....	66
5.2.13	Object 2087 <sub>h</sub> : Maximum Velocity.....	67
5.2.14	Object 2088 <sub>h</sub> : Maximum Acceleration .....	67
5.2.15	Object 2089 <sub>h</sub> : Standby Delay .....	67
5.2.16	Object 208C <sub>h</sub> : Velocity Dimension Index .....	68
5.2.17	Object 208E <sub>h</sub> : Acceleration Dimension Index .....	68
5.2.18	Object 2092 <sub>h</sub> : Chopper Blank Time.....	68
5.2.19	Object 2093 <sub>h</sub> : Chopper Mode.....	69
5.2.20	Object 2094 <sub>h</sub> : Chopper Hysteresis Decrement .....	69
5.2.21	Object 2095 <sub>h</sub> : Chopper Hysteresis End .....	69
5.2.22	Object 2096 <sub>h</sub> : Chopper Hysteresis Start.....	70
5.2.23	Object 2097 <sub>h</sub> : Chopper Off Time.....	70
5.2.24	Object 2098 <sub>h</sub> : Smart Energy Current Minimum .....	70
5.2.25	Object 2099 <sub>h</sub> : Smart Energy Current down Step .....	71
5.2.26	Object 209A <sub>h</sub> : Smart Energy Hysteresis.....	71
5.2.27	Object 209B <sub>h</sub> : Smart Energy Current up Step .....	71
5.2.28	Object 209C <sub>h</sub> : Smart Energy Hysteresis Start.....	72
5.2.29	Object 209D <sub>h</sub> : stallGuard2 Filter Enable .....	72
5.2.30	Object 209E <sub>h</sub> : stallGuard2 Threshold.....	72
5.2.31	Object 209F <sub>h</sub> : Slope Control High Side.....	73
5.2.32	Object 20A0 <sub>h</sub> : Slope Control Low Side.....	73
5.2.33	Object 20A1 <sub>h</sub> : Short Protection Disable .....	73
5.2.34	Object 20A2 <sub>h</sub> : Short Detection Timer .....	74
5.2.35	Object 20A3 <sub>h</sub> : Vsense.....	74
5.2.36	Object 20A4 <sub>h</sub> : Stop on Stall .....	74
5.2.37	Object 20A5 <sub>h</sub> : Smart Energy Threshold Speed .....	74
5.2.38	Object 20A6 <sub>h</sub> : Smart Energy Slow Run Current .....	75
5.2.39	Object 2100 <sub>h</sub> : Home Offset Display .....	75
5.2.40	Object 2101 <sub>h</sub> : Actual stallGuard2 Load Value .....	75
5.2.41	Object 2102 <sub>h</sub> : Driver Error Flags .....	75
5.2.42	Object 2103 <sub>h</sub> : Pulse Divisor Display .....	76
5.2.43	Object 2104 <sub>h</sub> : Maximum Velocity Display .....	76
5.2.44	Object 2105 <sub>h</sub> : Ramp Divisor Display.....	76
5.2.45	Object 2106 <sub>h</sub> : Maximum Acceleration Display .....	76
5.2.46	Object 2107 <sub>h</sub> : Microstep Resolution Display .....	77
5.2.47	Object 2700 <sub>h</sub> : TMCL Direct Communication.....	77
5.2.48	Object 2701 <sub>h</sub> : Manufacturer Specific Mode.....	77
5.2.49	Object 2702 <sub>h</sub> : Digital Inputs .....	78
5.2.50	Object 2703 <sub>h</sub> : Digital Outputs.....	78
5.2.51	Object 2704 <sub>h</sub> : CAN Bit Rate .....	78
5.2.52	Object 2705 <sub>h</sub> : Node ID.....	79
5.2.53	Object 2706 <sub>h</sub> : Store.....	79
5.2.54	Object 2707 <sub>h</sub> : CAN Bit Rate Load.....	79
5.2.55	Object 2708 <sub>h</sub> : Node ID Load .....	80
5.2.56	Object 270E <sub>h</sub> : Device Analogue Inputs.....	80
6	Emergency Messages .....	81
7	Life Support Policy.....	83
8	Revision History.....	84
8.1	Firmware Revision.....	84

8.2	Document Revision .....	84
9	References .....	85

## Table of Figures

Figure 1.1	stallGuard2 load measurement SG as a function of load .....	8
Figure 1.2	Energy efficiency example with coolStep.....	8
Figure 2.1	Overview CANopen NMT state machine.....	10
Figure 2.2	Communication architecture .....	11
Figure 2.3	Device model.....	12
Figure 3.1	Restore procedure .....	18
Figure 4.1	Finite state machine .....	30
Figure 4.2	Homing mode function.....	38
Figure 4.9	Multi-purpose I/O connector of TCM-1140.....	39
Figure 4.10	Multi-purpose I/O of TCM-1160.....	40
Figure 4.11	I/O connectors of the TCM-1180 .....	42
Figure 4.3	Homing on negative switch and index pulse .....	44
Figure 4.4	Homing on positive limit switch and index pulse .....	44
Figure 4.5	Homing on positive home switch and index pulse.....	45
Figure 4.6	Homing on negative home switch and index pulse.....	45
Figure 4.7	Homing without index pulse .....	46
Figure 4.8	Homing on index pulse .....	46
Figure 4.12	Home offset definition.....	49
Figure 4.13	Transfer characteristic of the velocity acceleration .....	55
Figure 5.1	Brake output timing.....	63

# 1 Preface

This document specifies objects and modes of operation for the TCM1140, TCM1160, and TCM1180 with CANopen firmware. As these products are stepper motor controller and driver modules the use of the CiA DSP402 protocol (described in the *CiA CANopen drives and motion control device profile, Part 2*) is fundamental. The CANopen firmware is designed to fulfill the DS301 version 4.02 and DS402 version 3.0 standards. The CiA conformance has also been tested. This manual assumes that the reader is already familiar with the basics of the CANopen protocol (especially DS301 and DS402). On the following pages you will find an overview and afterwards the information will be more in detail.

If necessary, you can always turn the board back into a TMCL module by loading the TMCL firmware into the drive again.

## 1.1 General Features of the CANopen Implementation

### MAIN CHARACTERISTICS

Communication according to standard CiA-301 V4.1  
CAN bit rate: 20... 1000kBit/s  
CAN ID: 11 bit  
Node ID: 1... 127 (use vendor specific objects)  
NMT services: NMT slave

### SDO communication

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

### PDO communication

- Producer
- Consumer
- *RPDOs*
  - 1, 2, 3, 4, 6
  - Transmission modes: asynchronous
  - Dynamic mapping with max. 3 mapping entries
  - Default mappings: according to CiA-402 for PDO 1, 2, 3 and 6, manufacturer specific for PDO4
- *TPDOs*
  - 1, 2, 3, 4, 6
  - Transmission modes: asynchronous, asynchronous with event timer, synchronous
  - Dynamic mapping with max. 3 mapping entries
  - Default mappings: according to CiA-402 for PDO 1, 2, 3 and 6, manufacturer specific for PDO

### Further Characteristics

SYNC: consumer (TPDO3 and TPDO6 are synchronous PDOs)  
Emergency: producer  
RTR: supported only for node guarding/life guarding

## 1.2 Abbreviations

ABBREVIATIONS	
CAN	Controller area network
CHGND	chassis ground / earth ground
COB	Communication object
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 1.1 Abbreviations**

## 1.3 Firmware Update

The software running on the microprocessor of each module consists of two parts, a boot loader and the CANopen firmware itself. Whereas the boot loader is installed during production and testing at TRINAMIC and remains untouched throughout the whole lifetime, the CANopen firmware can easily be updated by the user. The new firmware can be loaded into the module via the firmware update function of the TMCL-IDE.

### TMCM-1140 AND TMCM1160

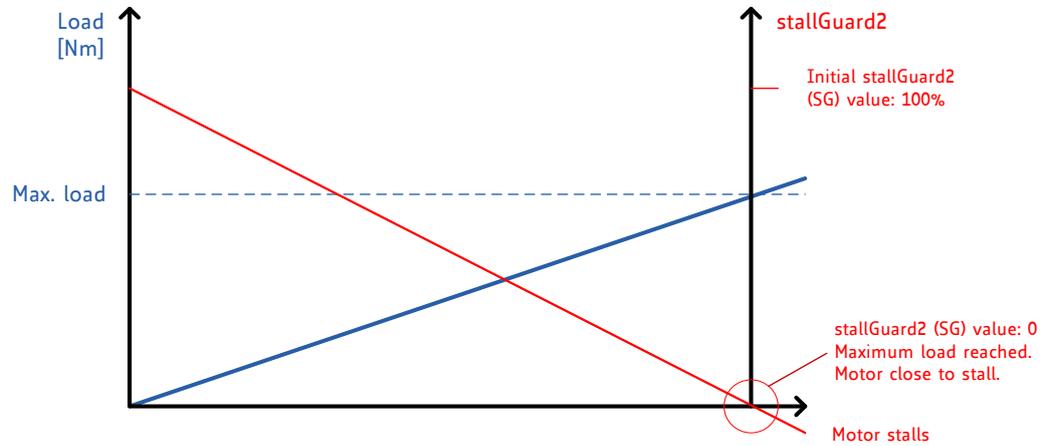
Connect the USB interface and use the bus-powered mode for a firmware update. Do not connect the power supply of the module.

### TMCM-1180

Update the firmware using the TMCL-IDE. It is not necessary to use a specific interface.

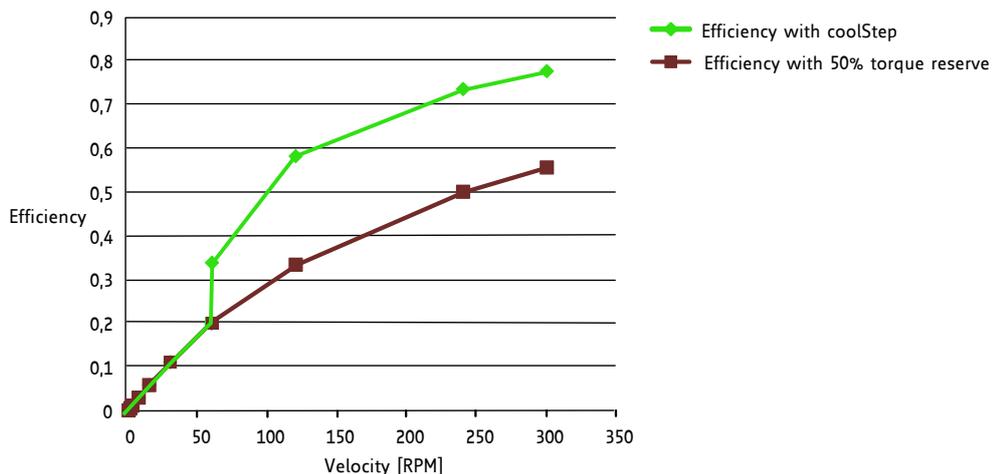
## TRINAMICS UNIQUE FEATURES – EASY TO USE WITH CANOPEN

**stallGuard2™** stallGuard2 is a high-precision sensorless load measurement using the back EMF on 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 goes to zero or near to zero. This is the most energy-efficient point of operation for the motor.



**Figure 1.1** stallGuard2 load measurement SG as a function of load

**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 reducing cost.



**Figure 1.2** Energy efficiency example with coolStep

## 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 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 CAN Network with peer service object(s) using a protocol.

The application and the application layer interact with *service primitives*.

SERVICE PRIMITIVES	
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 2.1 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	
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 2.2 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 contains the following states:

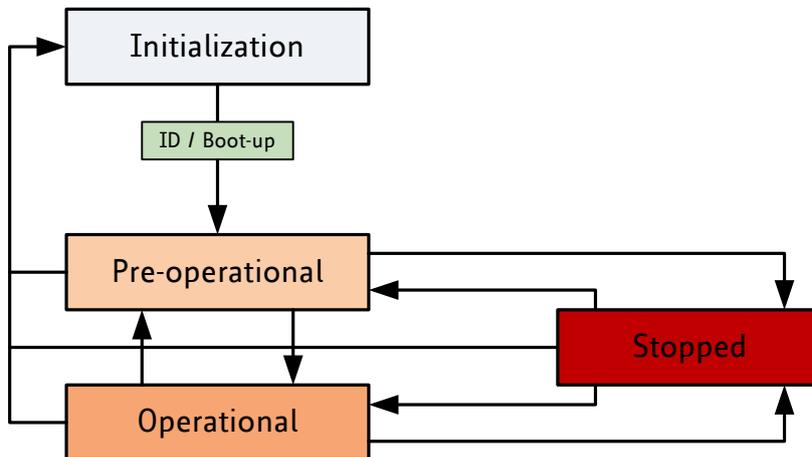


Figure 2.1 Overview CANopen NMT state machine

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

After the device initialization is finished, the device automatically transits to the **Pre-operational state** and indicates this state transition by sending the boot-up message. This way the device indicates that it is ready to work. A device that stays in Pre-operational state may start to transmit SYNC-, time stamp- or heartbeat message. In contrast to the PDO communication that has to be disabled in this state, the device can communicate via SDO.

The PDO communication is only possible within the **Operational state**. During Operational state the device can use all supported communication objects.

A device that was switched to the **Stopped state** only reacts on received NMT commands. In addition the device indicates the current NMT state by supporting the error control protocol during Stopped state.

The transitions between states are made by issuing a network management (NMT) communication object to the device. The NMT protocols are used to generate state machine change commands (e.g. to start and stop the device), detect remote device boot-ups and error conditions.

The Heartbeat message of a CANopen device contains the device status of the NMT state machine and is sent cyclically by the CANopen device.

The Figure 2.2 shows the situation of the state machine in this device profile.

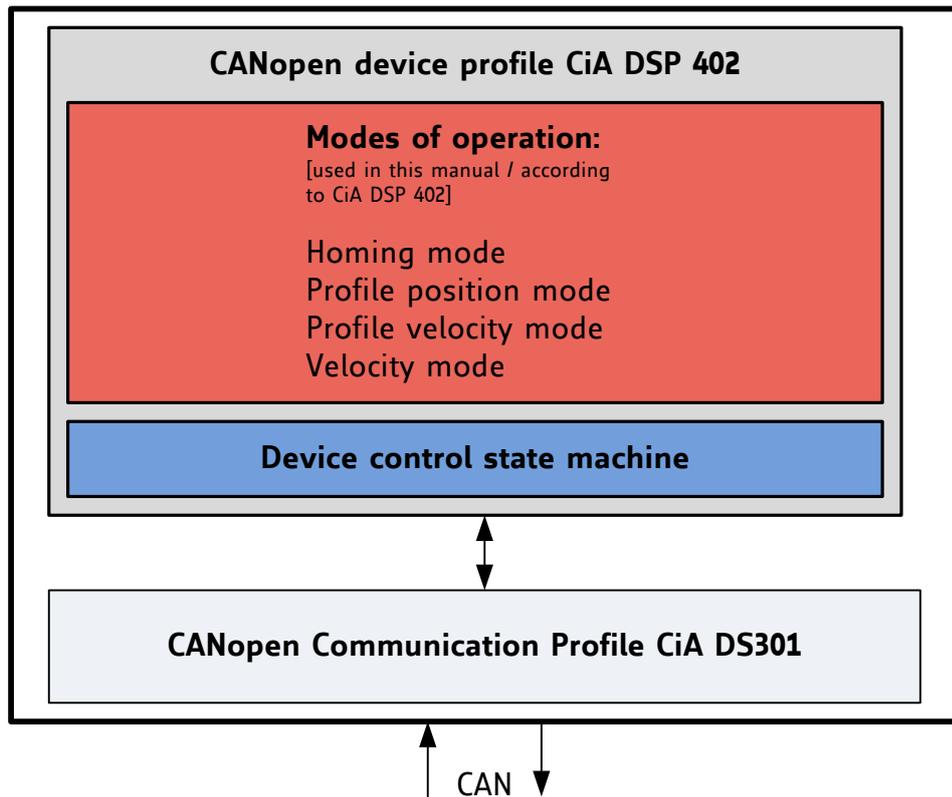


Figure 2.2 Communication architecture

## 2.3 Device Model

Following the *device model*, the device consists of three parts: communication, object dictionary, and application.

*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.

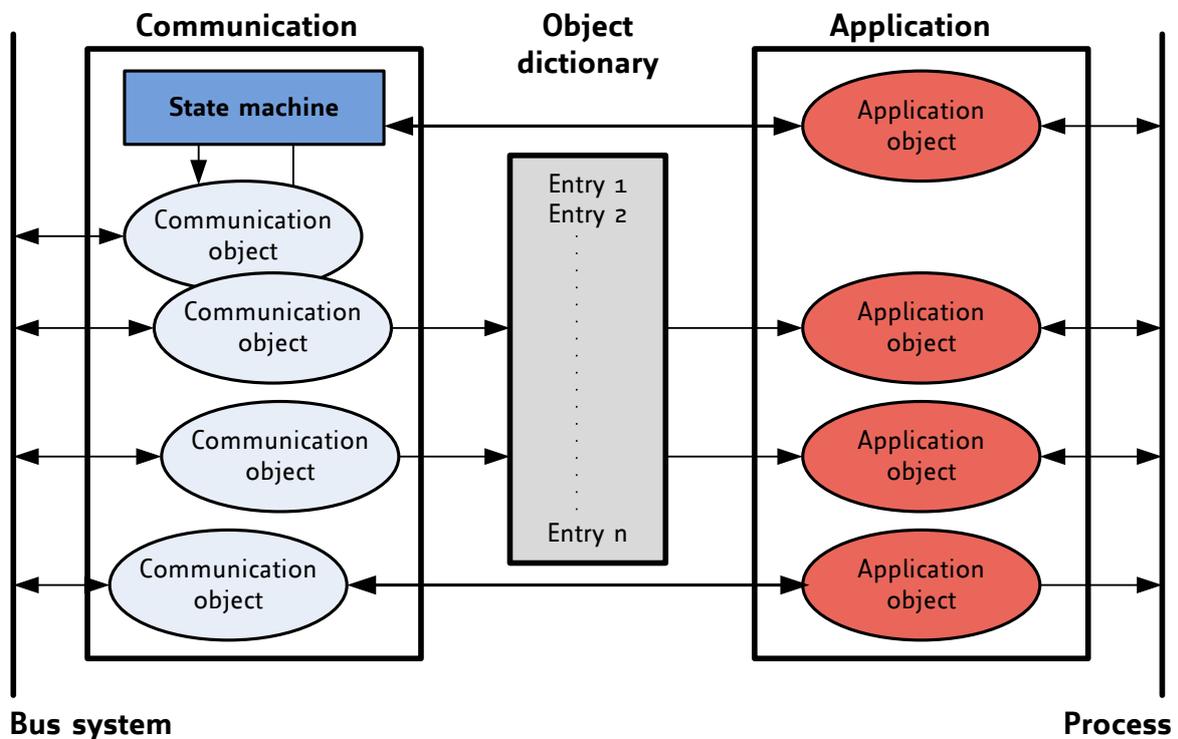


Figure 2.3 Device model

## 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 AS FOLLOWS:

Index (hex)	Object
0000	Not used
0001 - 001F	Static data types
0020 - 003F	Complex data types
0040 - 005F	Manufacturer specific complex data types
0060 - 007F	Device profile specific static data types
0080 - 009F	Device profile specific complex data types
00A0 - 0FFF	Reserved for further use
1000 - 1FFF	Communication profile area <sup>*1</sup>
2000 - 5FFF	Manufacturer specific profile area <sup>*2</sup>
6000 - 9FFF	Standardized device profile area <sup>*3</sup>
A000 - BFFF	Standardized interface profile area
C000 - FFFF	Reserved for further use

**Table 2.3 Object Dictionary**

- \*1 The communication profile area at indices 1000<sub>h</sub> through 1FFF<sub>h</sub> contains the communication specific parameters for the CAN network. These entries are common to all devices.
- \*2 The manufacturer segment at indices 2000<sub>h</sub> through 5FFF<sub>h</sub> contains manufacturer specific objects. These objects control the special features of the TRINAMIC motion control devices TCM-1140, TCM-1160, and TCM-1180.
- \*3 The standardized device profile area at indices 6000<sub>h</sub> through 9FFF<sub>h</sub> contains all data objects common to a class of devices that can be read or written via the network. The device profiles use entries from 6000<sub>h</sub> to 9FFF<sub>h</sub> to describe the device parameters and the device functionality.

## 3 Communication Objects

### 3.1 Detailed Object Specifications

#### 3.1.1 Object 1000<sub>h</sub>: Device Type

This object contains information about the device type. The object 1000<sub>h</sub> 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 gives additional information about optional functionality of the device.

##### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
1000 <sub>h</sub>	Device type	Variable	UNSIGNED32

##### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	ro	no	UNSIGNED32	40192 <sub>h</sub>

#### 3.1.2 Object 1001<sub>h</sub>: Error Register

This object is an error register. The module can map internal errors and object 1001<sub>h</sub> is part of an emergency object.

##### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
1001 <sub>h</sub>	Error register	Variable	UNSIGNED8

##### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	ro	yes	UNSIGNED8	no

##### STRUCTURE OF THE ERROR REGISTER

Bit	M/O	Description
0	M	Generic error
1	0	Current
2	0	Voltage
3	0	Temperature
4	0	Communication error
5	0	Device profile specific
6	0	Reserved (always 0)
7	0	Manufacturer specific

If a bit is set to 1, the specific error has occurred.

### 3.1.3 Object 1005<sub>h</sub>: COB-ID SYNC Message

This object defines the COB-ID of the synchronization Object (SYNC). Further, it defines whether the module generates the SYNC.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
1005 <sub>h</sub>	COB-ID SYNC Message	Variable	UNSIGNED32

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	UNSIGNED32	80 <sub>h</sub>

### 3.1.4 Object 1008<sub>h</sub>: Manufacturer Device Name

This object contains the manufacturer device name.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
1008 <sub>h</sub>	Manufacturer device name	Variable	Visible string

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	const.	no	no	TMC-1140 TMC-1160 TMC-1180 <i>depends on module</i>

### 3.1.5 Object 1009<sub>h</sub>: Manufacturer Hardware Version

This object contains the hardware version description.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
1009 <sub>h</sub>	Manufacturer hardware version	Variable	Visible string

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	const.	no	no	<i>depends on module</i>

### 3.1.6 Object 100A<sub>h</sub>: Manufacturer Software Version

This object contains the software version description.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
100A <sub>h</sub>	Manufacturer software version	Variable	Visible string

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	Const.	no	no	<i>according to software version</i>

### 3.1.7 Object 100C<sub>h</sub>: Guard Time

The objects at index 100C<sub>h</sub> and 100D<sub>h</sub> shall indicate the configured guard time respectively the life time factor. The life time factor multiplied with the guard time gives the life time for the life guarding protocol.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
100C <sub>h</sub>	Guard time	Variable	UNSIGNED16

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw <i>ro, if life guarding is not supported</i>	no	UNSIGNED16	0000 <sub>h</sub>

### 3.1.8 Object 100D<sub>h</sub>: Life Time Factor

The life time factor multiplied with the guard time gives the life time for the life guarding protocol.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
100D <sub>h</sub>	Life time factor	Variable	UNSIGNED8

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw <i>ro, if life guarding is not supported</i>	no	UNSIGNED8	00 <sub>h</sub>

### 3.1.9 Object 1010<sub>h</sub>: Store Parameters

This object supports the saving of parameters in non volatile memory. By read access the device provides information about its saving capabilities.

This command can only be carried out if the module is in ready to switch on mode.

#### SEVERAL PARAMETER GROUPS ARE DISTINGUISHED:

- Sub-index 0<sub>h</sub>: contains the largest sub-index that is supported.
- Sub-index 1<sub>h</sub>: saves all parameters.
- Sub-index 2<sub>h</sub>: saves communication parameters 100C<sub>h</sub>, 100D<sub>h</sub>, 1015<sub>h</sub>, 1017<sub>h</sub>, and 1029<sub>h</sub>.
- Sub-index 4<sub>h</sub>: saves motor axis 1 parameters (6040<sub>h</sub>... 6084<sub>h</sub>)
- Sub-index 7<sub>h</sub>: saves device parameters / TRINAMIC specific parameters (2000<sub>h</sub>... 270E<sub>h</sub>).

In order to avoid storage of parameters by mistake, storage is only executed when a specific signature is written to the appropriate Sub-Index. This signature is *safe*.

#### Storage write access structure

Signature	MSB		LSB	
ISO 8859 ASCII	e	v	a	s
hex	65 <sub>h</sub>	76 <sub>h</sub>	61 <sub>h</sub>	73 <sub>h</sub>

On reception of the correct signature in the appropriate sub-index the device stores the parameter and then confirms the SDO transmission (initiate download response). If the storing failed, the device responds with an Abort SDO transfer (abort code: 0606 0000<sub>h</sub>).

If a wrong signature is written, the device refuses to store and responds with Abort SDO transfer (abort code: 0800 002x<sub>h</sub>).

On read access to the appropriate sub-index the device provides information about its storage functionality with the following format.

#### Storage read access structure

UNSIGNED 32			
MSB		LSB	
bits	31-2	1	0
	Reserved	1/0	1/0

Bit-number	Value	Meaning
31-2	0	reserved
1	0	Device does not save parameters autonomously
	1	Device saves parameters autonomously
0	0	Device does not save parameters on command
	1	Device saves parameters on command

Autonomous saving means that a device stores the storable parameters in a non-volatile manner without user request.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
1010 <sub>h</sub>	Store parameters	ARRAY	UNSIGNED 32

#### ENTRY DESCRIPTION

Sub-Index	Description	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	Largest sub-index supported	ro	no	1 <sub>h</sub> -7F <sub>h</sub>	no
01 <sub>h</sub>	Save all parameters	rw	no	UNSIGNED32	no
02 <sub>h</sub>	Save special communication parameters	rw	no	UNSIGNED32	no
04 <sub>h</sub>	Save motor axis 1 parameters	rw	no	UNSIGNED32	no
07 <sub>h</sub>	Save device parameters (TRINAMIC specific parameters)	rw	no	UNSIGNED32	no

Please mind the figures above which explain the value ranges of the write access and the read access for the sub-indices.

### 3.1.10 Object 1011<sub>h</sub>: Restore Default Parameters

With this object the default values of parameters according to the communication or device profile are restored. By read access the device provides information about its capabilities to restore these values.

This command can only be carried out if the module is in ready to switch on mode.

#### SEVERAL PARAMETER GROUPS ARE DISTINGUISHED:

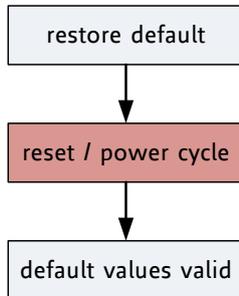
- Sub-index 0<sub>h</sub>: contains the largest sub-index that is supported.
- Sub-index 1<sub>h</sub>: restores all parameters.
- Sub-index 2<sub>h</sub>: restores communication parameters 100C<sub>h</sub>, 100D<sub>h</sub>, 1015<sub>h</sub>, 1017<sub>h</sub>, and 1029<sub>h</sub>.
- Sub-index 4<sub>h</sub>: restores motor axis 1 parameters (6040<sub>h</sub>... 6084<sub>h</sub>)
- Sub-index 7<sub>h</sub>: restores device parameters / TRINAMIC specific parameters (2000<sub>h</sub>... 270E<sub>h</sub>).

In order to avoid the restoring of default parameters by mistake, restoring is only executed when a specific signature is written to the appropriate sub-index. This signature is *load*.

Signature	MSB		LSB	
ASCII	d	a	o	l
hex	64 <sub>h</sub>	61 <sub>h</sub>	6F <sub>h</sub>	6C <sub>h</sub>

On reception of the correct signature in the appropriate sub-index the device restores the default parameters and then confirms the SDO transmission (initiate download response). If the restoring failed, the device responds with an Abort SDO Transfer (abort code: 0606 0000<sub>h</sub>). If a wrong signature is written, the device refuses to restore the defaults and responds with an Abort SDO Transfer (abort code: 0800 002x<sub>h</sub>).

The default values are set valid after the device is reset (reset node for sub-index 1h – 7Fh, reset communication for sub-index 2<sub>h</sub>) or power cycled.



**Figure 3.1 Restore procedure**

On read access to the appropriate sub-index the device provides information about its default parameter restoring capability with the following format.

Structure of restore read access

	UNSIGNED 32	
	MSB	LSB
bits	31-1	0
	Reserved (0)	0/1

**OBJECT DESCRIPTION**

Index	Name	Object Code	Data Type
1011 <sub>h</sub>	Restore default parameters	ARRAY	UNSIGNED 32

**ENTRY DESCRIPTION**

Sub-Index	Description	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	Largest sub-index supported	ro	no	1 <sub>h</sub> -7F <sub>h</sub>	no
01 <sub>h</sub>	Restore all default parameters	rw	no	UNSIGNED32	no
02 <sub>h</sub>	Restore special communication parameters	rw	no	UNSIGNED32	no
04 <sub>h</sub>	Restore motor axis 1 parameters	rw	no	UNSIGNED32	no
07 <sub>h</sub>	Restore device parameters (TRINAMIC specific parameters)	rw	no	UNSIGNED32	no

Please mind the figures above which explain the value ranges of the write access and the read access for the sub-indices.

### 3.1.11 Object 1014<sub>h</sub>: COB-ID Emergency Object

This object defines the COB-ID of the emergency object (EMCY).

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
1014 <sub>h</sub>	COB-ID emergency object	Variable	UNSIGNED32

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	UNSIGNED32	80 <sub>h</sub> + NODE ID

### 3.1.12 Object 1015<sub>h</sub>: Inhibit Time EMCY

The inhibit time for the EMCY message can be adjusted via this entry. The time has to be a multiple of 100µs.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
1015 <sub>h</sub>	Inhibit time EMCY	Variable	UNSIGNED16

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	UNSIGNED16	0

### 3.1.13 Object 1016<sub>h</sub>: Consumer Heartbeat Time

The consumer heartbeat time defines the expected heartbeat cycle time and thus has to be higher than the corresponding producer heartbeat time configured on the module producing this heartbeat. The monitoring starts after the reception of the first heartbeat. If the consumer heartbeat time is 0 the corresponding entry is not used. The time has to be a multiple of 1ms.

	MSB		LSB
<i>bits</i>	31-24	23-16	15-0
<i>value</i>	Reserved	Node-ID	Heartbeat time
<i>encoded as</i>	-	UNSIGNED8	UNSIGNED16

Table 3.1 Structure of consumer heartbeat time entry

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
1016 <sub>h</sub>	Consumer heartbeat time	ARRAY	UNSIGNED32

#### ENTRY DESCRIPTION

Sub-Index	Description	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	Number of entries	ro	no	0... 127	no
01 <sub>h</sub>	Consumer heartbeat time	rw	no	UNSIGNED32	no

### 3.1.14 Object 1017<sub>h</sub>: Producer Heartbeat Time

The producer heartbeat time defines the cycle time of the heartbeat. The producer heartbeat time is 0 if it is not used. The time has to be a multiple of 1ms.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
1017 <sub>h</sub>	Producer heartbeat time	Variable	UNSIGNED16

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	UNSIGNED16	0

### 3.1.15 Object 1018<sub>h</sub>: Identity Object

The object 1018<sub>h</sub> contains general information about the device.

- The Vendor ID (sub-index 01<sub>h</sub>) contains a unique value allocated to each manufacturer. The vendor ID of TRINAMIC is 0286<sub>h</sub>.
- The manufacturer-specific Product code (sub-index 2<sub>h</sub>) identifies a specific device version.
- The Manufacturer-specific Revision number (sub-index 3<sub>h</sub>) consists of a major revision number and a minor revision number.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type	Category
1018 <sub>h</sub>	Identity object	RECORD	Identity	Optional

#### ENTRY DESCRIPTION

Sub-Index	Description	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	Number of entries	ro	no	0.. 3	number of entries:3
01 <sub>h</sub>	Vendor ID (TRINAMIC)	ro	no	UNSIGNED32	reads 0286 <sub>h</sub>
02 <sub>h</sub>	Product code	ro	no	UNSIGNED32	1180, 1160, or 1140
03 <sub>h</sub>	Revision number	ro	no	UNSIGNED32	firmware revision number; reads e.g. 3016 <sub>h</sub> for version 3.16
04 <sub>h</sub>	Serial number	ro	no	UNSIGNED32	serial number

### 3.1.16 Object 1023<sub>h</sub>: OS Command

The OS Command object is used as a command driven interface to programmable devices. The host system puts the command into the object OS Command, which is of the type Command Par.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type	Category
1023 <sub>h</sub>	OS command	RECORD	Command Par	Optional

#### ENTRY DESCRIPTION

Sub-Index	Description	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	Number of supported entries	ro	no	3	number of entries: 3 (for direct TMCL communication)
01 <sub>h</sub>	Command (TMCL command)	rw	no	Octet string	no
02 <sub>h</sub>	Status (error code of a TMCL command)	ro	no	UNSIGNED8	no
03 <sub>h</sub>	Reply (reply of a TMCL command)	ro	no	Octet string	no

### 3.1.17 Object 1029<sub>h</sub>: Error Behavior

If a device failure is detected in operational state, the device can be configured to enter alternatively the stopped state or remain in the current state in case of a device failure. Device failures include the following errors:

- Communication error
- Application error

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
1029 <sub>h</sub>	Error behavior	ARRAY	UNSIGNED8

#### ENTRY DESCRIPTION

Sub-Index	Description	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	Number of error classes	ro	no	2 <sub>h</sub>	2
01 <sub>h</sub>	Communication error	rw	no	UNSIGNED8	0 (enter stopped state)
02 <sub>h</sub>	Application error	rw	no	UNSIGNED8	1 (remain in current state)

### 3.1.18 Objects 1400<sub>h</sub>-1405<sub>h</sub>: Receive PDO Communication Parameter

This object contains the communication parameters for the PDOs the device is able to receive. The sub-index 0<sub>h</sub> contains the number of valid entries within the communication record. Its value is at least 2.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
1400 <sub>h</sub> -1405 <sub>h</sub>	Receive PDO parameter	RECORD	PDO CommPar
1400 <sub>h</sub>	RPDO 1	RECORD	PDO CommPar
1401 <sub>h</sub>	RPDO 2	RECORD	PDO CommPar
1402 <sub>h</sub>	RPDO 3	RECORD	PDO CommPar
1403 <sub>h</sub>	RPDO 4	RECORD	PDO CommPar
1405 <sub>h</sub>	RPDO 6	RECORD	PDO CommPar

#### ENTRY DESCRIPTION

Sub-Index	Description	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	Largest sub-index supported	ro	no	Number of entries: 2	2
01 <sub>h</sub>	COB-ID used by PDO	rw	no	UNSIGNED32	Index 1400 <sub>h</sub> : 200 <sub>h</sub> + Node-ID Index 1401 <sub>h</sub> : 300 <sub>h</sub> + Node-ID Index 1402 <sub>h</sub> : 400 <sub>h</sub> + Node-ID Index 1403 <sub>h</sub> : 500 <sub>h</sub> + Node-ID Index 1405 <sub>h</sub> : 0
02 <sub>h</sub>	Transmission type	rw	no	UNSIGNED8	Index 1400 <sub>h</sub> : Ff <sub>h</sub> Index 1401 <sub>h</sub> : Ff <sub>h</sub> Index 1402 <sub>h</sub> : Ff <sub>h</sub> Index 1403 <sub>h</sub> : Ff <sub>h</sub> Index 1405 <sub>h</sub> : Ff <sub>h</sub>

### 3.1.19 Objects 1600<sub>h</sub>-1605<sub>h</sub>: Receive PDO Mapping

This object contains the mapping for the PDOs the device is able to receive. The sub-index 0<sub>h</sub> 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/received with the corresponding PDO. The sub-indices from 1<sub>h</sub> 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. The values are hexadecimally coded.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
1600 <sub>h</sub> -1605 <sub>h</sub>	Receive PDO mapping	RECORD	PDO Mapping
1600 <sub>h</sub>	Mapping for RPDO 1	RECORD	PDO Mapping
1601 <sub>h</sub>	Mapping for RPDO 2	RECORD	PDO Mapping
1602 <sub>h</sub>	Mapping for RPDO 3	RECORD	PDO Mapping
1603 <sub>h</sub>	Mapping for RPDO 4	RECORD	PDO Mapping
1605 <sub>h</sub>	Mapping for RPDO 6	RECORD	PDO Mapping

#### ENTRY DESCRIPTION

Sub-Index	Description	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	Number of mapped application objects in PDO	rw	no	1... 3 0 deactivated	Index 1600 <sub>h</sub> : 1 Index 1601 <sub>h</sub> : 2 Index 1602 <sub>h</sub> : 2 Index 1603 <sub>h</sub> : 2 Index 1605 <sub>h</sub> : 2
01 <sub>h</sub>	Mapping entry 1	rw	no	UNSIGNED32	Index 1600 <sub>h</sub> : 60400010 <sub>h</sub> Index 1601 <sub>h</sub> : 60400010 <sub>h</sub> Index 1602 <sub>h</sub> : 60400010 <sub>h</sub> Index 1603 <sub>h</sub> : 60400010 <sub>h</sub> Index 1605 <sub>h</sub> : 60400010 <sub>h</sub>

Sub-Index	Description	Access	PDO Mapping	Value Range	Default Value
02 <sub>h</sub>	Mapping entry 2	rw	no	UNSIGNED32	Index 1600 <sub>h</sub> : 0 Index 1601 <sub>h</sub> : 60600008 <sub>h</sub> Index 1602 <sub>h</sub> : 607A0020 <sub>h</sub> Index 1603 <sub>h</sub> : 60FF0020 <sub>h</sub> Index 1605 <sub>h</sub> : 60420010 <sub>h</sub>
03 <sub>h</sub>	Mapping entry 3	rw	no	UNSIGNED32	Index 1600 <sub>h</sub> : 0 Index 1601 <sub>h</sub> : 0 Index 1602 <sub>h</sub> : 0 Index 1603 <sub>h</sub> : 0 Index 1605 <sub>h</sub> : 0

### 3.1.20 Objects 1800<sub>h</sub>-1805<sub>h</sub>: Transmit PDO Communication Parameter

These objects contain the communication parameters for the PDOs the device is able to transmit.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
1800 <sub>h</sub> -1805 <sub>h</sub>	Transmit PDO communication parameter	RECORD	PDO CommPar
1800 <sub>h</sub>	TPDO 1 transmit communication parameter	RECORD	PDO CommPar
1801 <sub>h</sub>	TPDO 2 transmit communication parameter	RECORD	PDO CommPar
1802 <sub>h</sub>	TPDO 3 transmit communication parameter	RECORD	PDO CommPar
1803 <sub>h</sub>	TPDO 4 transmit communication parameter	RECORD	PDO CommPar
1805 <sub>h</sub>	TPDO 6 transmit communication parameter	RECORD	PDO CommPar

#### ENTRY DESCRIPTION

Sub-Index	Description	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	Largest sub-index supported	ro	no	2... 5 Number of entries: 5	5
01 <sub>h</sub>	TPDO 1-4, 6: COB-ID	rw	no	UNSIGNED32	Index 1800 <sub>h</sub> : 180 <sub>h</sub> + Node-ID Index 1801 <sub>h</sub> : 280 <sub>h</sub> + Node-ID Index 1802 <sub>h</sub> : 380 <sub>h</sub> + Node-ID Index 1803 <sub>h</sub> : 480 <sub>h</sub> + Node-ID Index 1805 <sub>h</sub> : 0
02 <sub>h</sub>	Transmission type	rw	no	UNSIGNED8	Index 1800 <sub>h</sub> : ff <sub>h</sub> Index 1801 <sub>h</sub> : ff <sub>h</sub> Index 1802 <sub>h</sub> : 1 Index 1803 <sub>h</sub> : 1 Index 1805 <sub>h</sub> : 1
03 <sub>h</sub>	Inhibit time	rw	no	UNSIGNED16	0
04 <sub>h</sub>	Compatibility entry	ro	no	UNSIGNED8	0
05 <sub>h</sub>	Event timer	rw	no	0 not used UNSIGNED16	0

### 3.1.21 Objects 1A00<sub>h</sub>-1A05<sub>h</sub>: Transmit PDO Mapping Parameter

These objects contain the mapping for the PDOs the device is able to transmit.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
1A00 <sub>h</sub> -1A05 <sub>h</sub>	Transmit PDO mapping parameter	RECORD	PDO mapping
1A00 <sub>h</sub>	TPDO 1: transmit PDO mapping parameter	RECORD	PDO mapping
1A01 <sub>h</sub>	TPDO 2: transmit PDO mapping parameter	RECORD	PDO mapping
1A02 <sub>h</sub>	TPDO 3: transmit PDO mapping parameter	RECORD	PDO mapping
1A03 <sub>h</sub>	TPDO 4: transmit PDO mapping parameter	RECORD	PDO mapping
1A05 <sub>h</sub>	TPDO 6: transmit PDO mapping parameter	RECORD	PDO mapping

#### ENTRY DESCRIPTION

Sub-Index	Description	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	Number of mapped application objects in PDO	rw	no	0 deactivated Number of entries: 1... 3	Index 1A00 <sub>h</sub> : 1 Index 1A01 <sub>h</sub> : 2 Index 1A02 <sub>h</sub> : 2 Index 1A03 <sub>h</sub> : 2 Index 1A05 <sub>h</sub> : 2
01 <sub>h</sub>	Mapping entry 1	rw	no	UNSIGNED32	Index 1A00 <sub>h</sub> : 60410010 <sub>h</sub> Index 1A01 <sub>h</sub> : 60410010 <sub>h</sub> Index 1A02 <sub>h</sub> : 60410010 <sub>h</sub> Index 1A03 <sub>h</sub> : 60410010 <sub>h</sub> Index 1A05 <sub>h</sub> : 60410010 <sub>h</sub>
02 <sub>h</sub>	Mapping entry 2	rw	no	UNSIGNED32	Index 1A00 <sub>h</sub> : 0 Index 1A01 <sub>h</sub> : 60610008 <sub>h</sub> Index 1A02 <sub>h</sub> : 60640020 <sub>h</sub> Index 1A03 <sub>h</sub> : 606c0020 <sub>h</sub> Index 1A05 <sub>h</sub> : 60440010 <sub>h</sub>
03 <sub>h</sub>	Mapping entry 3	rw	no	UNSIGNED32	Index 1A00 <sub>h</sub> : 0 Index 1A01 <sub>h</sub> : 0 Index 1A02 <sub>h</sub> : 0 Index 1A03 <sub>h</sub> : 0 Index 1A05 <sub>h</sub> : 0

## 4 Device Profile Objects (CiA402) and Modes of Operation

The PDS (power drive system) behavior depends on the activated mode of operation. The PDS implements several modes of operation. 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 OPERATION MODES CAN BE CHOSEN:

- Profile position mode
- Homing mode
- Velocity mode
- Profile velocity mode

Please refer to object 6060<sub>h</sub> (section 4.1.6) for information about how to choose an operation mode.

## 4.1 Detailed Object Specifications

### 4.1.1 Object 605A<sub>h</sub>: 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 operations. 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>

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
605A <sub>h</sub>	Quick stop option code	Variable	SIGNED16

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	Refer to CiA402-3	See value definition above	2

### 4.1.2 Object 605B<sub>h</sub>: 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 <i>drive function</i> (switch-off the drive power stage)

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
605B <sub>h</sub>	Shutdown option code	Variable	UNSIGNED16

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	r0	Refer to CiA402-3	See value definition above	0

### 4.1.3 Object 605C<sub>h</sub>: 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 with <i>slow down ramp</i> ; disable of the <i>drive function</i>

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
605C <sub>h</sub>	Disable operation option code	Variable	UNSIGNED16

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	r0	Refer to CiA402-3	See value definition above	1

### 4.1.4 Object 605D<sub>h</sub>: 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. The halt option code always has the value 1 as only this is supported.

#### VALUE DEFINITION

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

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
605D <sub>h</sub>	Halt option code	Variable	UNSIGNED16

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	r0	Refer to CiA402-3	See value definition above	1

### 4.1.5 Object 605E<sub>h</sub>: 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 <i>quick stop ramp</i>

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
605E <sub>h</sub> (motor 0)	Fault reaction option code	Variable	UNSIGNED16

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	r0	Refer to CiA402-3	See value definition above	2

### 4.1.6 Object 6060<sub>h</sub>: Modes of Operation

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

#### VALUE DEFINITION

Value	Definition
0	No mode*
1	Profile position mode
2	Velocity mode
3	Profile velocity mode
6	Homing mode

\* 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 Code	Data Type
6060 <sub>h</sub>	Modes of operation	Variable	SIGNED8

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	Refer to CiA402-3	See value definition above	0

### 4.1.7 Object 6061<sub>h</sub>: Modes of Operation Display

This object shows the operating mode that is set.

#### VALUE DEFINITION

Value	Definition
0	No mode*
1	Profile position mode
2	Velocity mode
3	Profile velocity mode
6	Homing mode

\* 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 Code	Data Type
6061 <sub>h</sub>	Modes of operation display	Variable	SIGNED8

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	ro	Refer to CiA402-3	See value definition above	0

### 4.1.8 Object 606A<sub>h</sub>: Sensor Selection Code

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

#### VALUE DEFINITION

Value	Definition
0	Encoder used
-1	No encoder

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
606A <sub>h</sub>	Sensor selection code	Variable	SIGNED16

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	Refer to CiA402-3	See value definition above	-1

### 4.1.9 Object 6085<sub>h</sub>: 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<sub>h</sub> is set to 2 [or 6]. The value is given in the same unit as *profile acceleration object* 6083<sub>h</sub>.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6085 <sub>h</sub>	Quick stop deceleration	Variable	UNSIGNED32

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	Refer to CiA402-3	UNSIGNED32	1000

### 4.1.10 Object 608F<sub>h</sub>: Position Encoder Resolution

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

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

All values are dimensionless.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
608F <sub>h</sub>	Encoder resolution	Array	UNSIGNED32

#### ENTRY DESCRIPTION

Sub-Index	Description	Entry category	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	Highest sub index supported	Mandatory	c	Refer to CiA402	02 <sub>h</sub>	2
01 <sub>h</sub>	Encoder increments	Mandatory	rw	Refer to CiA402	UNSIGNED32	1
02 <sub>h</sub>	Motor revolutions	Mandatory	ro	Refer to CiA402	UNSIGNED32	2

### 4.1.11 Object 60C5<sub>h</sub>: Max Acceleration

This object indicates the configured maximum acceleration. It is used to limit the acceleration to an acceptable value in order to prevent the motor and the moved mechanics from being destroyed. It is given in internal or user specific units (depending on object 208C<sub>h</sub>, paragraph 0, object 2087<sub>h</sub> and object 2088<sub>h</sub>, paragraph 0 and paragraph 5.2.14)

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
60C5 <sub>h</sub>	Max acceleration	Variable	UNSIGNED32

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	ro	Refer to CiA402-3	UNSIGNED32	Depends on the units

### 4.1.12 Object 60C6<sub>h</sub>: Max Deceleration

This object indicates the configured maximal deceleration. It is used to limit the deceleration to an acceptable value in order to prevent the motor and the moved mechanics from being destroyed. The max deceleration is given in internal or user specific units (depending on object 208C<sub>h</sub>, paragraph 0, object 2087<sub>h</sub> and object 2088<sub>h</sub>, paragraph 0 and paragraph 5.2.14). Object 60C6<sub>h</sub> has the same unit as the max acceleration object 60C5<sub>h</sub>.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
60C6 <sub>h</sub>	Max deceleration	Variable	UNSIGNED32

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	ro	Refer to CiA402-3	UNSIGNED32	Depends on the units

### 4.1.13 Object 60FD<sub>h</sub>: 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.

#### VALUE DEFINITION

Value	Definition
0 <sub>b</sub>	Switched off
1 <sub>b</sub>	Switched on

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
60FD <sub>h</sub>	Digital inputs	Variable	UNSIGNED32

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	r0	Refer to CiA402-3	UNSIGNED32	0000 0000 <sub>h</sub>

#### 4.1.14 Object 6502<sub>h</sub>: Supported Drive Modes

This object provides information on the supported drive modes and contains always the *value* 00000025<sub>h</sub>, which means that the following modes are provided by the drive:

- 0 = no mode
- 1 = profile position mode
- 2 = velocity mode
- 3 = profile velocity mode
- 6 = homing mode

##### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6502 <sub>h</sub>	Supported drive modes	Variable	UNSIGNED32

##### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	r0	Refer to CiA402-3	See above	00000025 <sub>h</sub>

## 4.2 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<sub>h</sub> (section 4.1.6) for information about how to choose an operation mode. Object 6061<sub>h</sub> (section 4.1.7) shows the operation mode that is set.

### 4.2.1 Detailed Object Specifications

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

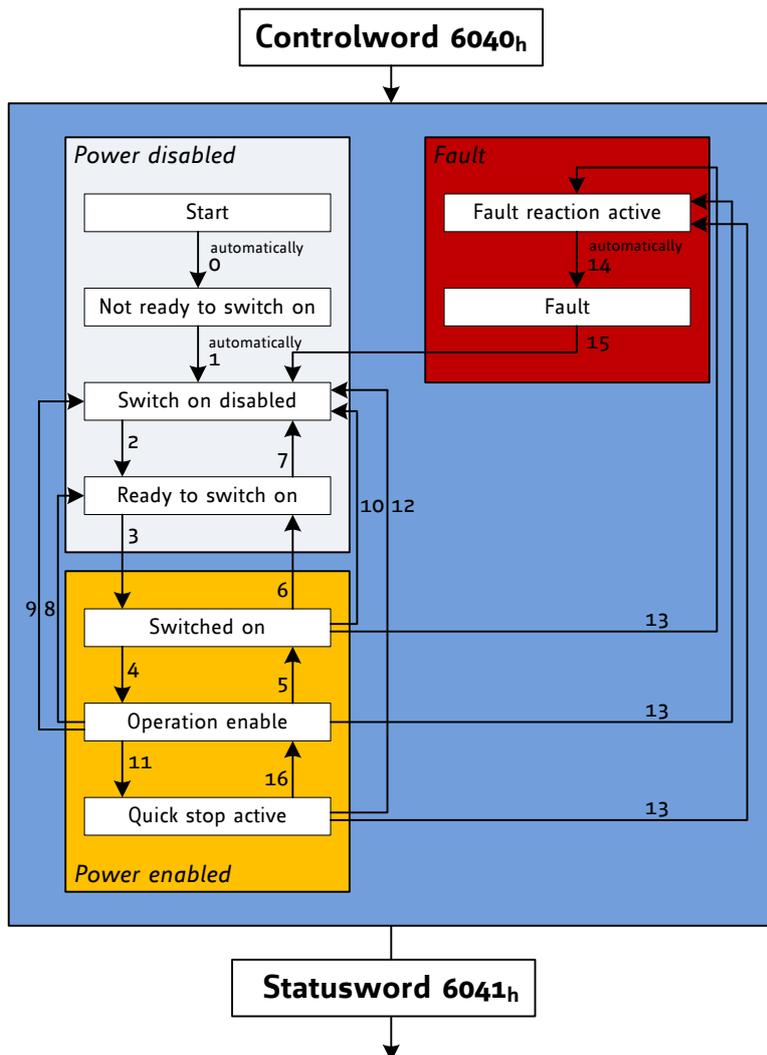


Figure 4.1 Finite state machine

#### 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.

### 4.2.1.1 Object 6040<sub>h</sub>: Controlword

This object indicates the received command controlling the power drive system finite state automation (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to *Figure 4.1 Finite state machine* for detailed information.

#### STRUCTURE OF CONTROLWORD / VALUE DEFINITION

15	11	10	9	8	7	6	4	3	2	1	0	
<i>nu</i>			<i>r</i>	<i>oms</i>	<i>h</i>	<i>fr</i>	<i>oms</i>		<i>eo</i>	<i>qs</i>	<i>ev</i>	<i>so</i>

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

#### OPERATION MODE SPECIFIC BITS IN PP MODE

Bit	Name	Definition
4	New set point	0-to-1: The next positioning will be started after the last one has completed.
5	Change immediately	Not supported
6	Absolute/relative	0 = new position is absolute; 1 = new position is relative
9	Change set point	Not supported

#### COMMAND CODING

Command	Bits of controlword					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 (NOTE)
Disable voltage	0	x	x	0	x	7, 9, 10, 12
Quick stop	0	x	0	1	x	7, 10, 11
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4, 16
Fault reset	↑	x	x	x	x	15

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6040 <sub>h</sub>	Controlword	Variable	UNSIGNED16

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	Refer to CiA402-3	See command coding above	Device and operation mode specific

### 4.2.1.2 Object 6041<sub>h</sub>: Statusword

This object provides the status of the PDS FSA. It reflects the status of the CiA402 state machine. Please refer to *Figure 4.1 Finite state machine* for detailed information. The object is structured as defined below.

For more information about the coding refer to the *CANopen Drives and motion control device profile, part 2* please.

#### STRUCTURE OF STATUSWORD / VALUE DEFINITION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<i>dir</i>	<i>mot</i>	<i>oms</i>	<i>ila</i>	<i>tr</i>	<i>rm</i>	<i>ms</i>	<i>w</i>	<i>sod</i>	<i>qs</i>	<i>ve</i>	<i>f</i>	<i>oe</i>	<i>so</i>	<i>rtso</i>	

MSB

LSB

**LEGEND:** *dir* = direction; *mot* = motor activity; *oms* = operation mode specific; *ila* = internal limit active; *tr* = target reached; *rm* = remote; *ms* = manufacturer specific; *w* = warning; *sod* = switch on disabled; *qs* = quick stop; *ve* = voltage enabled; *f* = fault; *oe* = operation enabled; *so* = switched on; *rtso* = ready to switch on

#### TRINAMIC SPECIFIC BITS

Bit	Name	Definition
14	Motor activity	0 = motor stop; 1 = motor rotates
15	Direction of rotation	This bit determines/shows the direction of rotation

#### OPERATION MODE SPECIFIC BITS IN PP MODE

Bit	Name	Definition
10	Target reached	Set when the motor is within the target position window
12	Set point acknowledge	0 = set point processed; 1 = set point still in process
13	Following error	Not supported

#### STATE CODING

Statusword	PDS FSA state
xxxx xxxx x0xx 0000 <sub>b</sub>	Not ready to switch on
xxxx xxxx x1xx 0000 <sub>b</sub>	Switch on disabled
xxxx xxxx x01x 0001 <sub>b</sub>	Ready to switch on
xxxx xxxx x01x 0011 <sub>b</sub>	Switched on
xxxx xxxx x01x 0111 <sub>b</sub>	Operation enabled
xxxx xxxx x00x 0111 <sub>b</sub>	Quick stop active
xxxx xxxx x0xx 1111 <sub>b</sub>	Fault reaction active
xxxx xxxx x0xx 1000 <sub>b</sub>	Fault

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6041 <sub>h</sub>	Statusword	Variable	UNSIGNED16

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	ro	Refer to CiA402-3	See state coding above	no

### 4.2.1.3 Object 6062<sub>h</sub>: Position Demand Value

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

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6062 <sub>h</sub>	Position demand value	Variable	SIGNED32

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	ro	Refer to CiA402-3	SIGNED32	No

### 4.2.1.4 Object 6063<sub>h</sub>: Position Actual Internal Value

This object provides the actual value of the encoder or the motor. Please use the sensor selection object 606A<sub>h</sub> (4.1.8) for selecting the motor or the encoder first. Object 6063<sub>h</sub> 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 Code	Data Type
6063 <sub>h</sub>	Position actual internal value	Variable	SIGNED32

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	ro	Refer to CiA402-3	SIGNED32	No

### 4.2.1.5 Object 6064<sub>h</sub>: Position Actual Value

This object provides the actual value of the position measurement device. It always contains the same value as object 6063<sub>h</sub>.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6064 <sub>h</sub>	Position actual value	Variable	SIGNED32

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	ro	Refer to CiA402-3	SIGNED32	No

### 4.2.1.6 Object 6065<sub>h</sub>: 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 in microsteps. If the value of the following error window is 0 or FFFF FFFF<sub>h</sub>, the following control will be switched off.

When the difference between motor position (object 6062<sub>h</sub>) and encoder position (object 6063<sub>h</sub> or 6064<sub>h</sub>) 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.

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

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6065 <sub>h</sub>	Following error window	Variable	UNSIGNED32

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	Refer to CiA402-3	UNSIGNED32	0... 8388607

### 4.2.1.7 Object 6067<sub>h</sub>: 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 FFFF FFFF<sub>h</sub>, 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<sub>h</sub>) has reached the target position (607A<sub>h</sub>).

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<sub>h</sub>) is within the target position - position window and target position + position window.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6067 <sub>h</sub>	Position window	Variable	UNSIGNED32

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	Refer to CiA402-3	UNSIGNED32	0... 8388607

### 4.2.1.8 Object 6068<sub>h</sub>: 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<sub>h</sub>) is set to a value greater than zero the target reached event will not be signaled until the actual position (6064<sub>h</sub>) is at least as many milliseconds within the position window as defined by this object.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6068 <sub>h</sub>	Position window time	Variable	UNSIGNED16

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	Refer to CiA402-3	UNSIGNED16	0

#### 4.2.1.9 Object 606C<sub>h</sub>: Velocity Actual Value

This object gives the actual velocity value derived either from the velocity sensor or from the position sensor.

##### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
606C <sub>h</sub>	Velocity actual value	Variable	INTEGER32

##### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	ro	Refer to CiA402-3	INTEGER32	no

#### 4.2.1.10 Object 607A<sub>h</sub>: 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 Code	Data Type
607A <sub>h</sub>	Target position	Variable	SIGNED32

##### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	Refer to CiA402-3	SIGNED32	-2147483647... +2147483647

#### 4.2.1.11 Object 607D<sub>h</sub>: 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

The limit positions are given in microsteps (same as target position). Two limits for moving the motor in positioning mode can be set here. It will then not be possible to leave this window.

##### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
607D <sub>h</sub>	Software position limit	Array	SIGNED32

##### ENTRY DESCRIPTION

Sub-Index	Description	Entry category	Access	PDO Mapping	Value Range	Default Value
01 <sub>h</sub>	Min position limit	Mandatory	rw	Refer to CiA402-3	SIGNED32	-2147483647
02 <sub>h</sub>	Max position limit	Mandatory	rw	Refer to CiA402-3	SIGNED32	-2147483647

#### 4.2.1.12 Object 607F<sub>h</sub>: Maximum Profile Velocity

This object indicates the configured maximal allowed velocity in either direction during a profiled motion. It is the maximum velocity that can be used for positioning. The value is given in internal or user specific units (depending on object 208C<sub>h</sub>, paragraph 0).

##### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
607F <sub>h</sub>	Maximum profile velocity	Variable	UNSIGNED32

##### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	ro	Refer to CiA402-3	UNSIGNED32	2047

#### 4.2.1.13 Object 6081<sub>h</sub>: 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<sub>h</sub>, paragraph 0)

##### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6081 <sub>h</sub>	Profile velocity	Variable	UNSIGNED32

##### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	Refer to CiA402-3	UNSIGNED32	1000

#### 4.2.1.14 Object 6083<sub>h</sub>: Profile Acceleration

This object indicates the configured acceleration. Object 6083<sub>h</sub> sets the maximum acceleration to be used in profile positioning mode.

Please choose the units for object 6083<sub>h</sub> with object 208E<sub>h</sub>, described in paragraph 0. Object 208E<sub>h</sub> reads 0 when internal units are selected or 179 when user units (PPS/s) are selected.

##### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6083 <sub>h</sub>	Profile acceleration	Variable	UNSIGNED32

##### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	Refer to CiA402-3	UNSIGNED32	1000

### 4.2.1.15 Object 6084<sub>h</sub>: Profile Deceleration

This object indicates the configured deceleration. It sets the maximum deceleration used in profile positioning mode.

Please choose the units for object 6084<sub>h</sub> with object 208E<sub>h</sub>, described in paragraph 0. Object 208E<sub>h</sub> reads 0 when internal units are selected or 179 when user units (PPS/s) are selected.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6084 <sub>h</sub>	Profile deceleration	Variable	UNSIGNED32

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	Refer to CiA402-3	UNSIGNED32	1000

## 4.2.2 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 start) 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<sub>h</sub>.
- Select pp mode by writing 1 to object 6060<sub>h</sub>.
- Write 6 to object 6040<sub>h</sub> to switch to READY\_TO\_SWITCH\_ON state.
- Write 7 to object 6040<sub>h</sub> to switch to SWITCHED\_ON state.
- Write 15 to object 6040<sub>h</sub> to switch to OPERATION\_ENABLED state.
- Write the desired target position (e.g. 500000) to object 607A<sub>h</sub>.
- Mark the new target position as active by writing 31 to object 6040<sub>h</sub>. The motor starts moving now.
- Reset the activation by writing 15 to object 6040<sub>h</sub> (this can be done while the motor is still moving).

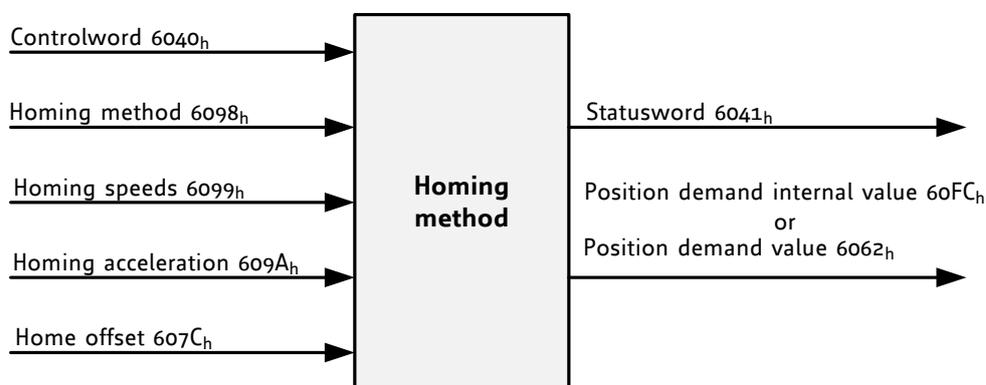
### 4.3 Homing Mode

This clause 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 4.2 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 zero in the coordinate system for the home position.



**Figure 4.2 Homing mode function**

**BY CHOOSING A HOMING METHOD, THE FOLLOWING IS DETERMINED:**

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

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

**THERE ARE FOUR SOURCES OF HOMING SIGNAL AVAILABLE:**

- negative and positive homing 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. There are several application specific methods. The homing method is used for selection.

Please refer to object  $6060_h$  (section 4.1.6) for information about how to choose an operation mode. Object  $6061_h$  (section 4.1.7) shows the operation mode that is set.

### 4.3.1 Connecting Home Switch, Left Switch, and Right Switch

For detailed information refer to the specific hardware manual of your module!

#### 4.3.1.1 TCM-1140

A 2mm pitch 8 pin JST B8B-PH-K connector is used for connecting general purpose inputs, outputs, home switch, and stop switches to the unit. If desired it is possible to connect a brake, too. Therefore, OUT\_0 can be used.

Mating connector housing: PHR-8

Mating connector contacts: SPH-002T-P0.5S

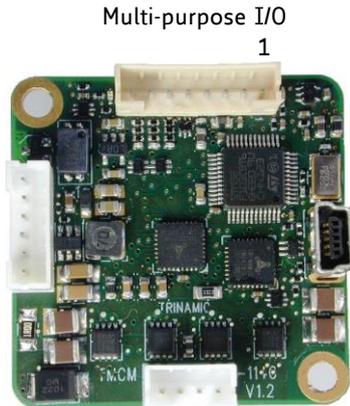


Figure 4.3 Multi-purpose I/O connector of TCM-1140

#### INPUT / STOP / HOME SWITCH CONNECTOR

	Pin	Label	Description
	1	GND	System and signal ground
	2	VDD	VDD, connected to VDD pin of the power and communication connector
	3	OUT_0	Open-drain output (max. 1A) Integrated freewheeling diode to VDD <i>Connect brake here if desired.</i>
	4	OUT_1	+5V supply output (max. 100mA) Can be switched on/off in software
	5	IN_0	Dedicated analog input, Input voltage range: 0..+10V Resolution: 12bit (0..4095)
	6	IN_1, STOP_L, ENC_A	General purpose digital input (+24V compatible) Alternate function 1: left stop switch input Alternate function 2: external incremental encoder channel A input
	7	IN_2, STOP_R, ENC_B	General purpose digital input (+24V compatible) Alternate function 1: right stop switch input Alternate function 2: external incremental encoder channel B input
	8	IN_3, HOME, ENC_N	General purpose digital input (+24V compatible) Alternate function 1: home switch input Alternate function 2: external incremental encoder index / zero channel input

### 4.3.1.2 TCM1160

A 2mm pitch 8-pin JST B8B-PH-K connector is used for connecting general purpose inputs, home switch, stop switches, and outputs (e.g. a brake) to the unit.

Mating connector housing: PHR-8

Mating connector contacts: SPH-002T-P0.5S

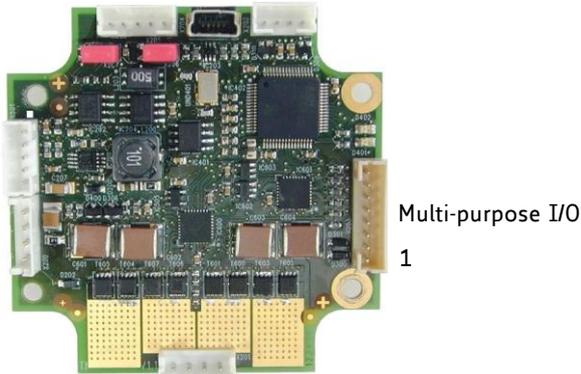


Figure 4.4 Multi-purpose I/O of TCM1160

	Pin	Label	Description
	1	OUT_0	General purpose output, open drain (max. 1A) Integrated freewheeling diode connected to +V <sub>Logic</sub>
	2	OUT_1	General purpose output, open drain (max. 1A) Integrated freewheeling diode connected to +V <sub>Logic</sub>
	3	IN_0	General purpose input (analog and digital), +24V compatible Resolution when used as analog input: 12bit (0... 4095)
	4	IN_1	General purpose input (analog and digital), +24V compatible Resolution when used as analog input: 12bit (0... 4095)
	5	STOP_L	Left stop switch input (digital input), +24V compatible, internal pull-up to +5V
	6	STOP_R	Right stop switch input (digital input), +24V compatible, internal pull-up to +5V
	7	HOME	Home switch input (digital input), +24V compatible, internal pull-up to +5V
	8	GND	Module ground (system and signal ground)

#### NOTE

All inputs have resistor based voltage dividers with protection diodes. These resistors also ensure a valid GND level when left unconnected.

#### 4.3.1.2.1 Digital Inputs STOP\_L, STOP\_R, and HOME

The eight pin connector of the TCMC-1160 provides three reference switch digital inputs STOP\_L, STOP\_R and HOME. All three inputs accept up to +24 V input signals. They are protected against these higher voltages using voltage resistor dividers together with limiting diodes against voltages below 0 V (GND) and above +3.3 V DC.

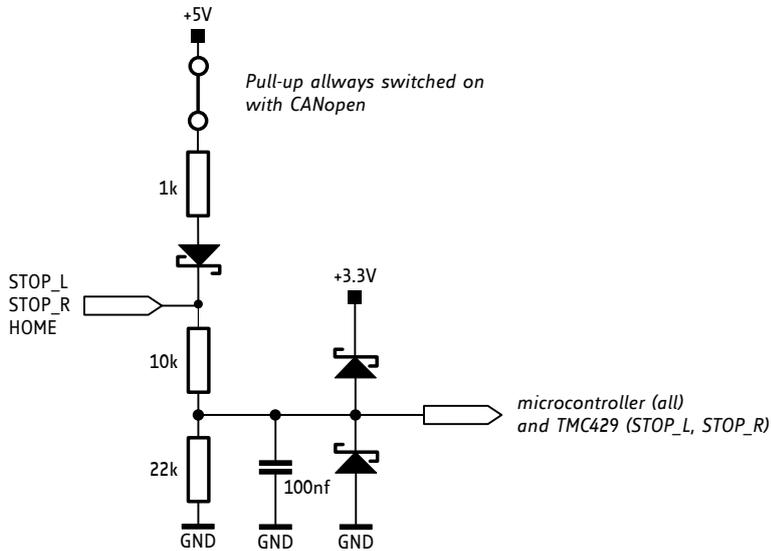


Figure 4.6 STOP\_L, STOP\_R and HOME inputs (simplified input circuit)

All three digital inputs are connected to the on-board processor and can be used as general purpose digital inputs!

### 4.3.1.3 TCM1180

First of all it is necessary to connect the switches. A 2mm pitch 6 pin JST B6B-PH-K connector is used for connecting general purpose inputs, home and stop switches to the unit.

Mating connector housing: PHR-6

Mating connector contacts: SPH-002T-P0.5S

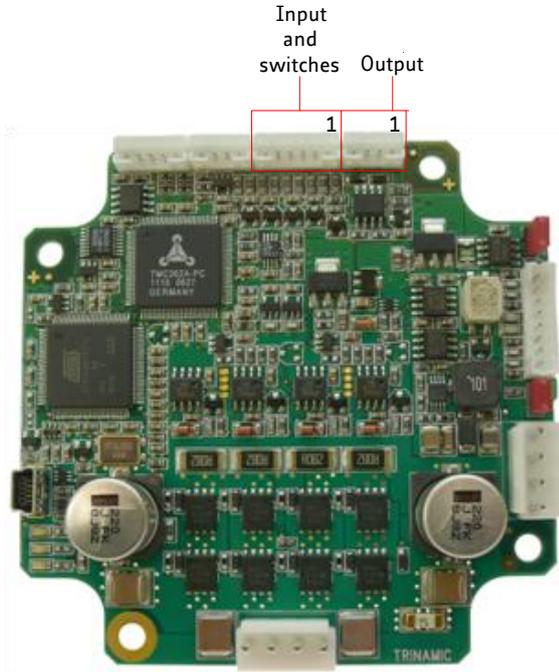


Figure 4.5 I/O connectors of the TCM1180

#### INPUT / STOP / HOME SWITCH CONNECTOR

	Pin	Label	Description
	1	IN_0	General purpose input, +24V compatible
	2	IN_1	General purpose input, +24V compatible
	3	STOP_L	Left stop switch input, +24V compatible, internal pull-up (1k to +5V)
	4	STOP_R	Right stop switch input, +24V compatible, internal pull-up (1k to +5V)
	5	HOME	Home switch input, +24V compatible, internal pull-up (1k to +5V)
	6	GND	Module ground (system and signal ground)

Table 4.1 Input / Stop / Home switch connector

#### 4.3.1.3.1 Connect a Brake

The module provides the opportunity to connect a brake. Please take OUT\_1 for it. A 2mm pitch 4 pin JST B4B-PH-K connector is used.

Mating connector housing: PHR-4

Mating connector contacts: SPH-002T-P0.5S

##### OUTPUT CONNECTOR

	Pin	Label	Description
	1	+U <sub>Logic</sub>	Module digital logic power supply – connected to pin 2 of power supply connector
	2	/Shutdown	/Shutdown input – has to be connected to power supply (e.g. pin 1 of this connector) in order to enable driver. Connecting this input to ground or leaving it unconnected will disable driver stage
	3	OUT_0	Open collector output with integrated freewheeling diode, +24V compatible
	4	OUT_1	<i>Connect brake here.</i>

## 4.3.2 Homing Methods

There are several different methods of homing. For choosing your homing method, refer to object 6098<sub>h</sub>, please.

### OVERVIEW: HOMING METHODS

Method no.	Description
0	No homing. This is the default setting.
1	Search the left end switch, than search the next encoder index pulse.
2	Search the right end switch, than search the next encoder index pulse.
3	Search the positive edge of the home switch, than search the next encoder index pulse.
5	Search the negative edge of the home switch, than 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 6062 <sub>h</sub> , 6063 <sub>h</sub> , and 6064 <sub>h</sub> ) are set to zero, but the motor will not move.

#### 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.

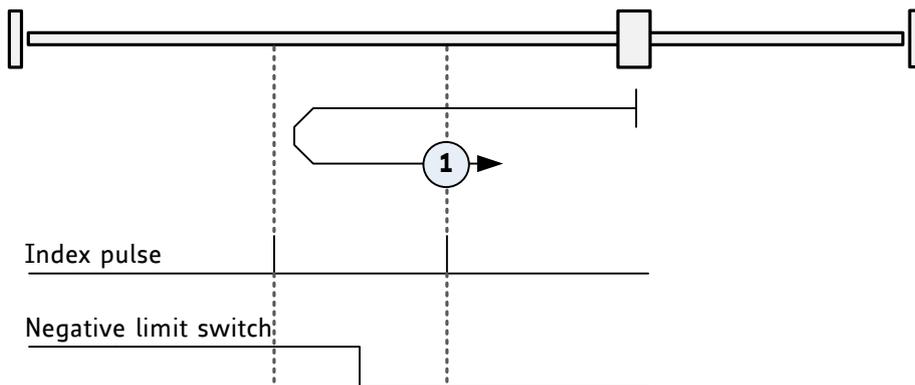


Figure 4.6 Homing on negative switch and index pulse

#### 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.

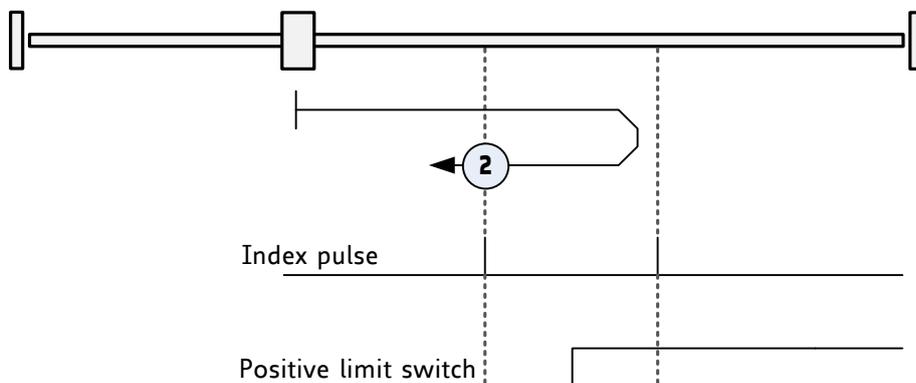
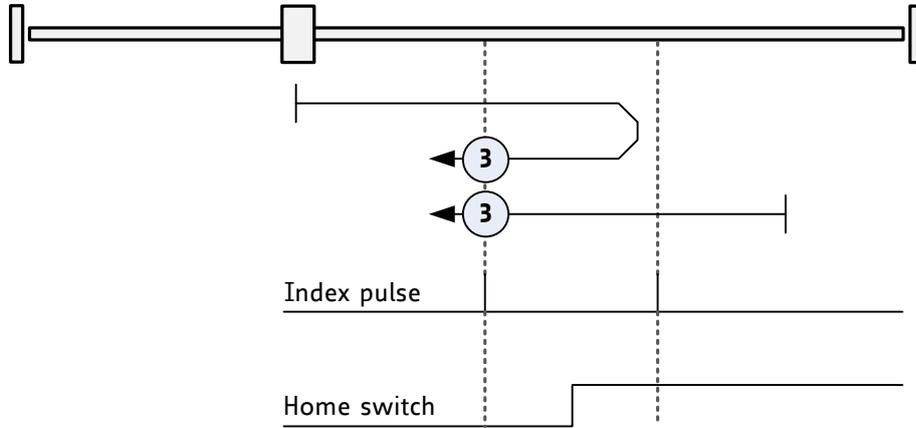


Figure 4.7 Homing on positive limit switch and index pulse

**METHOD 3: HOMING ON POSITIVE HOME SWITCH AND INDEX PULSE**

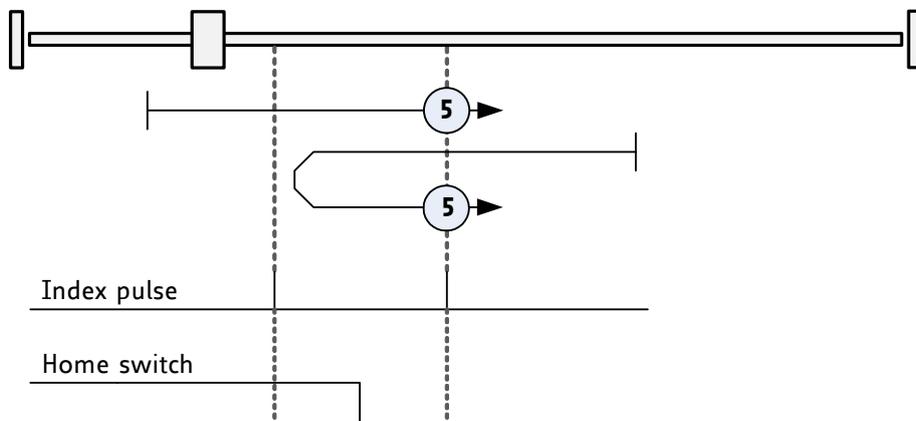
Using this method, the initial direction of movement shall be dependent on the state of the home switch. The home position shall be at the index pulse to either to the left or the right of the point where the home switch changes state. If the initial position is situated so that the direction of movement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



**Figure 4.8 Homing on positive home switch and index pulse**

**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 4.9 Homing on negative home switch and index pulse**

**METHOD 17, 18, 19, AND 21: HOMING WITHOUT INDEX PULSE**

These methods are similar to methods 1 to 5 except that the home position is not dependent on the index pulse but only dependent on the relevant home or limit switch transitions. Method 19 (similar to method 3) is shown in Figure 4.10.

Method no.	Description
17	Search the left end switch. (Similar to method 1)
18	Search the right end switch. (Similar to method 2)
19	Search the positive edge of the home switch. (Similar to method 3)
21	Search the negative edge of the home switch. (Similar to method 5.)

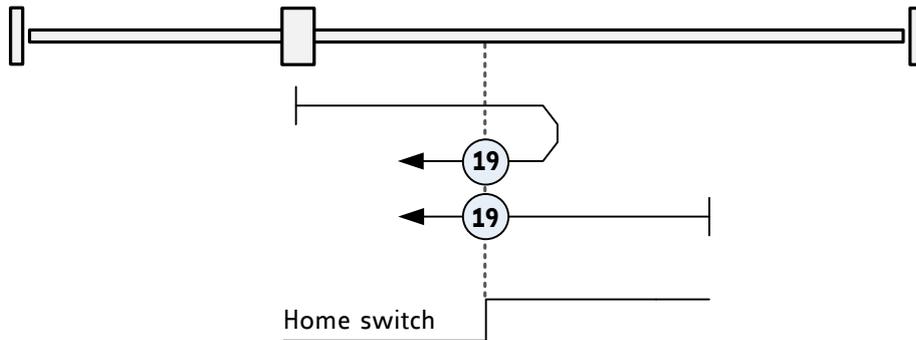


Figure 4.10 Homing without index pulse

**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 as shown in Figure 4.11.

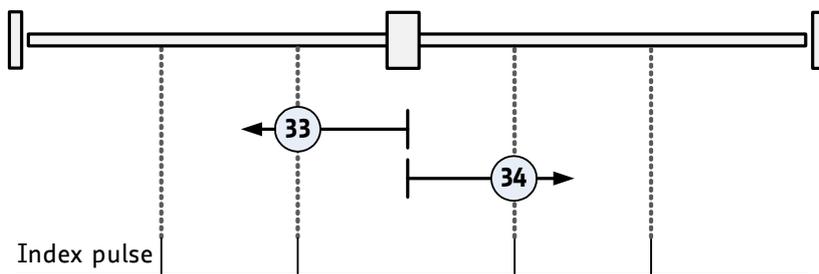


Figure 4.11 Homing on index pulse

**METHOD 35: HOMING ON INDEX PULSE / CURRENT POSITION AS HOME POSITION**

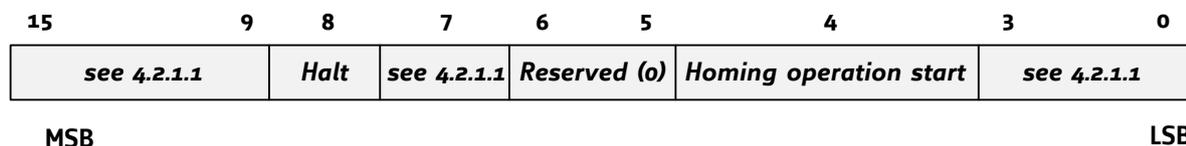
In this method, the current position shall be taken to be the home position. This method does not require the drive device to be in *operational enabled* state.

### 4.3.3 Detailed Object Definitions

#### 4.3.3.1 Object 6040<sub>h</sub>: Controlword

This object indicates the received command controlling the power drive system finite state automation (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to *Figure 4.1 Finite state machine* for detailed information.

##### STRUCTURE OF CONTROLWORD / VALUE DEFINITION



##### OPERATION MODE SPECIFIC BITS IN HM MODE

Bit	Name	Definition
4	Homing operation start	Set to 1 to start homing; setting to 0 stops homing
8	Halt	Not supported

##### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6040 <sub>h</sub>	Controlword	Variable	UNSIGNED16

##### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	Refer to CiA402-3	See command coding above	Device and operation mode specific

### 4.3.3.2 Object 6041<sub>h</sub>: Statusword

This object provides the status of the PDS FSA. It reflects the status of the CiA402 state machine. Please refer to *Figure 4.1 Finite state machine* for detailed information. The object is structured as defined below.

For more information about the coding refer to the *CANopen Drives and motion control device profile, part 2* please.

#### STRUCTURE OF STATUSWORD / VALUE DEFINITION

15	14	13	12	11	10	9	0
<i>see 4.2.1.2</i>	<b>Homing error</b>	<b>Homing attained</b>	<i>see 4.2.1.2</i>	<b>Target reached</b>	<i>see 4.2.1.2</i>		
<b>MSB</b>				<b>LSB</b>			

#### 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	Homing attained	Set when the zero position has been found
13	Homing error	Not supported

#### DEFINITION OF BIT 10, BIT 12, AND BIT 13

Bit 13	Bit 12	Bit 10	Definition
0	0	0	Homing procedure is in progress
0	0	1	Homing procedure is interrupted or not reached
0	1	0	Homing is attained, but target is not reached
0	1	1	Homing procedure is completed successfully
1	0	0	Homing error occurred, velocity is not 0
1	0	1	Homing error occurred, velocity is 0
1	1	x	reserved

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6041 <sub>h</sub>	Statusword	Variable	UNSIGNED16

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	r0	Refer to CiA402-3	See state coding above	No

### 4.3.3.3 Object 606C<sub>h</sub>: Velocity Actual Value

This object gives the actual velocity value derived either from the velocity sensor or from the position sensor.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
606C <sub>h</sub>	Velocity actual value	Variable	INTEGER32

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	ro	Refer to CiA402-3	INTEGER32	no

### 4.3.3.4 Object 607C<sub>h</sub>: 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.

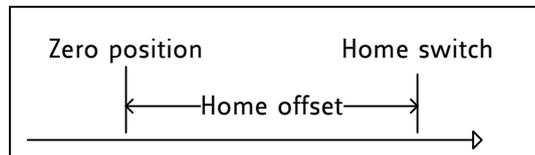


Figure 4.12 Home offset definition

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
607C <sub>h</sub>	Home offset	Variable	SIGNED32

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	Refer to CiA402-3	SIGNED32	-2147483647... +2147483647

### 4.3.3.5 Object 6098<sub>h</sub>: Homing Method

The homing method to be used in homing mode can be selected here. The actual firmware supports the following homing methods.

#### VALUE DEFINITION

Value	Definition
0	No homing. This is the default value, but there is actually no homing method 0.
1	Search the left end switch, than search the next encoder index pulse.
2	Search the right end switch, than search the next encoder index pulse.
3	Search the positive edge of the home switch, than search the next encoder index pulse.
5	Search the negative edge of the home switch, than 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 the home position. All position values (objects 6062 <sub>h</sub> , 6063 <sub>h</sub> , and 6064 <sub>h</sub> ) are set to zero, but the motor will not move.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6098 <sub>h</sub>	Homing method	Variable	SIGNED8

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	Refer to CiA402-3	0, 3, 5, 19, 21, 33, 34, 35	0

### 4.3.3.6 Object 6099<sub>h</sub>: Homing Speeds

This object indicates the configured speeds used during homing procedure. The values are given in units, which can be selected with object 208C<sub>h</sub>, described in paragraph 0.

Using object 6099<sub>h</sub> 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<sub>h</sub>) and then stopped at the exact switch point.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6099 <sub>h</sub>	Homing speeds	ARRAY	UNSIGNED32

#### ENTRY DESCRIPTION

Sub-Index	Description	Access	PDO Mapping	Value Range	Default Value
01 <sub>h</sub>	Fast homing speed	rw	Refer to CiA402-3	UNSIGNED32	Depends on the unit setting
02 <sub>h</sub>	Slow homing speed	rw	Refer to CiA402-3	UNSIGNED32	Depends on the unit setting

### 4.3.3.7 Object 609A<sub>h</sub>: Homing Acceleration

This object indicates the configured acceleration and deceleration to be used during homing operation. The values for accelerating to the fast homing speed and for decelerating to the slow homing speed can be set here. The values are given in units, which can be selected with object 208E<sub>h</sub>, described in paragraph 0.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
609A <sub>h</sub>	Homing acceleration	Variable	UNSIGNED32

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range
00 <sub>h</sub>	rw	Refer to CiA402-3	UNSIGNED32

### 4.3.3.8 Object 2100<sub>h</sub>: Home Offset Display

The value is given in microsteps.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2100 <sub>h</sub>	Home offset display	Variable	SIGNED32

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	ro	no	Refer to home offset	0

### 4.3.4 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 start) 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<sub>h</sub>.
- Write 6 to object 6040<sub>h</sub> to switch to READY\_TO\_SWITCH\_ON state.
- Write 7 to object 6040<sub>h</sub> to switch to SWITCHED\_ON state.
- Write 15 to object 6040<sub>h</sub> to switch to OPERATION\_ENABLED state.
- Select homing method 19 by writing 19 to object 6098<sub>h</sub>.
- Set the homing speeds by writing e.g. 50000 to object 6099<sub>h</sub> sub index 1 and e.g. 10000 to object 6099<sub>h</sub> sub index 2.
- Write 31 to object 6040<sub>h</sub> to start the homing process.
- Press and release the home switch.
- When homing has finished, write 15 to object 6040<sub>h</sub> again.

## 4.4 Velocity Mode

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

### THE OPERATION OF THE REFERENCE VALUE GENERATOR AND THE INPUT PARAMETERS INCLUDE:

- Velocity
- Acceleration
- Deceleration
- Emergency stop

Please refer to object 6060<sub>h</sub> (section 4.1.6) for information about how to choose an operation mode. Object 6061<sub>h</sub> (section 4.1.7) shows the operation mode that is set.

### 4.4.1 Detailed Object Definitions

#### 4.4.1.1 Object 6040<sub>h</sub>: Controlword

This object indicates the received command controlling the power drive system finite state automation (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to *Figure 4.1* Finite state machine for detailed information.

#### STRUCTURE OF CONTROLWORD / VALUE DEFINITION

<b>15</b>	<b>9</b>	<b>8</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>0</b>
<i>see 4.2.1.1</i>	<i>Halt</i>	<i>see 4.2.1.1</i>	<i>reference ramp</i>	<i>unlock ramp</i>	<i>enable ramp</i>	<i>see 4.2.1.1</i>		
<b>MSB</b>							<b>LSB</b>	

*There are no mode specific bits supported in vl mode.*

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6040 <sub>h</sub>	Controlword	Variable	UNSIGNED16

#### ENTRY DESCRIPTION

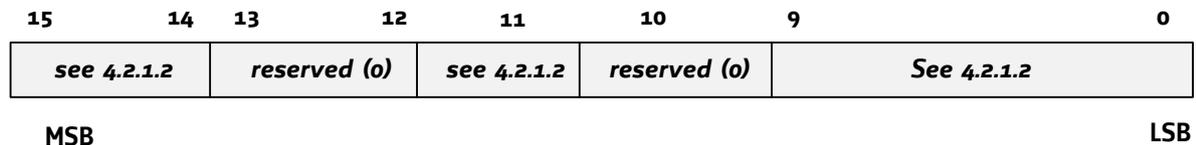
Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	Refer to CiA402-3	See command coding above	Operation mode specific

#### 4.4.1.2 Object 6041<sub>h</sub>: Statusword

This object provides the status of the PDS FSA. It reflects the status of the CiA402 state machine. Please refer to *Figure 4.1 Finite state machine* for detailed information. The object is structured as defined below.

For more information about the coding refer to the *CANopen Drives and motion control device profile, part 2* please.

##### STRUCTURE OF STATUSWORD / VALUE DEFINITION



*There are no mode specific bits supported in vl mode.*

##### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6041 <sub>h</sub>	Controlword	Variable	UNSIGNED16

##### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	r0	Refer to CiA402-3	See state coding above	No

#### 4.4.1.3 Object 6042<sub>h</sub>: vl Target Velocity

This object indicates the required velocity of the system. It is the target velocity when the velocity mode is active. Use this object to control the velocity of the motor in velocity mode. When this object is changed the motor will be accelerated or decelerated to the new velocity. The range depends on the chosen units, which can be internal or user-specific. Please refer to object 208C<sub>h</sub> (paragraph 0) for more information about that.

##### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6042 <sub>h</sub>	vl target velocity	Variable	SIGNED16

##### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	Refer to CiA402-3	SIGNED32	Depends on the units.

#### 4.4.1.4 Object 6043<sub>h</sub>: vI Velocity Demand

The object provides the instantaneous velocity generated by the ramp function. It shows the last target velocity that has been set using object 6042<sub>h</sub>. The value is given in the same unit as the target velocity, too. Please refer to object 208C<sub>h</sub> (paragraph 0) for more information about that.

##### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6043 <sub>h</sub>	vI velocity demand	Variable	SIGNED16

##### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	ro	Refer to CiA402-3	SIGNED32	0

#### 4.4.1.5 Object 6044<sub>h</sub>: vI Velocity Actual Value

This object shows the actual velocity of the motor when the motor is in velocity mode. The value is given in the same unit as vI target velocity and vI velocity demand. Please refer to object 208C<sub>h</sub> (paragraph 0) for more information about that.

##### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6044 <sub>h</sub>	vI velocity actual value	Variable	SIGNED16

##### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	ro	Refer to CiA402-3	SIGNED32	0

#### 4.4.1.6 Object 6046<sub>h</sub>: vI Velocity Min Max Amount

This object indicates the configured minimum and maximum amount of velocity in velocity mode.

##### OBJECT DESCRIPTION

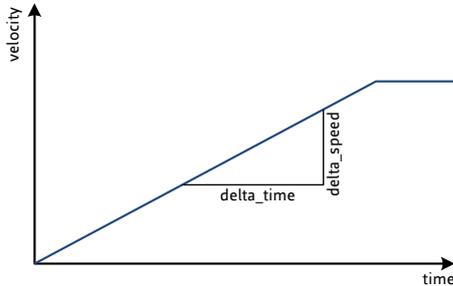
Index	Name	Object Code	Data Type
6046 <sub>h</sub>	vI velocity min max amount	ARRAY	UNSIGNED32

##### ENTRY DESCRIPTION

Sub-Index	Description	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	Highest sub-index supported	c	Refer to CiA402-3	02 <sub>h</sub>	2
01 <sub>h</sub>	Minimum velocity amount	rw/ro	Refer to CiA402-3	UNSIGNED32	0
02 <sub>h</sub>	Maximum velocity amount	rw/ro	Refer to CiA402-3	UNSIGNED32	2047

#### 4.4.1.7 Object 6048<sub>h</sub>: vI Velocity Acceleration

This object indicates the configured delta speed and delta time of the slope of the acceleration ramp as shown in the following figure. Object 6048<sub>h</sub> provides the acceleration used in velocity mode. This value is also used for the deceleration. The value of the delta speed is given in internal or in user units (rpm/s).



$$vI \text{ velocity deceleration} = \frac{\text{delta speed}}{\text{delta time}}$$

Figure 4.13 Transfer characteristic of the velocity acceleration

##### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6048 <sub>h</sub>	vI velocity acceleration	Record	vI acceleration deceleration

##### ENTRY DESCRIPTION

Sub-Index	Description	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	Highest sub-index supported	c	Refer to CiA402-3	02 <sub>h</sub>	2
01 <sub>h</sub>	Delta speed	rw	Refer to CiA402-3	UNSIGNED32	1000
02 <sub>h</sub>	Delta time	ro	Refer to CiA402-3	UNSIGNED16	1msec

#### 4.4.1.8 Object 6049<sub>h</sub>: vI Velocity Deceleration

This object contains the same value as set in object 6048<sub>h</sub>. A separate declaration value is not supported.

##### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6049 <sub>h</sub>	vI velocity deceleration	Record	vI acceleration deceleration

##### ENTRY DESCRIPTION

Sub-Index	Description	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	Highest sub-index supported	c	Refer to CiA402-3	02 <sub>h</sub>	2
01 <sub>h</sub>	Delta speed	ro	Refer to CiA402-3	UNSIGNED32	Depends on the units
02 <sub>h</sub>	Delta time	ro	Refer to CiA402-3	UNSIGNED16	10msec

### 4.4.2 How to Move a Motor in vI Mode

Here is a little example that shows how to get a motor running in vI mode. In this little example we assume that the module has been reset (and then switched to start) 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<sub>h</sub>.
- Select vI mode by writing 2 to object 6060<sub>h</sub>.
- Write 6 to object 6040<sub>h</sub> to switch to READY\_TO\_SWITCH\_ON state.
- Write 7 to object 6040<sub>h</sub> to switch to SWITCHED\_ON state.
- Write 15 to object 6040<sub>h</sub> to switch to OPERATION\_ENABLED state.
- Write the desired target speed (e.g. 500) to object 6042<sub>h</sub>. The motor now accelerates to that speed.
- Stop the motor by writing 0 to object 6042<sub>h</sub>.

## 4.5 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

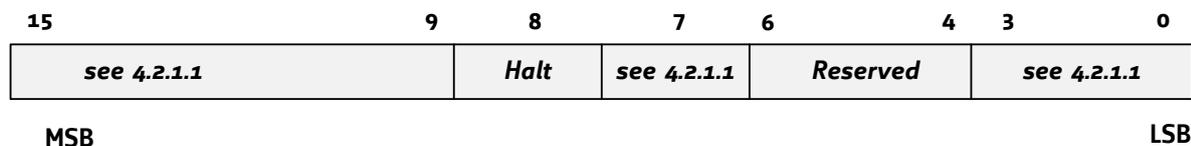
### THE OPERATION OF THE REFERENCE VALUE GENERATOR AND ITS INPUT PARAMETERS INCLUDE:

- Profile velocity
- Profile acceleration
- Profile deceleration
- Emergency stop
- Motion profile type
- Detailed object definitions

#### 4.5.1.1 Object 6040<sub>h</sub>: Controlword

This object indicates the received command controlling the power drive system finite state automation (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to Figure 4.1 for detailed information.

#### STRUCTURE OF CONTROLWORD / VALUE DEFINITION



There are no mode specific bits supported in this mode.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6040 <sub>h</sub>	Controlword	Variable	UNSIGNED16

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	Refer to CiA402-3	See command coding above	Device and operation mode specific

### 4.5.1.2 Object 6041<sub>h</sub>: Statusword

This object provides the status of the PDS FSA. It reflects the status of the CiA402 state machine. Please refer to *Figure 4.1 Finite state machine* for detailed information. The object is structured as defined below.

For more information about the coding refer to the *CANopen Drives and motion control device profile, part 2* please.

#### STRUCTURE OF STATUSWORD / VALUE DEFINITION

15	14	13	12	11	10	9	0
<b>see 4.2.1.2</b>	<b>Max slippage error</b>	<b>Speed</b>	<b>see 4.2.1.2</b>	<b>Target reached</b>	<b>see 4.2.1.2</b>		
<b>MSB</b>						<b>LSB</b>	

#### OPERATION MODE SPECIFIC BITS IN PV MODE

Bit	Name	Definition
10	Target reached	Indicates that the desired speed is reached
12	Speed	Not supported
13	Max slippage error	Not supported

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6041 <sub>h</sub>	Controlword	Variable	UNSIGNED16

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	ro	Refer to CiA402-3	See state coding above	No

### 4.5.1.3 Object 6062<sub>h</sub>: Position Demand Value

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

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6062 <sub>h</sub>	Position demand value	Variable	SIGNED32

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	ro	Refer to CiA402-3	SIGNED32	No

### 4.5.1.4 Object 6063<sub>h</sub>: Position Actual Internal Value

This object indicates the actual position of the encoder, re-scaled to the microstep resolution of the motor. If necessary, the data unit may be transformed from user-defined units to increments. The value is given in microsteps or encoder steps, depending on object 606A<sub>h</sub> (described in paragraph 4.1.8).

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6063 <sub>h</sub>	Position actual internal value	Variable	SIGNED32

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	ro	Refer to CiA402-3	SIGNED32	No

#### 4.5.1.5 Object 6064<sub>h</sub>: Position Actual Value

This object provides the actual value of the position measurement device. It always contains the same value as object 6063<sub>h</sub>. The value is given in microsteps.

##### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6064 <sub>h</sub>	Position actual value	Variable	SIGNED32

##### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	ro	Refer to CiA402-3	SIGNED32	No

#### 4.5.1.6 Object 6065<sub>h</sub>: 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 is given in microsteps. If the value of the following error window is 0 or FFFF FFFF<sub>h</sub>, the following control will be switched off.

When the difference between motor position (object 6062<sub>h</sub>) and encoder position (object 6063<sub>h</sub> or 6064<sub>h</sub>) 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. Setting this object to a too low value will lead to false alarms.

##### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6065 <sub>h</sub>	Following error window	Variable	UNSIGNED32

##### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	Refer to CiA402-3	UNSIGNED32	0... 8388607

#### 4.5.1.7 Object 606C<sub>h</sub>: Velocity Actual Value

This object gives the actual velocity value derived either from the velocity sensor or from the position sensor.

##### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
606C <sub>h</sub>	Velocity actual value	Variable	INTEGER32

##### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	ro	Refer to CiA402-3	INTEGER32	no

#### 4.5.1.8 Object 607D<sub>h</sub>: 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. The position actual value is always checked against these limits. The limit position is always relative to the machine home position. Before being compared with the target position, they have to be 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

This calculation needs only be performed when home offset or software position limit is changed.

The limit positions are given in microsteps (same as target position). Two limits for moving the motor in positioning mode can be set here. Overstepping the window will lead to an emergency message and the motor will be stopped.

##### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
607D <sub>h</sub>	Software position limit	Array	SIGNED32

##### ENTRY DESCRIPTION

Sub-Index	Description	Access	PDO Mapping	Value Range	Default Value
01 <sub>h</sub>	Min position limit	rw	Refer to CiA402-3	SIGNED32	-2147483647... +2147483647
02 <sub>h</sub>	Max position limit	rw	Refer to CiA402-3	SIGNED32	-2147483647... +2147483647

#### 4.5.1.9 Object 6083<sub>h</sub>: Profile Acceleration

This object indicates the configured acceleration. Object 6083<sub>h</sub> sets the maximum acceleration to be used in profile velocity mode.

Please choose the units for object 6083<sub>h</sub> with object 208E<sub>h</sub>, described in paragraph 0. Object 208E<sub>h</sub> reads 0 when internal units are selected or 179 when user units (PPS/s) are selected.

##### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6083 <sub>h</sub>	Profile acceleration	Variable	UNSIGNED32

##### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	Refer to CiA402-3	UNSIGNED32	Depends on the units

#### 4.5.1.10 Object 6084<sub>h</sub>: Profile Deceleration

This object indicates the configured deceleration. It sets the maximum deceleration used in profile velocity mode.

Please choose the units for object 6083<sub>h</sub> with object 208E<sub>h</sub>, described in paragraph 0. Object 208E<sub>h</sub> reads 0 when internal units are selected or 179 when user units (PPS/s) are selected.

##### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6084 <sub>h</sub>	Profile deceleration	Variable	UNSIGNED32

##### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	Refer to CiA402-3	UNSIGNED32	Depends on the units

### 4.5.1.11 Object 60FF<sub>h</sub>: Target Velocity

This object indicates the configured target velocity and is used as input for the trajectory generator. Object 60FF<sub>h</sub> 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<sub>h</sub> and 6084<sub>h</sub>. The values are given in units, which can be selected with object 208C<sub>h</sub>, described in paragraph 5.2.15.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
60FF <sub>h</sub>	Target velocity	Variable	SIGNED32

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	Refer to CiA402-3	SIGNED32	-2147483647... +2147483647 / depends on selected units

## 4.5.2 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 start) 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<sub>h</sub>.
- Select *pv* mode by writing 3 to object 6060<sub>h</sub>.
- Write 6 to object 6040<sub>h</sub> to switch to READY\_TO\_SWITCH\_ON state.
- Write 7 to object 6040<sub>h</sub> to switch to SWITCHED\_ON state.
- Write 15 to object 6040<sub>h</sub> to switch to OPERATION\_ENABLED state.
- Write the desired target speed (e.g. 100000) to object 60FF<sub>h</sub>. The motor now accelerates to that speed.
- Stop the motor by writing 0 to object 60FF<sub>h</sub>.

## 5 Manufacturer Specific Area

The manufacturer segment contains manufacturer specific objects. These objects control special features of the TRINAMIC motion control devices TCMC-1180, TCMC-1160, and TCMC-1140.

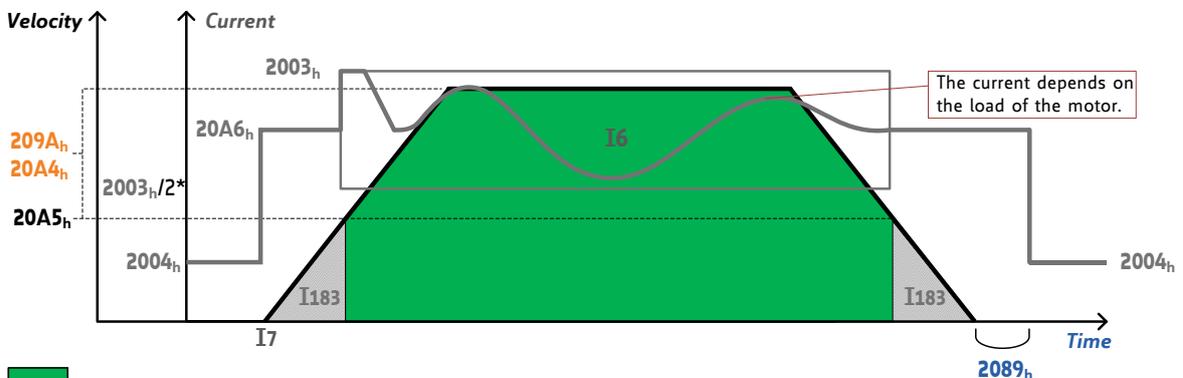
### 5.1 coolStep™ related Objects (Overview)

The figure below gives an overview of the coolStep related objects. Please have in mind that the figure shows only one example for a drive. There are parameters which concern the configuration of the current. Other parameters are for velocity regulation and for time adjustment.

#### 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



- coolStep area
- area without coolStep
- Current and objects
- Velocity and objects

Time object

stallGuard2 objects

\* The lower threshold of the coolStep current can be adjusted up to  $2003_h/4$ . Refer to parameter object 2098<sub>h</sub>.

COOLSTEP RELATED OBJECTS		
Note: smartEnergy is an earlier name for coolStep		
Object	Name	Description
2003 <sub>h</sub>	Absolute max. current (CS / Current Scale)	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; e.g. step 0 = 0... 7, step 1 = 8... 15 and so on). <i>The most important motor setting, since too high values might cause motor damage!</i>
2004 <sub>h</sub>	Standby current	The current limit two seconds after the motor has stopped.

<b>COOLSTEP RELATED OBJECTS</b>		
<i>Note: smartEnergy is an earlier name for coolStep</i>		
<b>Object</b>	<b>Name</b>	<b>Description</b>
<b>2098<sub>h</sub></b>	smartEnergy current minimum (SEIMIN)	Sets the lower motor current limit for coolStep operation by scaling the CS (current Scale, see object 2003 <sub>h</sub> ) value. Minimum motor current: 0 – 1/2 of CS 1 – 1/4 of CS
<b>2099<sub>h</sub></b>	smartEnergy current down step	Sets the number of stallGuard2 readings above the upper threshold necessary for each current decrement of the motor current. Number of stallGuard2 measurements per decrement: Scaling: 0... 3: 32, 8, 2, 1 0: slow decrement 3: fast decrement
<b>209B<sub>h</sub></b>	smartEnergy current up step	Sets the current increment step. The current becomes incremented for each measured stallGuard2 value below the lower threshold (see smartEnergy hysteresis start). current increment step size: Scaling: 0... 3: 1, 2, 4, 8 0: slow increment 3: fast increment / fast reaction to rising load
<b>20A6<sub>h</sub></b>	smartEnergy slow run current	Sets the motor current which is used below the threshold speed. Please adjust the threshold speed with axis parameter 182.
<b>209A<sub>h</sub></b>	smartEnergy hysteresis	Sets the distance between the lower and the upper threshold for stallGuard2 reading. Above the upper threshold the motor current becomes decreased.
<b>20A4<sub>h</sub></b>	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.
<b>20A5<sub>h</sub></b>	smartEnergy threshold speed	Above this speed coolStep™ becomes enabled.
<b>2089<sub>h</sub></b>	Standby delay	Standstill period before the current is changed down to standby current. The standard value is 200 (value equates 2000msec).

## 5.2 Detailed Object Specifications

### 5.2.1 Object 2000<sub>h</sub>: Microstep Resolution

This object sets the microstep resolution of the drive. A value of 6 means 64 microsteps ( $2^6$ ). It is only writeable in the SWITCHED\_ON\_DISABLED state, but always readable.

#### VALUE DESCRIPTION

Value	Description
0	1 step
1	2 microsteps
2	4 microsteps
3	8 microsteps
4	16 microsteps
5	32 microsteps
6	64 microsteps
7	128 microsteps
8	256 microsteps

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2000 <sub>h</sub>	Microstep resolution	Variable	UNSIGNED8

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	0... 8	6

### 5.2.2 Object 2001<sub>h</sub>: Fullstep Resolution

This object shows the fullstep resolution of the motor. For the PD-1180 the default value is normally 200.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2001 <sub>h</sub>	Fullstep resolution	Variable	UNSIGNED16

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	ro	no	200	<i>value is module specific</i>

### 5.2.3 Object 2002<sub>h</sub>: Brake Delay Times

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

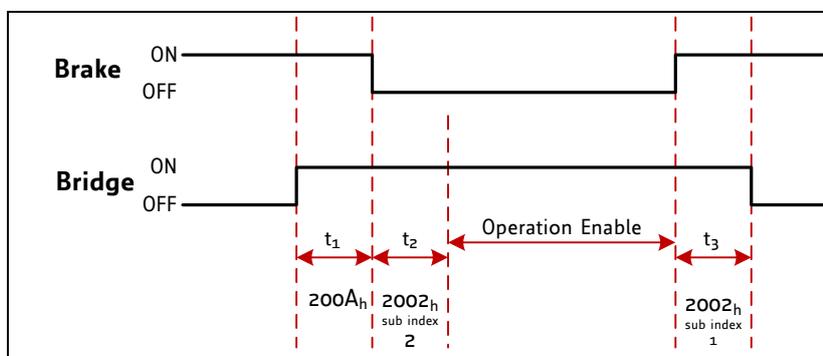


Figure 5.1 Brake output timing

**OBJECT DESCRIPTION**

Index	Name	Object Code	Data Type
2002 <sub>h</sub>	Brake delay times	ARRAY	UNSIGNED16

**ENTRY DESCRIPTION**

Sub-Index	Description	Access	PDO Mapping	Value Range	Default Value
01 <sub>h</sub>	Time between applying brake / disabling power stage	rw	no	0... 65535	0
02 <sub>h</sub>	Time between releasing brake / switching the state machine to operational	rw	no	0... 65535	0

**5.2.4 Object 2003<sub>h</sub>: 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 Code	Data Type
2003 <sub>h</sub>	Maximum current	Variable	UNSIGNED8

**ENTRY DESCRIPTION**

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	0... 255	<i>value is module specific</i>

**5.2.5 Object 2004<sub>h</sub>: 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 Code	Data Type
2004 <sub>h</sub>	Standby current	Variable	UNSIGNED8

**ENTRY DESCRIPTION**

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	0... 255	<i>value is module specific</i>

**5.2.6 Object 2005<sub>h</sub>: 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

**OBJECT DESCRIPTION**

Index	Name	Object Code	Data Type
2005 <sub>h</sub>	Limit switches	Variable	UNSIGNED32

**ENTRY DESCRIPTION**

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	0... 7	0

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.

## 5.2.7 Object 200A<sub>h</sub>: 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 2002<sub>h</sub>.

### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
200A <sub>h</sub>	Enable drive delay time	Variable	UNSIGNED16

### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	0... 65535	0

## 5.2.8 Object 200B<sub>h</sub>: Encoder Parameters

This object defines the polarity of the encoder null channel. It is only writable in SWITCHED\_ON\_DISABLED state.

### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
200B <sub>h</sub>	Encoder parameters	ARRAY	UNSIGNED8

### ENTRY DESCRIPTION

Sub-Index	Description	Access	PDO Mapping	Value Range	Default Value
01 <sub>h</sub>	Null channel polarity	rw	no	UNSIGNED8	7
02 <sub>h</sub>	Direction of rotation	rw	no	0/1	0
03 <sub>h</sub>	Initialize position <sup>*)</sup>	rw	no	0/1	1

<sup>\*)</sup> Write 0 to sub-index 03<sub>h</sub> to stop automatic transfer of actual encoder position to all related registers. Per default, the encoder position is transferred automatically.

## 5.2.9 Object 200C<sub>h</sub>: 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. The default configuration is made for this case.

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 Code	Data Type
200C <sub>h</sub>	Brake current feed	ARRAY	UNSIGNED8

### ENTRY DESCRIPTION

Sub-Index	Description	Access	PDO Mapping	Value Range	Default Value
01 <sub>h</sub>	Apply current	rw	no	UNSIGNED8	0
02 <sub>h</sub>	Release current	rw	no	UNSIGNED8	255

## 5.2.10 Object 204E<sub>h</sub>: Boost Current

This object is used to set the current used for acceleration and deceleration phases. If set to 0 the current set with object 2003<sub>h</sub> (maximum current setting) will be used. The value 1 can be chosen for a minimum boost current setting. A value of 255 means 100% of the maximum possible current of the module (this depends on the module type).

### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
204E <sub>h</sub>	Boost current	Variable	UNSIGNED8

### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	0... 255	0

## 5.2.11 Object 2085<sub>h</sub>: Ramp Divisor

Use this object to define the ramp divisor when internal units are selected for velocity and acceleration. This object can only be written in the SWITCHED\_ON\_DISABLE state (but is always readable). The exponent of the scaling factor for the ramp generator- should be de/incremented carefully (in steps of one).

### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2085 <sub>h</sub>	Ramp divisor	Variable	UNSIGNED8

### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw <i>not writable if user units are selected</i>	no	0... 13	7

## 5.2.12 Object 2086<sub>h</sub>: Pulse Divisor

Use this object to define the pulse divisor when internal units are to be used for velocity and acceleration. This object can only be written in the SWITCHED\_ON\_DISABLED state (but is always readable). The exponent of the scaling factor for the pulse (step) generator – should be de/incremented carefully (in steps of one).

### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2086 <sub>h</sub>	Pulse divisor	Variable	UNSIGNED8

### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw <i>not writable if user units are selected</i>	no	0... 13	3

### 5.2.13 Object 2087<sub>h</sub>: Maximum Velocity

The functionality of this object depends on the unit selection (refer to object 208C<sub>h</sub> / chapter 0):

- If internal units are selected this object will be set to the fixed value of 2047 and cannot be changed.
- If user units are selected the maximum velocity (RPM or PPS) that is to be used must be set here. A lower value in this object leads to a better accuracy of the unit conversion.

This value can only be changed when the drive is in the SWITCHED\_ON\_DISABLED state.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2087 <sub>h</sub>	Maximum velocity	Variable	UNSIGNED32

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	143... 293 (rpm) resp. 2047 (internal)	2047

### 5.2.14 Object 2088<sub>h</sub>: Maximum Acceleration

The functionality of this object also depends on the unit selection (refer to object 208E<sub>h</sub> / chapter 0):

- If internal units are selected this object will be set to the fixed value of 2047 and cannot be changed.
- If user units are selected the maximum acceleration (RPM/s or PPS/s) that is to be used must be set here. A lower value in this object leads to a better accuracy of the unit conversion.

This value can only be change when the drive is in the SWITCHED\_ON\_DISABLED state.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2088 <sub>h</sub>	Maximum acceleration	Variable	UNSIGNED32

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	2183... 4000 (rpm/s) resp. 2047 (internal)	2047

### 5.2.15 Object 2089<sub>h</sub>: Standby Delay

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

**UNIT:** 10msec

Index	Name	Object Code	Data Type
2089 <sub>h</sub>	Standby delay	Variable	INTEGER16

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	1... 65535	200

## 5.2.16 Object 208C<sub>h</sub>: Velocity Dimension Index

With this object different units can be chosen:

- Writing 0 selects internal units.
- Writing 164 sets RPM for velocity and RPM/s for acceleration.
- 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 Code	Data Type
208C <sub>h</sub>	Velocity dimension index	Variable	UNSIGNED8

### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	0, 164, 181	0

When using PPS units the full velocity range cannot be used in velocity mode. Use profile velocity mode instead.

## 5.2.17 Object 208E<sub>h</sub>: Acceleration Dimension Index

With this object, the unit for acceleration can be read out. The unit can be set using object 208C<sub>h</sub>.

Object 208E<sub>h</sub> reads 0 when internal units are selected. It reads 177 when RPM/s is selected, and the value for PPS/s is 179.

### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
208E <sub>h</sub>	Acceleration dimension index	Variable	UNSIGNED8

### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	ro	no	0, 177, 179	0

## 5.2.18 Object 2092<sub>h</sub>: 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 Code	Data Type
2092 <sub>h</sub>	Chopper blank time	Variable	UNSIGNED8

### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	0... 3	2

## 5.2.19 Object 2093<sub>h</sub>: Chopper Mode

Select the chopper mode with this object:

- 0 – spreadCycle™ chopper
- 1 – classic const. off time chopper

### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2093 <sub>h</sub>	Chopper mode	Variable	UNSIGNED8

### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	0/1	0

## 5.2.20 Object 2094<sub>h</sub>: 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 Code	Data Type
2094 <sub>h</sub>	Chopper hysteresis decrement	Variable	UNSIGNED8

### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	0... 3	0

## 5.2.21 Object 2095<sub>h</sub>: 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<sub>h</sub>.

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

### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2095 <sub>h</sub>	Chopper hysteresis end	Variable	SIGNED8

### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	-3... 12	2

### 5.2.22 Object 2096<sub>h</sub>: Chopper Hysteresis Start

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

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2096 <sub>h</sub>	Chopper hysteresis start	Variable	UNSIGNED8

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	0... 8	3

### 5.2.23 Object 2097<sub>h</sub>: Chopper Off Time

The off time setting controls the minimum chopper frequency. An off time within the range of 5µs to 20µs will fit.

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 the motor can free-wheel.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2097 <sub>h</sub>	Chopper off time	Variable	UNSIGNED8

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	0 / 2... 15	5

### 5.2.24 Object 2098<sub>h</sub>: Smart Energy Current Minimum

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

Minimum motor current:

0 – 1/2 of CS

1 – 1/4 of CS

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2098 <sub>h</sub>	Smart energy current minimum	Variable	UNSIGNED8

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	0/1	1

### 5.2.25 Object 2099<sub>h</sub>: 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.

Number of stallGuard2 measurements per decrement:

Scaling: 0... 3 – 32, 8, 2, 1

0 – slow decrement

3 – fast decrement

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2099 <sub>h</sub>	Smart energy current down step	Variable	UNSIGNED8

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	0... 3	0

### 5.2.26 Object 209A<sub>h</sub>: 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 stallGuard threshold:

(smartEnergy hysteresis start + smartEnergy hysteresis + 1) \* 32

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
209A <sub>h</sub>	Smart energy hysteresis	Variable	UNSIGNED8

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	0... 15	0

### 5.2.27 Object 209B<sub>h</sub>: 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<sub>h</sub>).

Current increment step size:

Scaling: 0... 3 – 1, 2, 4, 8

0 – slow increment

3 – fast increment / fast reaction to rising load

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
209B <sub>h</sub>	Smart energy current up step	Variable	UNSIGNED8

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	0... 3	0

## 5.2.28 Object 209C<sub>h</sub>: Smart Energy Hysteresis Start

This object serves to set the lower threshold for the stallGuard2 value (see *smart Energy current up step* above).

### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
209C <sub>h</sub>	Smart energy hysteresis start	Variable	UNSIGNED8

### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	0... 15	0

## 5.2.29 Object 209D<sub>h</sub>: stallGuard2 Filter Enable

This object is used to set the stallGuard2 filter for more precision of the measurement. If set, reduces the measurement frequency to one measurement per four fullsteps.

In most cases it is expedient to set the filtered mode before using coolStep. Use the standard mode for step loss detection.

0 – standard mode

1 – filtered mode

### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
209D <sub>h</sub>	stallGuard2™ filter enable	Variable	UNSIGNED8

### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	0/1	1

## 5.2.30 Object 209E<sub>h</sub>: 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.

Value	Description
0	Indifferent value
1... 63	less sensitivity
-1... -64	higher sensitivity

### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
209E <sub>h</sub>	stallGuard2™ threshold	Variable	SIGNED8

### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	-64... 63	5

### 5.2.31 Object 209F<sub>h</sub>: Slope Control High Side

This object determines the slope of the motor driver outputs.

0 – lowest slope

3 – fastest slope

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
209F <sub>h</sub>	Slope control high side	Variable	UNSIGNED8

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	0.. 3	3

### 5.2.32 Object 20A0<sub>h</sub>: Slope Control Low Side

This object determines the slope of the motor driver outputs. Set it identical to *slope control high side* object 209F<sub>h</sub>.

0 – lowest slope

3 – fastest slope

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
20A0 <sub>h</sub>	Slope control low side	Variable	UNSIGNED8

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	0.. 3	3

### 5.2.33 Object 20A1<sub>h</sub>: Short Protection Disable

This object is used to enable or to disable the short to ground protection. *Use default value!*

0 – Short to GND protection

1 – Short to GND protection disabled

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
20A1 <sub>h</sub>	Short protection disable	Variable	UNSIGNED8

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	0/1	0

### 5.2.34 Object 20A2<sub>h</sub>: Short Detection Timer

This object provides the timing of the short detection. *Use default value!*

Value	Timing
0	3.2µs
1	1.6µs
2	1.2µs
3	0.8µs

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
20A2 <sub>h</sub>	Short detection timer	Variable	UNSIGNED8

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	0.. 3	0

### 5.2.35 Object 20A3<sub>h</sub>: 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!*

0 – Full scale sense resistor voltage is 1/18 VDD

1 – Full scale sense resistor voltage is 1/36 VDD

(This refers to a current setting of 31 and DAC value 255.)

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
20A3 <sub>h</sub>	Vsense	Variable	UNSIGNED8

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	0/1	1

### 5.2.36 Object 20A4<sub>h</sub>: Stop on Stall

Below this speed the motor will not be stopped. Above this speed the motor will stop in case the stallGuard2 load value reaches zero.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
20A4 <sub>h</sub>	Stop on stall	Variable	UNSIGNED8

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	0.. 2047	0

### 5.2.37 Object 20A5<sub>h</sub>: Smart Energy Threshold Speed

Above this speed coolStep becomes enabled.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
20A5 <sub>h</sub>	Smart energy threshold speed	Variable	UNSIGNED16

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	0.. 2147483647 [pps]	0

## 5.2.38 Object 20A6<sub>h</sub>: Smart Energy Slow Run Current

This object can be used for setting the motor current below the threshold speed.

### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
20A6 <sub>h</sub>	Smart energy slow run current	Variable	UNSIGNED8

### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	0... 255 $\left[ \frac{\text{max.module current}}{255} \right]$	0

## 5.2.39 Object 2100<sub>h</sub>: Home Offset Display

The value is given in microsteps.

### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2100 <sub>h</sub>	Home offset display	Variable	SIGNED32

### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	ro	no	Refer to home offset (object 607C <sub>h</sub> / 4.3.3.4)	0

## 5.2.40 Object 2101<sub>h</sub>: Actual stallGuard2 Load Value

This object shows the actual stallGuard2 load value. It is needed for configuring stallGuard2 and for finding the fitting velocity.

### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2101 <sub>h</sub>	Actual stallGuard2™ load value	Variable	UNSIGNED8

### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	ro	no	0... 7	0

## 5.2.41 Object 2102<sub>h</sub>: Driver Error Flags

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

### THERE ARE THE FOLLOWING ERROR BITS:

Bit	Name	Function	Remark
7	OT	Overtemperature	1 = chip of due to overtemperature
6	OTPW	Temperature prewarning	1= prewarning temperature exceeded
5	UV	Driver undervoltage	1 = undervoltage on VS
4	OCHS	Overcurrent high side	3 PWM cycles with overcurrent within 63 PWM cycles
3	OLB	Open load bridge B	No PWM switch off for 14 oscillator cycles
2	OLA	Open load bridge A	No PWM switch off for 14 oscillator cycles
1	OCB	Overcurrent bridge B low side	3 PWM cycles with overcurrent within 63 PWM cycles
0	OCA	Overcurrent bridge A low side	3 PWM cycles with overcurrent within 63 PWM cycles

### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2102 <sub>h</sub>	Driver error flags	Variable	UNSIGNED8

### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	ro	no	0... 255	0

### 5.2.42 Object 2103<sub>h</sub>: Pulse Divisor Display

The pulse divisor display shows the pulse divisor that has been calculated by the drive when user units are selected or that can be set using object 2086<sub>h</sub> when internal units are selected.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2103 <sub>h</sub>	Pulse divisor display	Variable	UNSIGNED8

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	ro	no	0... 13	3

### 5.2.43 Object 2104<sub>h</sub>: Maximum Velocity Display

This object shows the value of object 2087<sub>h</sub>.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2104 <sub>h</sub>	Maximum velocity display	Variable	UNSIGNED32

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	ro	no	0... 2047	2047

### 5.2.44 Object 2105<sub>h</sub>: Ramp Divisor Display

The ramp divisor display shows the ramp divisor that has been calculated by the drive when user units are selected or that can be set using object 2085<sub>h</sub> when internal units are selected.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2105 <sub>h</sub>	Ramp divisor display	Variable	UNSIGNED8

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	ro	no	0... 13	7

### 5.2.45 Object 2106<sub>h</sub>: Maximum Acceleration Display

This object shows the value of object 2088<sub>h</sub>.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2106 <sub>h</sub>	Maximum acceleration display	Variable	UNSIGNED32

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	ro	no	0... 2047	2047

### 5.2.46 Object 2107<sub>h</sub>: Microstep Resolution Display

This object shows the microstep resolution (set by object 2000<sub>h</sub> when internal units are selected or calculated when user units are selected).

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2107 <sub>h</sub>	Microstep resolution display	Variable	UNSIGNED8

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	ro	no	0... 8	6

### 5.2.47 Object 2700<sub>h</sub>: TMCL Direct Communication

After writing the *make signature* 656b616d<sub>h</sub> to this object the drive switches to TMCL mode. The drive can then only be controlled via TMCL commands written to the *OS command* object 1023/01<sub>h</sub>

The drive has to be reset (using the *DS-301 reset application command*) to return to normal CANopen functionality.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2700 <sub>h</sub>	Microstep resolution display	Variable	UNSIGNED32

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	ro	no	Make signature 656b616d <sub>h</sub>	0

### 5.2.48 Object 2701<sub>h</sub>: Manufacturer Specific Mode

Writing the *make signature* to this object turns on the manufacturer specific mode. The manufacturer specific mode can be turned off again by writing the *kill signature* to this object.

<u>read:</u>	0 = manufacturer specific mode is inactive 1 = manufacturer specific mode is active
<u>write:</u>	656b616d <sub>h</sub> = make signature 6c6c696b <sub>h</sub> = kill signature

The manufacturer specific mode has the following features:

- PDOs do not need to be disabled and re-enabled when the PDO mapping is to be changed
- The RTR bit in the COB-ID of PDO definitions is ignored.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2701 <sub>h</sub>	Manufacturer specific mode	Variable	UNSIGNED32

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	see textbox	0

### 5.2.49 Object 2702<sub>h</sub>: Digital Inputs

Bits 16... 23 of this object reflect the states of the digital inputs of the module.

Bit	Input
16	IN_0
17	IN_1
18	IN_2
19	IN_3
etc.	

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2702 <sub>h</sub>	Digital inputs	Variable	UNSIGNED32

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	ro	no	-	0

### 5.2.50 Object 2703<sub>h</sub>: Digital Outputs

With this object the digital outputs (general purpose outputs) can be set. Bits 16... 23 of sub index 1 switch outputs OUT\_0 and OUT\_1 of the module. Bits 16... 23 of sub index 2 determine which outputs can be switched. The number of available digital outputs depends on the module type.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2703 <sub>h</sub>	Digital outputs	Array	

#### ENTRY DESCRIPTION

Sub-Index	Description	Access	PDO Mapping	Value Range	Default Value
01 <sub>h</sub>	Physical outputs	rw	no	UNSIGNED32	0
02 <sub>h</sub>	Output mask	rw	no	UNSIGNED32	0

### 5.2.51 Object 2704<sub>h</sub>: CAN Bit Rate

With this object it is possible to change the CAN bit rate. To do this, first write the new value to this object. Then, store the new setting by writing the save signature to object 2706<sub>h</sub>. After that, reset the module. The new setting becomes active now.

#### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2704 <sub>h</sub>	CAN bit rate load	Variable	UNSIGNED16

#### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	20, 50, 125, 250, 500, 1000	250

#### HOW TO CHANGE THE CAN BIT RATE:

- Write new bit rate in object 2704<sub>h</sub>.
- Write safe signature 65766173<sub>h</sub> in sub-index 01<sub>h</sub> of object 2706<sub>h</sub>.
- Reset module.

## 5.2.52 Object 2705<sub>h</sub>: Node ID

On modules that do not have address switches the node ID can be selected using this object. On modules with address switches the node ID is normally selected using the address switches. Only when the address switches are set to an invalid value (0 or >127) this object overrides the address switch setting.

To change the node ID, first write the new node ID to this object. Then, store the new setting by writing the save signature to object 2706<sub>h</sub>. After that, reset the module. The new setting becomes active now.

### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2705 <sub>h</sub>	Node ID load	Variable	UNSIGNED8

### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	rw	no	1... 127	1

### HOW TO CHANGE THE NODE ID:

- Write new node ID in object 2705<sub>h</sub>.
- Write safe signature 65766173<sub>h</sub> in sub-index 01<sub>h</sub> of object 2706<sub>h</sub>.
- Reset module.

## 5.2.53 Object 2706<sub>h</sub>: Store

Writing the save signature to this object permanently saves changes made on objects 2707<sub>h</sub> and 2708<sub>h</sub>.

Save signature: 65766173<sub>h</sub>

### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2706 <sub>h</sub>	Store	Variable	UNSIGNED32

### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	ro	no	Highest sub-index supported	1
01 <sub>h</sub>	rw	no	Save signature	0

## 5.2.54 Object 2707<sub>h</sub>: CAN Bit Rate Load

This object shows the selected CAN bit rate.

### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2707 <sub>h</sub>	CAN bit rate	Variable	UNSIGNED16

### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	ro	No	20, 50, 125, 250, 500, 800, 1000	1000

## 5.2.55 Object 2708<sub>h</sub>: Node ID Load

This object shows the selected node ID.

### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2708 <sub>h</sub>	Node ID	Variable	UNSIGNED8

### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	ro	no	1... 127	Depends on ID switches

## 5.2.56 Object 270E<sub>h</sub>: Device Analogue Inputs

This object defines the analog inputs that can be used.

Please note, that the readout value of analog inputs that are not supported by a specific module is always 0.

### OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
270E <sub>h</sub>	Device analogue inputs	ARRAY	UNSIGNED32

### ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	Highest sub-index supported	ro	-	8
01 <sub>h</sub>	Analogue input 1	ro	TCM-1140: 0... 4095 TCM-1160: 0... 4095 TCM-1180: 0... 1023	no
02 <sub>h</sub>	Analogue input 2	ro	TCM-1140: 0... 4095 TCM-1160: 0... 4095 TCM-1180: 0... 1023	no
03 <sub>h</sub>	Analogue input 3	ro	TCM-1140: 0... 4095 TCM-1160: 0... 4095 TCM-1180: 0... 1023	no
04 <sub>h</sub>	Analogue input 4	ro	TCM-1140: 0... 4095 TCM-1160: 0... 4095 TCM-1180: 0... 1023	no
05 <sub>h</sub>	Analogue input 5	ro	TCM-1140: 0... 4095 TCM-1160: 0... 4095 TCM-1180: 0... 1023	no
06 <sub>h</sub>	Analogue input 6	ro	TCM-1140: 0... 4095 TCM-1160: 0... 4095 TCM-1180: 0... 1023	no
07 <sub>h</sub>	Analogue input 7	ro	TCM-1140: 0... 4095 TCM-1160: 0... 4095 TCM-1180: 0... 1023	no
08 <sub>h</sub>	Analogue input 8	ro	TCM-1140: 0... 4095 TCM-1160: 0... 4095 TCM-1180: 0... 1023	no

## 6 Emergency Messages

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<sub>h</sub> (error register) is part of every emergency object.

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

### ERROR CODES FOR TCMC-1140 / TCMC-1160 / TCMC-1180

Error code	Add. byte					Description
	1	2	3	4	5	
0x0000	0	0/1/2	0	0	0	<b>Fault reset</b> The fault reset command has been executed.
0x1000	1	0/1/2	0	0	0	<b>Generic error: open load bridge A</b> 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.
0x1000	2	0/1/2	0	0	0	<b>Generic error: open load bridge B</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.
0x2310	0	0/1/2	0	0	0	<b>Overcurrent high side</b> The motor driver indicates an overcurrent on the high side. This can be caused by a short circuit in the driver stage.
0x2311	0	0/1/2	0	0	0	<b>Overcurrent bridge B</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.
0x2312	0	0/1/2	0	0	0	<b>Overcurrent bridge A</b> 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.
0x3230	0	0/1/2	0	0	0	<b>stallGuard2™ error</b> The actual load value exceeds the stallGuard2 limit.
0x4310	1	0/1/2	0	0	0	<b>Overtemperature pre-warning</b> The temperature in the motor driver exceeds the pre-warning limit.
0x4310	2	0/1/2	0	0	0	<b>Overtemperature error</b> The motor driver has been switched off because the temperature limit has been exceeded.
0x5441	0	255	0	0	0	<b>Shutdown switch active</b> The enable signal is missing (due to the shutdown switch) and the motor driver has been switched off.
0x6320	0	255	0	0	0	<b>Parameter error</b> The data in the received PDO is either wrong or cannot be accepted due to the internal state of the drive.
0x8110	1	255	0	0	0	<b>CAN controller overflow</b> The receive message buffer of the CAN controller hardware is full and some CAN messages are lost.
0x81110	2	255	0	0	0	<b>CAN Tx buffer overflow</b> The software CAN transmit buffer is full and thus some CAN messages are lost.
0x81110	3	255	0	0	0	<b>CAN Rx buffer overflow</b> The software CAN receive buffer is full and so some CAN messages are lost.
0x8120	0	255	0	0	0	<b>CAN error passive</b> The CAN controller has detected communication errors and has entered the CAN Error passive state.
0x8140	0	255	0	0	0	<b>CAN controller recovered from bus-off state</b> The CAN controller had detected too many errors and had changed into the bus-off state. The drive has been stopped and disabled. This message is sent after the CAN controller has recovered from bus-off state and is bus-on again.
0x8611	0	0/1/2	0	0	0	<b>Following error</b> The deviation between motor position counter and encoder position counter has exceeded the following error window.
0xff00	0	0/1/2	0	0	0	<b>Undervoltage</b> The supply voltage is too low to drive a motor.
0xff01	1	0/1/2	0	0	0	<b>Positive software limit</b> The actual position is outside the range defined by object 0x607d.

Error code	Add. byte					Description
	1	2	3	4	5	
0xff01	2	0/1/2	0	0	0	<b>Negative software limit</b> The actual position is outside the range defined by object 0x607d.
0xff01	3	0/1/2	0	0	0	<b>Positive limit switch</b> The positive limit switch has been touched outside of the homing function.
0xff01	4	0/1/2	0	0	0	<b>Negative limit switch</b> The negative limit switch has been touched outside of the homing function.

## 7 Life Support Policy

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.

© TRINAMIC Motion Control GmbH & Co. KG 2013

Information given in this data sheet is believed to be accurate and reliable. However neither 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.

All trademarks used are property of their respective owners.



## 8 Revision History

### 8.1 Firmware Revision

Version	Date	Author	Description
3.05	2010-NOV-13	OK	First version supporting all features of the PD-1180
3.16	2013-APR-18	OK	Several new or modified objects: <ul style="list-style-type: none"> <li>- save parameters</li> <li>- restore parameters</li> <li>- sensor selection code</li> <li>- profile jerk</li> <li>- target velocity</li> <li>- Boost current</li> <li>- standby delay</li> <li>- velocity dimension index</li> <li>- acceleration dimension index</li> <li>- all coolStep related parameters</li> </ul> <i>Attention!</i> Per default, the encoder is not active. Set the sensor selection code (object 606A <sub>h</sub> ) to meet your needs.

Table 8.1 Firmware revision

### 8.2 Document Revision

Version	Date	Author	Description
1.00	2010-DEC-14	SD	Initial version
1.01	2011-AUG-01	SD	Objects 1010 <sub>h</sub> and 1011 <sub>h</sub> added; object 1016 <sub>h</sub> completed.
2.00	2013-JUN-06	SD	<ul style="list-style-type: none"> <li>- Changes related to design.</li> <li>- Manual enlarged: new modules TCM-1140 and TCM-1160.</li> <li>- Profile velocity mode new.</li> <li>- Several new or modified objects related to the other modes of operation:               <ul style="list-style-type: none"> <li>- save parameters</li> <li>- restore parameters</li> <li>- sensor selection code</li> <li>- profile jerk</li> <li>- target velocity</li> <li>- boost current</li> <li>- standby delay</li> <li>- velocity dimension index</li> <li>- acceleration dimension index</li> <li>- all coolStep related parameters</li> </ul> </li> </ul> <i>Attention!</i> Per default, the encoder is not active. Set the sensor selection code (object 606A <sub>h</sub> ) to meet your needs.

Table 8.2 Document revision

## 9 References

[TCM-1180 / PD-1180]

TCM-1180 and PD-1180 hardware manuals

[TCM-1160 / PD-1160]

TCM-1160 and PD-1160 hardware manuals

[TCM-1140 / PD-1140]

TCM-1140 and PD-1140 hardware manuals