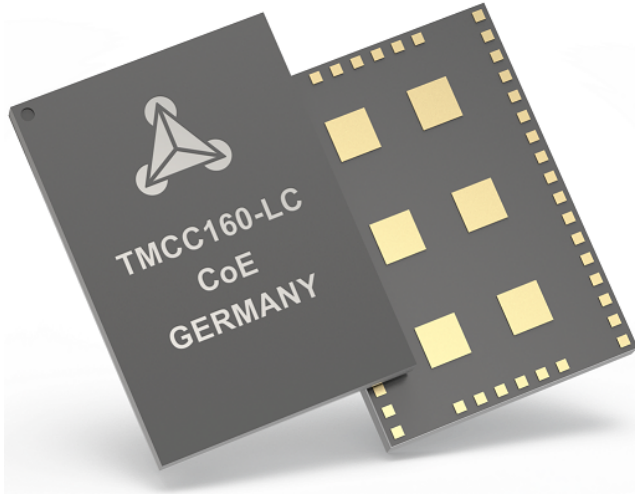


# TMCC160 CoE Firmware Manual

Firmware Version V4.00 | Document Revision V1.00 • 2017-Oct-25

The CoE Firmware is used in combination with the integrated TMCC160 motionCookie™ microsystem with 3-Phase BLDC/PMSM gate driver for up to 24V and 1A gate current.



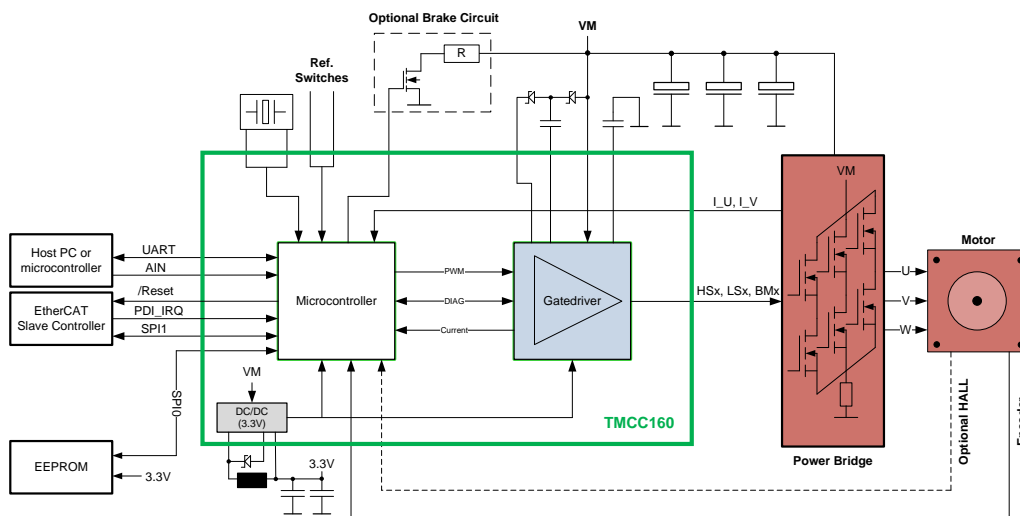
## Features

- Single axis field oriented control for BLDC/PMSM motor
- Digital Hall and ABN encoder support
- 28V DC supply voltage
- RS232/RS485 interface
- CANopen CiA 402 drive profile for Torque, Velocity, and Position control

## Applications

- Life Sciences
- Test & Measurement
- Robotics / Automation

## Simplified Block Diagram



# Contents

<b>1</b>	<b>Preface</b>	<b>5</b>
1.1	General Features of this CoE Implementation	5
1.2	Abbreviations used in this Manual	6
1.3	Firmware Update	6
<b>2</b>	<b>Communication</b>	<b>7</b>
2.1	Reference Model	7
2.2	NMT State Machine	9
2.3	Device Model	10
2.4	Object Dictionary	11
<b>3</b>	<b>Communication Area</b>	<b>12</b>
3.1	Detailed Object Specifications	12
3.1.1	Object 1000 <sub>h</sub> : Device Type	12
3.1.2	Object 1001 <sub>h</sub> : Error Register	12
3.1.3	Object 1008 <sub>h</sub> : Manufacturer Device Name	13
3.1.4	Object 1009 <sub>h</sub> : Manufacturer Hardware Version	13
3.1.5	Object 100A <sub>h</sub> : Manufacturer Software Version	14
3.1.6	Object 1018 <sub>h</sub> : Identity Object	14
3.1.7	Object 1600 <sub>h</sub> : Receive PDO Mapping Parameter	15
3.1.8	Objects 1A00 <sub>h</sub> : Transmit PDO Mapping Parameter	16
3.1.9	Objects 1C00 <sub>h</sub> : Sync Manager Communication Type	16
3.1.10	Objects 1C12 <sub>h</sub> : Sync Manager 2 PDO Assignment	17
3.1.11	Objects 1C13 <sub>h</sub> : Sync Manager 3 PDO Assignment	18
<b>4</b>	<b>Manufacturer specific area</b>	<b>19</b>
4.1	Detailed object specifications	19
4.1.1	Object 2005 <sub>h</sub> : Limit Switches	19
4.1.2	Object 200D <sub>h</sub> : Status Flags	20
4.1.3	Object 200E <sub>h</sub> : Supply Voltage	20
4.1.4	Object 200F <sub>h</sub> : Driver Temperatur	21
4.1.5	Object 2010 <sub>h</sub> : Motor Settings	21
4.1.6	Object 2020 <sub>h</sub> : Limits	22
4.1.7	Object 2030 <sub>h</sub> : Torque Mode Settings	22
4.1.8	Object 2040 <sub>h</sub> : Velocity Mode Settings	23
4.1.9	Object 2050 <sub>h</sub> : Position Mode Settings	23
4.1.10	Object 2055 <sub>h</sub> : Commutation Mode	24
4.1.11	Object 2056 <sub>h</sub> : Velocity Ramp Mode	24
4.1.12	Object 2060 <sub>h</sub> : Open Loop Settings	25
4.1.13	Object 2070 <sub>h</sub> : Hall Sensor Settings	25
4.1.14	Object 2080 <sub>h</sub> : ABN Encoder Settings	26
4.1.15	Object 2090 <sub>h</sub> : Brake Chopper Settings	26
4.1.16	Object 2095 <sub>h</sub> : IIT Settings	27
4.1.17	Object 2100 <sub>h</sub> : Home Offset Display	27
4.1.18	Object 2702 <sub>h</sub> : Digital Inputs	27
4.1.19	Object 2706 <sub>h</sub> : Store Axis Parameter	28
4.1.20	Object 270E <sub>h</sub> : Analog Inputs	28



<b>5</b>	<b>Profile specific area</b>	<b>30</b>
5.1	Detailed object specifications	31
5.1.1	Object 6060 <sub>h</sub> : Modes of operation	31
5.1.2	Object 6061 <sub>h</sub> : Modes of Operation Display	31
5.1.3	Object 6502 <sub>h</sub> : Supported Drive Modes	32
<b>6</b>	<b>Profile Position Mode</b>	<b>33</b>
6.1	Detailed Object Specifications	33
6.2	Detailed Object Specifications	34
6.2.1	Object 6040 <sub>h</sub> : Control Word	34
6.2.2	Object 6041 <sub>h</sub> : Status Word	36
6.2.3	Object 6062 <sub>h</sub> : Position Demand Value	37
6.2.4	Object 6063 <sub>h</sub> : Position Actual Internal Value	38
6.2.5	Object 6064 <sub>h</sub> : Position Actual Value	38
6.2.6	Object 6067 <sub>h</sub> : Position Window	39
6.2.7	Object 606C <sub>h</sub> : Velocity Actual Value	39
6.2.8	Object 607A <sub>h</sub> : Target Position	39
6.2.9	Object 607D <sub>h</sub> : Software Position Limit	40
6.2.10	Object 607F <sub>h</sub> : Max Profile Velocity	40
6.2.11	Object 6081 <sub>h</sub> : Profile Velocity (pp)	41
6.2.12	Object 6082 <sub>h</sub> : End Velocity	41
6.2.13	Object 6083 <sub>h</sub> : Profile Acceleration	42
6.2.14	Object 6084 <sub>h</sub> : Profile Deceleration	42
6.2.15	Object 6085 <sub>h</sub> : Quick Stop Deceleration	43
6.3	How to move a Motor in pp Mode	44
<b>7</b>	<b>Profile Velocity Mode</b>	<b>45</b>
7.1	Detailed Object Specifications	45
7.1.1	Object 6040 <sub>h</sub> : Control Word	45
7.1.2	Object 6041 <sub>h</sub> : Status Word	46
7.1.3	Object 606B <sub>h</sub> : Velocity Demand Value	48
7.1.4	Object 606C <sub>h</sub> : Velocity Actual Value	48
7.1.5	Object 6083 <sub>h</sub> : Profile Acceleration	49
7.1.6	Object 60FF <sub>h</sub> : Target Velocity	49
7.2	How to move a Motor in pv Mode	50
<b>8</b>	<b>Homing mode</b>	<b>51</b>
8.1	Homing Methods	52
8.1.1	Homing Method 17 and 18: Homing without Index Pulse	52
8.1.2	Homing Method 35: Current Position as Home Position	52
8.2	Detailed Object Specifications	53
8.2.1	Object 6040 <sub>h</sub> : Control Word	53
8.2.2	Object 6041 <sub>h</sub> : Status Word	54
8.2.3	Object 606C <sub>h</sub> : Velocity Actual Value	55
8.2.4	Object 607C <sub>h</sub> : Home Offset	56
8.2.5	Object 6098 <sub>h</sub> : Homing Method	56
8.2.6	Object 6099 <sub>h</sub> : Homing Speeds	57
8.2.7	Object 609A <sub>h</sub> : Homing Acceleration	57
8.2.8	Object 2100 <sub>h</sub> : Home Offset Display	58
8.3	How to start a Homing in hm Mode	59



<b>9</b>	<b>Cyclic synchronous Torque Mode</b>	<b>60</b>
9.1	Detailed Object Specifications . . . . .	60
9.1.1	Object 6040 <sub>h</sub> : Control Word . . . . .	60
9.1.2	Object 6041 <sub>h</sub> : Status Word . . . . .	61
9.1.3	Object 6071 <sub>h</sub> : Target Torque . . . . .	62
9.1.4	Object 6072 <sub>h</sub> : Max Torque . . . . .	63
9.1.5	Object 6074 <sub>h</sub> : Torque Demand Value . . . . .	63
9.1.6	Object 6077 <sub>h</sub> : Torque Actual Value . . . . .	64
9.1.7	Object 60B2 <sub>h</sub> : Torque offset . . . . .	64
9.2	How to move a Motor in cst Mode . . . . .	65
<b>10</b>	<b>Emergency Messages (EMCY)</b>	<b>66</b>
<b>11</b>	<b>Figures Index</b>	<b>68</b>
<b>12</b>	<b>Tables Index</b>	<b>69</b>
<b>13</b>	<b>Supplemental Directives</b>	<b>71</b>
13.1	Producer Information . . . . .	71
13.2	Copyright . . . . .	71
13.3	Trademark Designations and Symbols . . . . .	71
13.4	Target User . . . . .	71
13.5	Disclaimer: Life Support Systems . . . . .	71
13.6	Disclaimer: Intended Use . . . . .	71
13.7	Collateral Documents & Tools . . . . .	72
<b>14</b>	<b>Revision History</b>	<b>73</b>
14.1	Firmware Revision . . . . .	73
14.2	Document Revision . . . . .	73



# 1 Preface

This document specifies objects and modes of operation of the Trinamic TMCC160 BLDC/PMSM motor control module with CANopen-over-EtherCAT (CoE) firmware. The CoE firmware is designed to fulfill the EtherCAT version of the CANopen DS402 standards. The EtherCAT conformance has also been tested. This manual assumes that the reader is already familiar with the basics of EtherCAT and the CoE protocol (especially DS402).

## 1.1 General Features of this CoE Implementation

### Main Characteristics

- Communication according to EtherCAT standards
- Protocols: CoE

### SDO Communication

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

### PDO Communication

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

### Sync managers

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

### Further Characteristics

- Emergency: producer



## 1.2 Abbreviations used in this Manual

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

*Table 1: Abbreviations used in this Manual*

## 1.3 Firmware Update

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



## 2 Communication

### 2.1 Reference Model

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

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

The application and the application layer interact with service primitives.

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

*Table 2: Service Primitives*

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



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

*Table 3: Service Types*





## 2.2 NMT State Machine

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

Starting and resetting the device is controlled via the state machine. The NMT state machine consists of the states shown in figure 1.

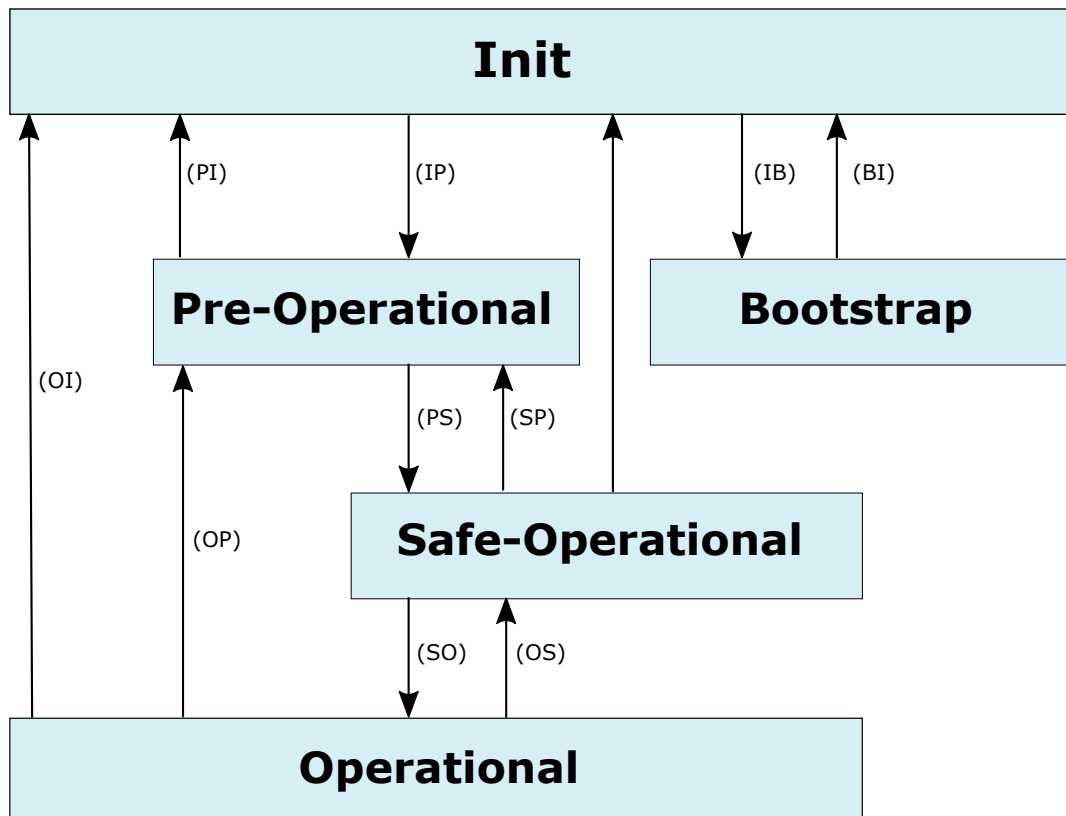


Figure 1: NMT State Machine

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

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

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

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

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



Safe-Operational state.

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

### 2.3 Device Model

A CoE device mainly consists of the following parts:

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

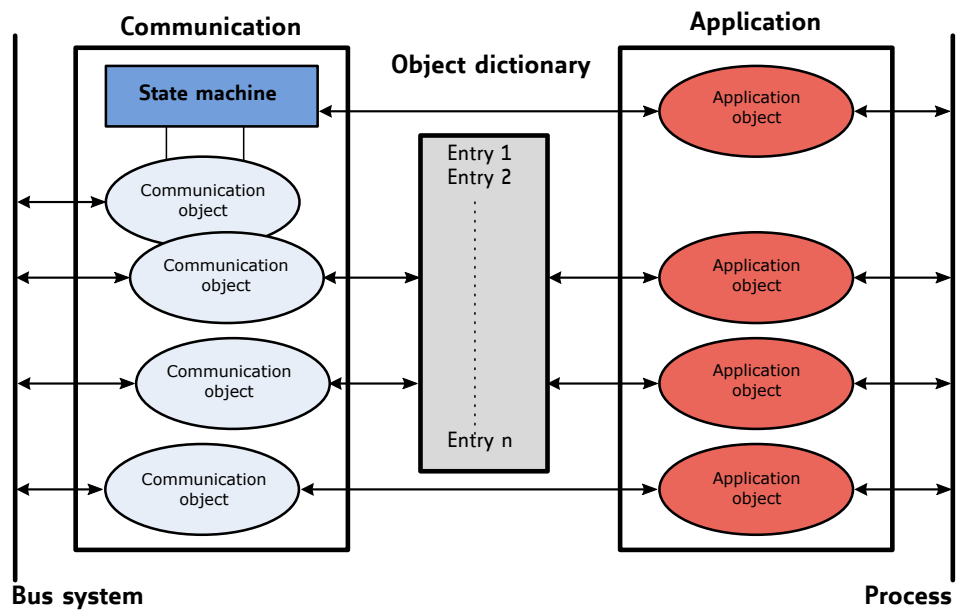


Figure 2: Device Model



## 2.4 Object Dictionary

The most important part of a device profile is the object dictionary description. The object dictionary is essentially a grouping of objects accessible via the network in an ordered pre-defined fashion. Each object within the dictionary is addressed using a 16-bit index. The overall layout of the standard object dictionary is shown in table 4:

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

*Table 4: Object Dictionary*

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.

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 TMCC160 motion control device.

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. They describe the device parameters and the device functionality of the device profile.



## 3 Communication Area

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

### 3.1 Detailed Object Specifications

#### 3.1.1 Object 1000<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 provides additional information about optional functionality of the device.

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

Table 5: Object Description (1000<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	no	UNSIGNED32	FF7A0192 <sub>h</sub>

Table 6: Entry Description (1000<sub>h</sub>)

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

This object contains error information. The CANopen device maps internal errors into object 1001<sub>h</sub>. It is part of an emergency object.

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

Table 7: Object Description (1001<sub>h</sub>)

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

Table 8: Entry Description (1001<sub>h</sub>)



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

Table 9: Error Register Bits

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

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

Object Description			
Index	Name	Object Type	Data Type
1008 <sub>h</sub>	Manufacturer Device Name	Variable	Visible String

Table 10: Object Description (1008<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	no	—	TMCC160

Table 11: Entry Description (1008<sub>h</sub>)

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

This object contains the hardware version description.

Object Description			
Index	Name	Object Type	Data Type
1009 <sub>h</sub>	Manufacturer Hardware Version	Variable	Visible String

Table 12: Object Description (1009<sub>h</sub>)



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

Table 13: Entry Description (1009<sub>h</sub>)

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

This object contains the software version description.

Object Description			
Index	Name	Object Type	Data Type
100A <sub>h</sub>	Manufacturer Software Version	Variable	Visible String

Table 14: Object Description (100A<sub>h</sub>)

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

Table 15: Entry Description (100A<sub>h</sub>)

### 3.1.6 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 286<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 Type	Data Type
1018 <sub>h</sub>	Identity object	Record	Identity

Table 16: Object Description (1018<sub>h</sub>)



Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	Number of entries	ro	no	0...3	3
01 <sub>h</sub>	Vendor ID	ro	no	UNSIGNED32	0286 <sub>h</sub>
02 <sub>h</sub>	Product code	ro	no	UNSIGNED32	160
03 <sub>h</sub>	Revision number	ro	no	UNSIGNED32	e.g. 20003 <sub>h</sub> for version 2.3

Table 17: Entry Description (1018<sub>h</sub>)

### 3.1.7 Object 1600<sub>h</sub>: Receive PDO Mapping Parameter

This object contains the mapping parameters for the RPDO the device is able to receive. The sub-index 00<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 received with the corresponding RPDO. The sub-indices from 01<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.

Object Description			
Index	Name	Object Type	Data Type
1600 <sub>h</sub>	Receive PDO mapping parameter	RECORD	PDO Mapping

Table 18: Object Description (1600<sub>h</sub>)

Entry Description				
Sub-index	Description	Access	Value Range	Default Value
00 <sub>h</sub>	Number of mapped application objects in PDO	rw	0...9	Index 1600 <sub>h</sub> : 4
01 <sub>h</sub>	Mapping entry 1	rw	UNSIGNED32	60400010 <sub>h</sub>
02 <sub>h</sub>	Mapping entry 3	rw	UNSIGNED32	607A0020 <sub>h</sub>
03 <sub>h</sub>	Mapping entry 4	rw	UNSIGNED32	60710010 <sub>h</sub>
04 <sub>h</sub>	Mapping entry 5	rw	UNSIGNED32	60FF0020 <sub>h</sub>
05 <sub>h</sub>	Mapping entry 2	rw	UNSIGNED32	0 <sub>h</sub>
06 <sub>h</sub>	Mapping entry 6	rw	UNSIGNED32	0 <sub>h</sub>
07 <sub>h</sub>	Mapping entry 7	rw	UNSIGNED32	0 <sub>h</sub>
08 <sub>h</sub>	Mapping entry 8	rw	UNSIGNED32	0 <sub>h</sub>
09 <sub>h</sub>	Mapping entry 9	rw	UNSIGNED32	0 <sub>h</sub>

Table 19: Entry Description (1600<sub>h</sub>)

### 3.1.8 Objects 1A00<sub>h</sub>: Transmit PDO Mapping Parameter

This object contains the mapping parameters for the TPDO the device is able to transmit. The sub-index 00<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 with the corresponding TPDO. The sub-indices from 01<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.

Object Description			
Index	Name	Object Type	Data Type
1A00 <sub>h</sub>	Transmit PDO mapping parameter	RECORD	PDO Mapping

Table 20: Object Description (1A00<sub>h</sub>)

Entry Description				
Sub-index	Description	Access	Value Range	Default Value
00 <sub>h</sub>	Number of mapped application objects in PDO	rw	0...9	6
01 <sub>h</sub>	Mapping entry 1	rw	UNSIGNED32	60410010 <sub>h</sub>
02 <sub>h</sub>	Mapping entry 2	rw	UNSIGNED32	60610008 <sub>h</sub>
03 <sub>h</sub>	Mapping entry 3	rw	UNSIGNED32	60640020 <sub>h</sub>
04 <sub>h</sub>	Mapping entry 4	rw	UNSIGNED32	60770010 <sub>h</sub>
05 <sub>h</sub>	Mapping entry 5	rw	UNSIGNED32	606C0020 <sub>h</sub>
06 <sub>h</sub>	Mapping entry 6	rw	UNSIGNED32	60FD0020 <sub>h</sub>
07 <sub>h</sub>	Mapping entry 7	rw	UNSIGNED32	0 <sub>h</sub>
08 <sub>h</sub>	Mapping entry 8	rw	UNSIGNED32	0 <sub>h</sub>
09 <sub>h</sub>	Mapping entry 9	rw	UNSIGNED32	0 <sub>h</sub>

Table 21: Entry Description (1A00<sub>h</sub>)

### 3.1.9 Objects 1C00<sub>h</sub>: Sync Manager Communication Type

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





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

Table 22: Sync Manager Communication Types

Object Description			
Index	Name	Object Type	Data Type
1C00 <sub>h</sub>	Sync manager communication type	RECORD	UNSIGNED8

Table 23: Object Description (1C00<sub>h</sub>)

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

Table 24: Entry Description (1C00<sub>h</sub>)

### 3.1.10 Objects 1C12<sub>h</sub>: Sync Manager 2 PDO Assignment

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

Object Description			
Index	Name	Object Type	Data Type
1C12 <sub>h</sub>	Sync manager 2 PDO assignment	RECORD	PDO assignment

Table 25: Object Description (1C12<sub>h</sub>)



Entry Description				
Sub-index	Description	Access	Value Range	Default Value
00 <sub>h</sub>	Number of assigned PDOs	rw	0...1	1
01 <sub>h</sub>	PDO mapping index of assigned RPDO	rw	UNSIGNED16	1600 <sub>h</sub>

Table 26: Entry Description (1C12<sub>h</sub>)

### 3.1.11 Objects 1C13<sub>h</sub>: Sync Manager 3 PDO Assignment

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

Object Description			
Index	Name	Object Type	Data Type
1C13 <sub>h</sub>	Sync manager 3 PDO assignment	RECORD	PDO assignment

Table 27: Object Description (1C13<sub>h</sub>)

Entry Description				
Sub-index	Description	Access	Value Range	Default Value
00 <sub>h</sub>	Number of assigned PDOs	rw	0...1	1
01 <sub>h</sub>	PDO mapping index of assigned TPDO	rw	UNSIGNED16	1A00 <sub>h</sub>

Table 28: Entry Description (1C13<sub>h</sub>)



## 4 Manufacturer specific area

The manufacturer segment contains manufacturer specific objects. These objects control the special features of the Trinamic Motion Control device TMCC160.

### 4.1 Detailed object specifications

#### 4.1.1 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 Type	Data Type
2005 <sub>h</sub>	Limit switches	Variable	UNSIGNED32

Table 29: Object Description (2005<sub>h</sub>)

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

Table 30: Entry Description (2005<sub>h</sub>)

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

Table 31: Bit Definitions (2005<sub>h</sub>)



### 4.1.2 Object 200D<sub>h</sub>: Status Flags

This object provides information about the actual module status flags. (0: not active, 1: active).

This object is organized bit-wise. The bits have the following meaning:

- Bit 0: OVERCURRENT
- Bit 1: UNDERVOLTAGE
- Bit 2: OVERVOLTAGE
- Bit 3: OVERTEMPERATURE
- Bit 4: MOTORHALTED
- Bit 5: HALLERROR
- Bit 6: DRIVER\_ERROR
- Bit 7: INIT\_ERROR
- Bit 8: STOP\_MODE
- Bit 9: VELOCITY\_MODE
- Bit 10: POSITION\_MODE
- Bit 11: TORQUE\_MODE
- Bit 12: EMERGENCYSTOP
- Bit 13: FREERUNNING
- Bit 14: POSITION\_END
- Bit 15: MODULE\_INITIALIZED
- Bit 16: unused
- Bit 17: IIT\_EXCEEDED

Object Description			
Index	Name	Object Type	Data Type
200D <sub>h</sub>	Status Flags	Variable	UNSIGNED32

Table 32: Object Description (200D<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Status Flags	no	0	4294967295	0		R

Table 33: Entry Description (200D<sub>h</sub>)

### 4.1.3 Object 200E<sub>h</sub>: Supply Voltage

The actual supply voltage.

Object Description			
Index	Name	Object Type	Data Type
200E <sub>h</sub>	Supply Voltage	Variable	UNSIGNED32

Table 34: Object Description (200E<sub>h</sub>)



Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Supply Voltage	no	0	480	240	[100mV]	R

Table 35: Entry Description (200E<sub>h</sub>)

#### 4.1.4 Object 200F<sub>h</sub>: Driver Temperatur

The actual temperature of the motor driver.

Object Description			
Index	Name	Object Type	Data Type
200F <sub>h</sub>	Driver Temperatur	Variable	SIGNED32

Table 36: Object Description (200F<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Driver Temperature	no	-20	150	0	[degree]	R

Table 37: Entry Description (200F<sub>h</sub>)

#### 4.1.5 Object 2010<sub>h</sub>: Motor Settings

Object Description			
Index	Name	Object Type	Data Type
2010 <sub>h</sub>	Motor Settings	Variable	Record

Table 38: Object Description (2010<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	MotorPoles	no	2	254	8		RW

Table 39: Entry Description (2010<sub>h</sub>)



### 4.1.6 Object 2020<sub>h</sub>: Limits

Object Description			
Index	Name	Object Type	Data Type
2020 <sub>h</sub>	Limits	Variable	Record

Table 40: Object Description (2020<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	MaxTorque	no	0	90000	4000	[mA] (peak)	RW
2	MaxVelocity	no	0	200000	4000	[rpm]	RW
3	MaxAcceleration	no	0	100000	2000	[rpm/s]	RW

Table 41: Entry Description (2020<sub>h</sub>)

### 4.1.7 Object 2030<sub>h</sub>: Torque Mode Settings

Object Description			
Index	Name	Object Type	Data Type
2030 <sub>h</sub>	Torque Mode Settings	Variable	Record

Table 42: Object Description (2030<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	ActualCurrent	no	-2147483648	2147483647	0	[mA] (peak)	R
2	TargetCurrent	no	-90000	90000	0	[mA] (peak)	R
3	RampTargetCurrent	no	-90000	90000	0	[mA] (peak)	R
4	P_Parameter	no	0	65535	0		RW
5	I_Parameter	no	0	65535	0		RW
6	PI_Torque_Error	no	-2147483648	2147483647	0	[mA]	R
7	PI_Torque_Error_Sum	no	-2147483648	2147483647	0		R
8	PI_Flux_Error	no	-2147483648	2147483647	0	[mA]	R
9	PI_Flux_Error_Sum	no	-2147483648	2147483647	0		R

Table 43: Entry Description (2030<sub>h</sub>)



### 4.1.8 Object 2040<sub>h</sub>: Velocity Mode Settings

Object Description			
Index	Name	Object Type	Data Type
2040 <sub>h</sub>	Velocity Mode Settings	Variable	Record

Table 44: Object Description (2040<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	ActualVelocity	no	-2147483648	2147483647	0	[rpm]	R
2	TargetVelocity	no	-200000	200000	0	[rpm]	R
3	RampTargetVelocity	no	-2147483648	2147483647	0	[rpm]	R
4	MotorHaltedVelocity	no	0	200000	5	[rpm]	RW
5	P_Parameter	no	0	65535	0		RW
6	I_Parameter	no	0	65535	0		RW
7	PI_Velocity_Error	no	-2147483648	2147483647	0	[rpm]	R
8	PI_Velocity_Error_Sum	no	-2147483648	2147483647	0		R

Table 45: Entry Description (2040<sub>h</sub>)

### 4.1.9 Object 2050<sub>h</sub>: Position Mode Settings

Object Description			
Index	Name	Object Type	Data Type
2050 <sub>h</sub>	Position Mode Settings	Variable	Record

Table 46: Object Description (2050<sub>h</sub>)



Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	ActualPosition	no	-2147483648	2147483647	0		RW
2	TargetPosition	no	-2147483648	2147483647	0		R
3	RampTargetPosition	no	-2147483648	2147483647	0		R
4	P_Parameter	no	0	65535	0		RW
5	PI_Position_Error	no	-2147483648	2147483647	0		R
6	TargetReachedVelocity	no	0	200000	500	[rpm]	RW
7	TargetReachedDistance	no	0	100000	5		RW

Table 47: Entry Description (2050<sub>h</sub>)

#### 4.1.10 Object 2055<sub>h</sub>: Commutation Mode

Select a commutation mode that fits best to your motor’s sensors.

- 6 : FOC (hall sensor)
- 7 : FOC (encoder)
- 8 : FOC (controlled)

Object Description			
Index	Name	Object Type	Data Type
2055 <sub>h</sub>	Commutation Mode	Variable	Record

Table 48: Object Description (2055<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Commutation Mode	no	6	8	6		RW

Table 49: Entry Description (2055<sub>h</sub>)

#### 4.1.11 Object 2056<sub>h</sub>: Velocity Ramp Mode

An activated ramp allows a defined acceleration for velocity and position mode.

Object Description			
Index	Name	Object Type	Data Type
2056 <sub>h</sub>	Velocity Ramp Mode	Variable	UNSIGNED8

Table 50: Object Description (2056<sub>h</sub>)





Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Velocity Ramp Mode	no	0	1	1		RW

Table 51: Entry Description (2056<sub>h</sub>)

#### 4.1.12 Object 2060<sub>h</sub>: Open Loop Settings

Object Description			
Index	Name	Object Type	Data Type
2060 <sub>h</sub>	Open Loop Settings	Variable	Record

Table 52: Object Description (2060<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	ActualAngle	no	-32768	32767	0		R
2	OpenLoopCurrent	no	0	90000	1500	[mA] (peak)	RW

Table 53: Entry Description (2060<sub>h</sub>)

#### 4.1.13 Object 2070<sub>h</sub>: Hall Sensor Settings

Object Description			
Index	Name	Object Type	Data Type
2070 <sub>h</sub>	Hall Sensor Settings	Variable	Record

Table 54: Object Description (2070<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	ActualAngle	no	-32768	32767	0		R
2	Inversion	no	0	1	0		RW
3	Interpolation	no	0	1	0		RW

Table 55: Entry Description (2070<sub>h</sub>)



#### 4.1.14 Object 2080<sub>h</sub>: ABN Encoder Settings

Object Description			
Index	Name	Object Type	Data Type
2080 <sub>h</sub>	ABN Encoder Settings	Variable	Record

Table 56: Object Description (2080<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	ActualAngle	no	-32768	32767	0		R
2	StepsPerRotation	no	0	65535	4000		RW
3	Offset	no	0	65535	0		RW
4	Direction	no	0	1	0		RW
5	InitMode	no	0	2	1		RW
6	InitDelay	no	0	10000	1000	[ms]	RW
7	InitVelocity	no	-200000	200000	100	[rpm]	RW

Table 57: Entry Description (2080<sub>h</sub>)

#### 4.1.15 Object 2090<sub>h</sub>: Brake Chopper Settings

Object Description			
Index	Name	Object Type	Data Type
2090 <sub>h</sub>	Brake Chopper Settings	Variable	Record

Table 58: Object Description (2090<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	EnableBrakeChopper	no	0	1	0		RW
2	BrakeChopperVoltageLimit	no	50	480	260	[100mV]	RW
3	BrakeChopperHysteresis	no	0	50	5	[100mV]	RW

Table 59: Entry Description (2090<sub>h</sub>)



#### 4.1.16 Object 2095<sub>h</sub>: IIT Settings

Object Description			
Index	Name	Object Type	Data Type
2095 <sub>h</sub>	IIT Settings	Variable	Record

Table 60: Object Description (2095<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	ThermalWindingTimeConstant	no	0	300000	30000	[ms]	RW
2	IItLimit	no	0	270000000	5880000		RW
3	IItSum	no	0	4294967295	0		R
4	IItExceededCounter	no	0	4294967295	0		RW
5	ClearIItExceededFlag	no	0	0	0		W

Table 61: Entry Description (2095<sub>h</sub>)

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

This object shows the home offset. The value is given in encoder or hall increments.

Object Description			
Index	Name	Object Type	Data Type
2100 <sub>h</sub>	Home Offset Display	Variable	SIGNED32

Table 62: Object Description (2100<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Home Offset Display	no	-2147483648	2147483647	0		R

Table 63: Entry Description (2100<sub>h</sub>)

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

- Bit0: Enable
- Bit1: Right limit switch status
- Bit2: Left limit switch status



Object Description			
Index	Name	Object Type	Data Type
2702 <sub>h</sub>	Digital Inputs	Variable	UNSIGNED32

Table 64: Object Description (2702<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Digital Inputs	no	0	7	0		R

Table 65: Entry Description (2702<sub>h</sub>)

#### 4.1.19 Object 2706<sub>h</sub>: Store Axis Parameter

Writing the save signature to this object permanently saves all axis parameters. The save signature is 65766173h.

Object Description			
Index	Name	Object Type	Data Type
2706 <sub>h</sub>	Store Axis Parameter	Variable	UNSIGNED32

Table 66: Object Description (2706<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Store Axis Parameter	no	0	4294967295	0		RW

Table 67: Entry Description (2706<sub>h</sub>)

#### 4.1.20 Object 270E<sub>h</sub>: Analog Inputs

Object Description			
Index	Name	Object Type	Data Type
270E <sub>h</sub>	Analog Inputs	Variable	Record

Table 68: Object Description (270E<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	ADC_IN_0	no	0	65535	0		R
2	ADC_SINGLE_SHUNT	no	0	65535	32767		R
3	ADC_phase_A	no	0	65535	32767		R
4	ADC_phase_B	no	0	65535	32767		R
5	ADC_VSupply	no	0	65535	0		R
6	ADC_Temp	no	0	65535	0		R

Table 69: Entry Description (270E<sub>n</sub>)



## 5 Profile specific area

The profile segment contains CiA-402 standard motion control objects. These objects control the motion control functions of the TMCC160. Since it is not possible to operate the modes in parallel, the user is able to activate the required function by selecting a mode of operation. The control device writes to the modes of operation object in order to select the operation mode. The drive device provides the modes of operation display object to indicate the actual activated operation mode. Controlword, statusword, and set-points are used mode-specific. This implies the responsibility of the control device to avoid inconsistencies and erroneous behavior.

The following operating modes (selectable via object 6060<sub>n</sub>, please see 5.1.1) are implemented on the TMCC160:

- Profile position mode (pp)
- Profile velocity mode (pv)
- Cyclic synchronous torque mode (cst)



## 5.1 Detailed object specifications

### 5.1.1 Object 6060<sub>h</sub>: Modes of operation

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

Value Definition	
Value	Mode
0	No mode
1	Profile position mode (pp)
3	Profile velocity mode (pv)
6	Homing mode (hm)
10	Cyclic synchronous torque mode (cst)

Table 70: Value Description (6060<sub>h</sub>)

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

Object Description			
Index	Name	Object Type	Data Type
6060 <sub>h</sub>	Modes of operation	Variable	SIGNED8

Table 71: Object Description (6060<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	refer to CiA-402	0/1/3/10	0

Table 72: Entry Description (6060<sub>h</sub>)

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

This object shows the operating mode that is currently set.



Value Definition	
Value	Mode
0	No mode
1	Profile position mode (pp)
3	Profile velocity mode (pv)
6	Homing mode (hm)
10	Cyclic synchronous torque mode (cst)

Table 73: Value Description (6061<sub>h</sub>)

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

Object Description			
Index	Name	Object Type	Data Type
6061 <sub>h</sub>	Modes of operation display	Variable	SIGNED8

Table 74: Object Description (6061<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	refer to CiA-402	0/1/3/10	0

Table 75: Entry Description (6061<sub>h</sub>)

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

This object provides information on the supported drive modes (0: not supported, 1: supported). This object is organized bit-wise. The bits have the following meaning:

- Bit 0: profile position mode
- Bit 1: velocity mode
- Bit 2: profile velocity mode
- Bit 3: profile torque mode
- Bit 4: reserved
- Bit 5: homing mode
- Bit 6: interpolated position mode
- Bit 7: cyclic synchronous position mode
- Bit 8: cyclic synchronous velocity mode
- Bit 9: cyclic synchronous torque mode
- Bit 10-15: reserved
- Bit 16-31: manufacturer-specific





Object Description			
Index	Name	Object Type	Data Type
6502 <sub>h</sub>	Supported Drive Modes	Variable	UNSIGNED32

Table 76: Object Description (6502<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Supported Drive Modes	no	0	4294967295	0		R

Table 77: Entry Description (6502<sub>h</sub>)

## 6 Profile Position Mode

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

Please refer to object 6060<sub>h</sub> (section 5.1.1) for information about how to choose an operation mode. Object 6061<sub>h</sub> (section 5.1.2) shows the operation mode that is set.

### 6.1 Detailed Object Specifications

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



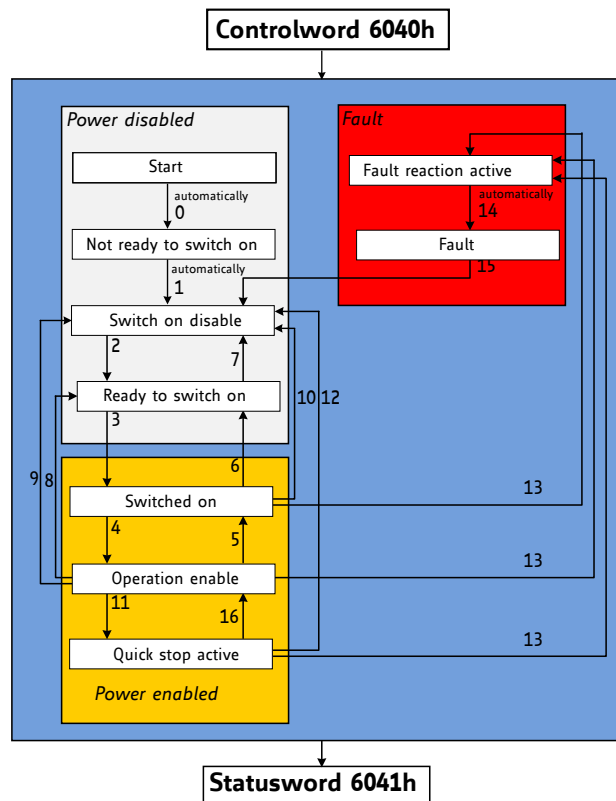


Figure 3: DS402 Finite State Machine

Notes on state transitions:

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

## 6.2 Detailed Object Specifications

### 6.2.1 Object 6040<sub>n</sub>: Control Word

This object indicates the received command controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to figure 3 for detailed information.



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

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

Table 78: Structure of the Control Word in pp Mode

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

Table 79: Operation Mode specific Bits in pp Mode

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

Table 80: Command Coding

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

Table 81: Object Description (6040<sub>h</sub> in pp Mode)



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

Table 82: Entry Description (6040<sub>h</sub> in pp Mode)

### 6.2.2 Object 6041<sub>h</sub>: Status Word

This object provides the status of the PDS FSA. It reflects the status of the CiA-402 state machine. Please refer to figure 3 for detailed information. The object is structured as defined below. For more information about the coding please refer to the CANopen Drives and motion control device profile, part 2.

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

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

Table 83: Structure of the Staus Word in pp Mode

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

Table 84: Trinamic Specific Bits

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

Table 85: Operation Mode specific Bits in pp Mode



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

Table 86: State Coding

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

Table 87: Object Description (6041<sub>h</sub> in pp Mode)

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

Table 88: Entry Description (6041<sub>h</sub> in pp Mode)

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

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

Object Description			
Index	Name	Object Type	Data Type
6062 <sub>h</sub>	Position Demand Value	Variable	SIGNED32

Table 89: Object Description (6062<sub>h</sub>)



Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Position Demand Value	yes	-2147483648	2147483647	0		R

Table 90: Entry Description (6062<sub>h</sub>)

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

This object provides the actual position value of the motor.

Object Description			
Index	Name	Object Type	Data Type
6063 <sub>h</sub>	Position Actual Internal Value	Variable	SIGNED32

Table 91: Object Description (6063<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Position Actual Internal Value	no	-2147483648	2147483647	0		R

Table 92: Entry Description (6063<sub>h</sub>)

#### 6.2.5 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 6063h.

Object Description			
Index	Name	Object Type	Data Type
6064 <sub>h</sub>	Position Actual Value	Variable	SIGNED32

Table 93: Object Description (6064<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Position Actual Value	yes	-2147483648	2147483647	0		R

Table 94: Entry Description (6064<sub>h</sub>)

### 6.2.6 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 FFFFFFFFh, 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 (6062h) has reached the target position (6064h). 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 (6064h) is within (target\_position - position\_window) and (target\_position + position\_window).

Object Description			
Index	Name	Object Type	Data Type
6067 <sub>h</sub>	Position Window	Variable	UNSIGNED32

Table 95: Object Description (6067<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Position Window	no	0	4294967295	4294967295		RW

Table 96: Entry Description (6067<sub>h</sub>)

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

This object shows the actual velocity value derived from the velocity sensor.

Object Description			
Index	Name	Object Type	Data Type
606C <sub>h</sub>	Velocity Actual Value	Variable	SIGNED32

Table 97: Object Description (606C<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Velocity Actual Value	yes	-2147483648	2147483647	0	[rpm]	R

Table 98: Entry Description (606C<sub>h</sub>)

### 6.2.8 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 actual settings of motion control parameters (such as velocity, acceleration, deceleration, etc.). The value of this object is interpreted as absolute or relative depending on the abs/rel flag in the controlword.



Object Description			
Index	Name	Object Type	Data Type
607A <sub>h</sub>	Target Position	Variable	SIGNED32

Table 99: Object Description (607A<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Target Position	yes	-2147483648	2147483647	0		RW

Table 100: Entry Description (607A<sub>h</sub>)

### 6.2.9 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

Object Description			
Index	Name	Object Type	Data Type
607D <sub>h</sub>	Software Position Limit	Array	SIGNED32

Table 101: Object Description (607D<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Min Position Limit	no	-2147483648	2147483647	-2147483648		RW
2	Max Position Limit	no	-2147483648	2147483647	2147483647		RW

Table 102: Entry Description (607D<sub>h</sub>)

### 6.2.10 Object 607F<sub>h</sub>: Max Profile Velocity

Max. absolute velocity for velocity and positioning mode.





Object Description			
Index	Name	Object Type	Data Type
607F <sub>h</sub>	Max Profile Velocity	Variable	UNSIGNED32

Table 103: Object Description (607F<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	MaxVelocity	no	0	200000	4000	[rpm]	RW

Table 104: Entry Description (607F<sub>h</sub>)

### 6.2.11 Object 6081<sub>h</sub>: Profile Velocity (pp)

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.

Object Description			
Index	Name	Object Type	Data Type
6081 <sub>h</sub>	Profile Velocity (pp)	Variable	UNSIGNED32

Table 105: Object Description (6081<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Profile Velocity	no	0	200000	0	[rpm]	RW

Table 106: Entry Description (6081<sub>h</sub>)

### 6.2.12 Object 6082<sub>h</sub>: End Velocity

This object indicates the configured velocity normally attained at the end of the deceleration ramp during a profiled motion and is valid for both directions of motion. The end velocity is the velocity used when reaching the new position.

Object Description			
Index	Name	Object Type	Data Type
6082 <sub>h</sub>	End Velocity	Variable	SIGNED32

Table 107: Object Description (6082<sub>h</sub>)



Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	End Velocity	no	-200000	200000	0	[rpm]	RW

Table 108: Entry Description (6082<sub>h</sub>)

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

This object indicates the configured acceleration. Object 6083h sets the maximum acceleration to be used in profile positioning mode, and profile velocity mode.

Object Description			
Index	Name	Object Type	Data Type
6083 <sub>h</sub>	Profile Acceleration	Variable	UNSIGNED32

Table 109: Object Description (6083<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Profile Acceleration	no	0	100000	2000	[rpm/s]	RW

Table 110: Entry Description (6083<sub>h</sub>)

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

This object indicates the configured deceleration.

Object Description			
Index	Name	Object Type	Data Type
6084 <sub>h</sub>	Profile Deceleration	Variable	UNSIGNED32

Table 111: Object Description (6084<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Profile Deceleration	no	0	100000	2000	[rpm/s]	RW

Table 112: Entry Description (6084<sub>h</sub>)

### 6.2.15 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).

Object Description			
Index	Name	Object Type	Data Type
6085 <sub>h</sub>	Quick Stop Deceleration	Variable	UNSIGNED32

Table 113: Object Description (6085<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Quick Stop Deceleration	no	0	100000	2000	[rpm/s]	RW

Table 114: Entry Description (6085<sub>h</sub>)



### 6.3 How to move a Motor in pp Mode

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

- Select pp mode by writing 1 to object 6060<sub>h</sub> (Modes\_of\_Operation).
- Write 6 to object 6040<sub>h</sub> (Controlword) to switch to READY\_TO\_SWITCH\_ON state.
- Write 7 to object 6040<sub>h</sub> to switch to SWITCHED\_ON state.
- Write the desired target position (e.g. 50000) to object 607A<sub>h</sub> (Target\_Position).
- Write 15 to object 6040<sub>h</sub> to switch to OPERATION\_ENABLED state. The motor now accelerates to the target position.



## 7 Profile Velocity Mode

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

The profile velocity mode covers the following sub-functions:

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

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

- Profile velocity
- Profile acceleration
- Motion profile type

### 7.1 Detailed Object Specifications

#### 7.1.1 Object 6040<sub>n</sub>: Control Word

This object indicates the received command controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to figure 3 for detailed information.

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

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

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

Table 115: Structure of the Control Word in pv Mode



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

Table 116: Command Coding

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

Table 117: Object Description (6040<sub>h</sub> in pv Mode)

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

Table 118: Entry Description (6040<sub>h</sub> in pv Mode)

### 7.1.2 Object 6041<sub>h</sub>: Status Word

This object provides the status of the PDS FSA. It reflects the status of the CiA-402 state machine. Please refer to figure 3 for detailed information. The object is structured as defined below. For more information about the coding please refer to the CANopen Drives and motion control device profile, part 2.

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

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

Table 119: Structure of the Status Word in pv Mode



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

Table 120: Trinamic Specific Bits

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

Table 121: Operation Mode specific Bits in pv Mode

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

Table 122: State Coding

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

Table 123: Object Description (6041<sub>h</sub> in pv Mode)



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

Table 124: Entry Description (6041<sub>h</sub> in pv Mode)

### 7.1.3 Object 606B<sub>h</sub>: Velocity Demand Value

This objects provides the output value of the trajectory generator.

Object Description			
Index	Name	Object Type	Data Type
606B <sub>h</sub>	Velocity Demand Value	Variable	SIGNED32

Table 125: Object Description (606B<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Velocity Demand Value	yes	-2147483648	2147483647	0		R

Table 126: Entry Description (606B<sub>h</sub>)

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

This object shows the actual velocity value derived from the velocity sensor.

Object Description			
Index	Name	Object Type	Data Type
606C <sub>h</sub>	Velocity Actual Value	Variable	SIGNED32

Table 127: Object Description (606C<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Velocity Actual Value	yes	-2147483648	2147483647	0	[rpm]	R

Table 128: Entry Description (606C<sub>h</sub>)





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

This object indicates the configured acceleration. Object 6083h sets the maximum acceleration to be used in profile positioning mode, and profile velocity mode.

Object Description			
Index	Name	Object Type	Data Type
6083 <sub>h</sub>	Profile Acceleration	Variable	UNSIGNED32

Table 129: Object Description (6083<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Profile Acceleration	no	0	100000	2000	[rpm/s]	RW

Table 130: Entry Description (6083<sub>h</sub>)

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

This object indicates the configured target velocity and is used as input for the trajectory generator. Object 60FFh 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 6083h and 6084h.

Object Description			
Index	Name	Object Type	Data Type
60FF <sub>h</sub>	Target Velocity	Variable	SIGNED32

Table 131: Object Description (60FF<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Target Velocity	yes	-200000	200000	0	[rpm]	RW

Table 132: Entry Description (60FF<sub>h</sub>)



## 7.2 How to move a Motor in pv Mode

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

- Select pv mode by writing 3 to object 6060<sub>h</sub> (Modes\_of\_Operation).
- Write 6 to object 6040<sub>h</sub> (Controlword) to switch to READY\_TO\_SWITCH\_ON state.
- Write 7 to object 6040<sub>h</sub> to switch to SWITCHED\_ON state.
- Write the desired target velocity (e.g. 2000) to object 60FF<sub>h</sub> (Target\_Velocity).
- Write 15 to object 6040<sub>h</sub> to switch to OPERATION\_ENABLED state. The motor now accelerates to the target velocity.
- Stop the motor by writing 0 to object 60FF<sub>h</sub>.



## 8 Homing mode

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

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

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

Figure 4 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<sub>h</sub> allows displacing the zero in point the coordinate system for the home position.

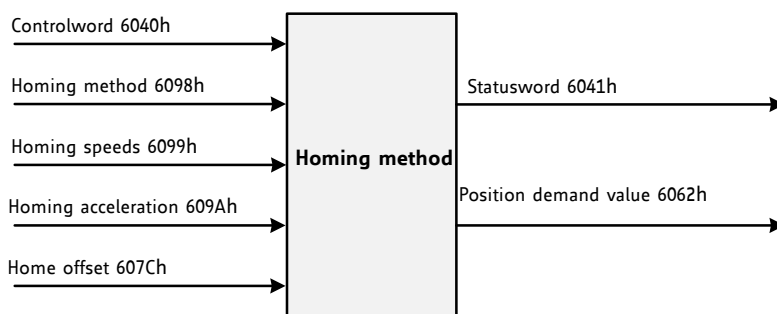


Figure 4: Homing Mode Function

Choosing a homing mode determines the following things:

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

The home position and the zero position are offset by the home offset (see object 607C<sub>h</sub>, section 8.2.4).

There are four sources of homing signals available:

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

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



## 8.1 Homing Methods

The TMCC160 supports a subset of different standard CANopen homing methods. The homing method that is to be used can be chosen via object 6098<sub>h</sub> (section 8.2.5).

Supported Homing Methods	
Method	Description
0	No homing (default value for object 6098 <sub>h</sub> ).
17	Search the left end switch.
18	Search the right end switch.
35	The actual position is used as home position. All position values (objects 6062h, 6063h, and 6064h) are set to zero, but the motor will not move.

Table 133: Supported CANopen Homing Methods

When using homing methods that need end switch inputs or home switch inputs please take care of their configuration (object 2005<sub>h</sub>).

### 8.1.1 Homing Method 17 and 18: 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.

Homing Methods 17... 21	
Method	Description
17	Search the left end switch. (Similar to method 1)
18	Search the right end switch. (Similar to method 2)

Table 134: Homing Methods 17 – 21

### 8.1.2 Homing Method 35: Current Position as Home Position

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



## 8.2 Detailed Object Specifications

### 8.2.1 Object 6040<sub>h</sub>: Control Word

This object indicates the received command controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to figure 3 for detailed information.

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

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

Table 135: Structure of the Control Word in hm Mode

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

Table 136: Operation Mode specific Bits in hm Mode

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

Table 137: Command Coding



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

Table 138: Object Description (6040<sub>h</sub> in hm Mode)

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

Table 139: Entry Description (6040<sub>h</sub> in hm Mode)

### 8.2.2 Object 6041<sub>h</sub>: Status Word

This object provides the status of the PDS FSA. It reflects the status of the CiA-402 state machine. Please refer to figure 3 for detailed information. The object is structured as defined below. For more information about the coding please refer to the CANopen Drives and motion control device profile, part 2.

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

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

Table 140: Structure of the Status Word in hm Mode

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

Table 141: Trinamic Specific Bits



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

Table 142: Operation Mode specific Bits in hm Mode

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

Table 143: State Coding

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

Table 144: Object Description (6041<sub>h</sub> in hm Mode)

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

Table 145: Entry Description (6041<sub>h</sub> in hm Mode)

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

This object shows the actual velocity value derived from the velocity sensor.



Object Description			
Index	Name	Object Type	Data Type
606C <sub>h</sub>	Velocity Actual Value	Variable	SIGNED32

Table 146: Object Description (606C<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Velocity Actual Value	yes	-2147483648	2147483647	0	[rpm]	R

Table 147: Entry Description (606C<sub>h</sub>)

### 8.2.4 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. Negative values indicate the opposite direction.

Object Description			
Index	Name	Object Type	Data Type
607C <sub>h</sub>	Home Offset	Variable	SIGNED32

Table 148: Object Description (607C<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Home Offset	no	-2147483648	2147483647	0		RW

Table 149: Entry Description (607C<sub>h</sub>)

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

The actual homing method.

Object Description			
Index	Name	Object Type	Data Type
6098 <sub>h</sub>	Homing Method	Variable	SIGNED8

Table 150: Object Description (6098<sub>h</sub>)





Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Homing Method	no	0	35	0		RW

Table 151: Entry Description (6098<sub>h</sub>)

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

This object indicates the configured speeds used during fast and slow homing procedure. 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>) to search for the exact switch point. When the switch point has been found the motor will be stopped at that point.

Object Description			
Index	Name	Object Type	Data Type
6099 <sub>h</sub>	Homing Speeds	Array	UNSIGNED32

Table 152: Object Description (6099<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Fast Homing Speed	no	0	4294967295	0		RW
2	Slow Homing Speed	no	0	4294967295	0		RW

Table 153: Entry Description (6099<sub>h</sub>)

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

This object indicates the configured acceleration and deceleration to be used during homing operation.

Object Description			
Index	Name	Object Type	Data Type
609A <sub>h</sub>	Homing Acceleration	Variable	UNSIGNED32

Table 154: Object Description (609A<sub>h</sub>)



Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Homing Acceleration	no	0	100000	2000	[rpm/s]	RW

Table 155: Entry Description (609A<sub>h</sub>)

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

This object shows the home offset. The value is given in encoder or hall increments.

Object Description			
Index	Name	Object Type	Data Type
2100 <sub>h</sub>	Home Offset Display	Variable	SIGNED32

Table 156: Object Description (2100<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Home Offset Display	no	-2147483648	2147483647	0		R

Table 157: Entry Description (2100<sub>h</sub>)



### 8.3 How to start a Homing in hm Mode

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

- Select hm mode by writing 6 to object 6060<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 17 (or 18) by writing 17 (or 18) to object 6098<sub>h</sub>.
- Set the homing speeds by writing e.g. 500 to object 6099<sub>h</sub> sub index 1 and e.g. 200 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.



## 9 Cyclic synchronous Torque Mode

The cyclic synchronous torque mode is used to directly control the torque of the motor, without the need for position or velocity control. It contains limit functions, but not a trajectory generator.

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

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

### 9.1 Detailed Object Specifications

#### 9.1.1 Object 6040<sub>n</sub>: Control Word

This object indicates the received command controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to figure 3 for detailed information. The cyclic synchronous torque mode does not use any mode specific bits of the control word.

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

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

Table 158: Structure of the Control Word in cst Mode

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

Table 159: Command Coding



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

Table 160: Object Description (6040<sub>h</sub> in cst Mode)

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

Table 161: Entry Description (6040<sub>h</sub> in cst Mode)

### 9.1.2 Object 6041<sub>h</sub>: Status Word

This object provides the status of the PDS FSA. It reflects the status of the CiA-402 state machine. Please refer to figure 3 for detailed information. The object is structured as defined below.

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

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

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

Table 162: Structure of the Status Word in cst Mode

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

Table 163: Trinamic Specific Bits



Operation Mode specific Bits in cst Mode		
Bit	Name	Definition
10	Reserved	Not used.
12	Target torque ignored	0: Target torque ignored. 1: Target torque used as input to control loop.
13	Reserved	Not used.

Table 164: Operation Mode specific Bits in cst Mode

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

Table 165: State Coding

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

Table 166: Object Description (6041<sub>h</sub> in cst Mode)

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

Table 167: Entry Description (6041<sub>h</sub> in cst Mode)

### 9.1.3 Object 6071<sub>h</sub>: Target Torque

This object gives the target motor current.



Object Description			
Index	Name	Object Type	Data Type
6071 <sub>h</sub>	Target Torque	Variable	SIGNED32

Table 168: Object Description (6071<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Target Torque	yes	-90000	90000	0	[mA]	RW

Table 169: Entry Description (6071<sub>h</sub>)

### 9.1.4 Object 6072<sub>h</sub>: Max Torque

Max. allowed absolute motor current. \*This value can be temporarily exceeded marginal due to the operation of the current regulator.

Object Description			
Index	Name	Object Type	Data Type
6072 <sub>h</sub>	Max Torque	Variable	UNSIGNED32

Table 170: Object Description (6072<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Max Torque	no	0	90000	4000	[mA] (peak)	RW

Table 171: Entry Description (6072<sub>h</sub>)

### 9.1.5 Object 6074<sub>h</sub>: Torque Demand Value

This object gives the internal target motor current.

Object Description			
Index	Name	Object Type	Data Type
6074 <sub>h</sub>	Torque Demand Value	Variable	SIGNED32

Table 172: Object Description (6074<sub>h</sub>)



Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Torque Demand Value	yes	-90000	90000	0	[mA]	R

Table 173: Entry Description (6074<sub>h</sub>)

### 9.1.6 Object 6077<sub>h</sub>: Torque Actual Value

The actual motor current.

Object Description			
Index	Name	Object Type	Data Type
6077 <sub>h</sub>	Torque Actual Value	Variable	SIGNED32

Table 174: Object Description (6077<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Torque Actual Value	yes	-2147483648	2147483647	0		R

Table 175: Entry Description (6077<sub>h</sub>)

### 9.1.7 Object 60B2<sub>h</sub>: Torque offset

The actual set torque offset.

Object Description			
Index	Name	Object Type	Data Type
60B2 <sub>h</sub>	Torque offset	Variable	SIGNED32

Table 176: Object Description (60B2<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Torque offset	no	-2147483648	2147483647	0		RW

Table 177: Entry Description (60B2<sub>h</sub>)



## 9.2 How to move a Motor in cst Mode

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

- Select cst mode by writing 10 to object 6060<sub>h</sub> (Modes\_of\_Operation).
- Write 6 to object 6040<sub>h</sub> (Controlword) to switch to READY\_TO\_SWITCH\_ON state.
- Write 7 to object 6040<sub>h</sub> to switch to SWITCHED\_ON state.
- Write the desired torque (e.g. 1000) to object 6071<sub>h</sub> (Target\_Torque).
- Write 15 to object 6040<sub>h</sub> to switch to OPERATION\_ENABLED state to start the motor.
- To stop the motor, write 0 to object 6071<sub>h</sub>.



## 10 Emergency Messages (EMCY)

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

Emergency Messages (EMCY) of the TMCC160						
Error code	Additional byte					Description
	1	2	3	4	5	
0000 <sub>h</sub>	0	0	0	0	0	<b>Fault reset</b> The fault reset command has been executed.
1000 <sub>h</sub>	1	0	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.
1000 <sub>h</sub>	2	0	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.
2310 <sub>h</sub>	0	0	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.
2311 <sub>h</sub>	0	0	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.
2312 <sub>h</sub>	0	0	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.
3230 <sub>h</sub>	0	0	0	0	0	<b>stallGuard2 error</b> The actual load value exceeds the stallGuard2 limit.
4310 <sub>h</sub>	1	0	0	0	0	<b>Overtemperature pre-warning</b> The temperature in the motor driver exceeds the pre-warning limit.
4310 <sub>h</sub>	2	0	0	0	0	<b>Overtemperature error</b> The motor driver has been switched off because the temperature limit has been exceeded.
5441 <sub>h</sub>	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.
6320 <sub>h</sub>	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.



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

Table 178: Emergency Messages (EMCY)



## 11 Figures Index

1	NMT State Machine . . . . .	9	3	DS402 Finite State Machine . . . . .	34
2	Device Model . . . . .	10	4	Homing Mode Function . . . . .	51



## 12 Tables Index

1	Abbreviations used in this Manual . . .	6	53	Entry Description (2060 <sub>h</sub> ) . . . . .	25
2	Service Primitives . . . . .	7	54	Object Description (2070 <sub>h</sub> ) . . . . .	25
3	Service Types . . . . .	8	55	Entry Description (2070 <sub>h</sub> ) . . . . .	25
4	Object Dictionary . . . . .	11	56	Object Description (2080 <sub>h</sub> ) . . . . .	26
5	Object Description (1000 <sub>h</sub> ) . . . . .	12	57	Entry Description (2080 <sub>h</sub> ) . . . . .	26
6	Entry Description (1000 <sub>h</sub> ) . . . . .	12	58	Object Description (2090 <sub>h</sub> ) . . . . .	26
7	Object Description (1001 <sub>h</sub> ) . . . . .	12	59	Entry Description (2090 <sub>h</sub> ) . . . . .	26
8	Entry Description (1001 <sub>h</sub> ) . . . . .	12	60	Object Description (2095 <sub>h</sub> ) . . . . .	27
9	Error Register Bits . . . . .	13	61	Entry Description (2095 <sub>h</sub> ) . . . . .	27
10	Object Description (1008 <sub>h</sub> ) . . . . .	13	62	Object Description (2100 <sub>h</sub> ) . . . . .	27
11	Entry Description (1008 <sub>h</sub> ) . . . . .	13	63	Entry Description (2100 <sub>h</sub> ) . . . . .	27
12	Object Description (1009 <sub>h</sub> ) . . . . .	13	64	Object Description (2702 <sub>h</sub> ) . . . . .	28
13	Entry Description (1009 <sub>h</sub> ) . . . . .	14	65	Entry Description (2702 <sub>h</sub> ) . . . . .	28
14	Object Description (100A <sub>h</sub> ) . . . . .	14	66	Object Description (2706 <sub>h</sub> ) . . . . .	28
15	Entry Description (100A <sub>h</sub> ) . . . . .	14	67	Entry Description (2706 <sub>h</sub> ) . . . . .	28
16	Object Description (1018 <sub>h</sub> ) . . . . .	14	68	Object Description (270E <sub>h</sub> ) . . . . .	28
17	Entry Description (1018 <sub>h</sub> ) . . . . .	15	69	Entry Description (270E <sub>h</sub> ) . . . . .	29
18	Object Description (1600 <sub>h</sub> ) . . . . .	15	70	Value Description (6060 <sub>h</sub> ) . . . . .	31
19	Entry Description (1600 <sub>h</sub> ) . . . . .	15	71	Object Description (6060 <sub>h</sub> ) . . . . .	31
20	Object Description (1A00 <sub>h</sub> ) . . . . .	16	72	Entry Description (6060 <sub>h</sub> ) . . . . .	31
21	Entry Description (1A00 <sub>h</sub> ) . . . . .	16	73	Value Description (6061 <sub>h</sub> ) . . . . .	32
22	Sync Manager Communication Types	17	74	Object Description (6061 <sub>h</sub> ) . . . . .	32
23	Object Description (1C00 <sub>h</sub> ) . . . . .	17	75	Entry Description (6061 <sub>h</sub> ) . . . . .	32
24	Entry Description (1C00 <sub>h</sub> ) . . . . .	17	76	Object Description (6502 <sub>h</sub> ) . . . . .	33
25	Object Description (1C12 <sub>h</sub> ) . . . . .	17	77	Entry Description (6502 <sub>h</sub> ) . . . . .	33
26	Entry Description (1C12 <sub>h</sub> ) . . . . .	18	78	Structure of the Control Word in pp Mode . . . . .	35
27	Object Description (1C13 <sub>h</sub> ) . . . . .	18	79	Operation Mode specific Bits in pp Mode	35
28	Entry Description (1C13 <sub>h</sub> ) . . . . .	18	80	Command Coding . . . . .	35
29	Object Description (2005 <sub>h</sub> ) . . . . .	19	81	Object Description (6040 <sub>h</sub> in pp Mode)	35
30	Entry Description (2005 <sub>h</sub> ) . . . . .	19	82	Entry Description (6040 <sub>h</sub> in pp Mode)	36
31	Bit Definitions (2005 <sub>h</sub> ) . . . . .	19	83	Structure of the Staus Word in pp Mode	36
32	Object Description (200D <sub>h</sub> ) . . . . .	20	84	Trinamic Specific Bits . . . . .	36
33	Entry Description (200D <sub>h</sub> ) . . . . .	20	85	Operation Mode specific Bits in pp Mode	36
34	Object Description (200E <sub>h</sub> ) . . . . .	20	86	State Coding . . . . .	37
35	Entry Description (200E <sub>h</sub> ) . . . . .	21	87	Object Description (6041 <sub>h</sub> in pp Mode)	37
36	Object Description (200F <sub>h</sub> ) . . . . .	21	88	Entry Description (6041 <sub>h</sub> in pp Mode)	37
37	Entry Description (200F <sub>h</sub> ) . . . . .	21	89	Object Description (6062 <sub>h</sub> ) . . . . .	37
38	Object Description (2010 <sub>h</sub> ) . . . . .	21	90	Entry Description (6062 <sub>h</sub> ) . . . . .	38
39	Entry Description (2010 <sub>h</sub> ) . . . . .	21	91	Object Description (6063 <sub>h</sub> ) . . . . .	38
40	Object Description (2020 <sub>h</sub> ) . . . . .	22	92	Entry Description (6063 <sub>h</sub> ) . . . . .	38
41	Entry Description (2020 <sub>h</sub> ) . . . . .	22	93	Object Description (6064 <sub>h</sub> ) . . . . .	38
42	Object Description (2030 <sub>h</sub> ) . . . . .	22	94	Entry Description (6064 <sub>h</sub> ) . . . . .	38
43	Entry Description (2030 <sub>h</sub> ) . . . . .	22	95	Object Description (6067 <sub>h</sub> ) . . . . .	39
44	Object Description (2040 <sub>h</sub> ) . . . . .	23	96	Entry Description (6067 <sub>h</sub> ) . . . . .	39
45	Entry Description (2040 <sub>h</sub> ) . . . . .	23	97	Object Description (606C <sub>h</sub> ) . . . . .	39
46	Object Description (2050 <sub>h</sub> ) . . . . .	23	98	Entry Description (606C <sub>h</sub> ) . . . . .	39
47	Entry Description (2050 <sub>h</sub> ) . . . . .	24	99	Object Description (607A <sub>h</sub> ) . . . . .	40
48	Object Description (2055 <sub>h</sub> ) . . . . .	24	100	Entry Description (607A <sub>h</sub> ) . . . . .	40
49	Entry Description (2055 <sub>h</sub> ) . . . . .	24	101	Object Description (607D <sub>h</sub> ) . . . . .	40
50	Object Description (2056 <sub>h</sub> ) . . . . .	24	102	Entry Description (607D <sub>h</sub> ) . . . . .	40
51	Entry Description (2056 <sub>h</sub> ) . . . . .	25	103	Object Description (607F <sub>h</sub> ) . . . . .	41
52	Object Description (2060 <sub>h</sub> ) . . . . .	25			



104	Entry Description (607F <sub>h</sub> ) . . . . .	41	142	Operation Mode specific Bits in hm Mode . . . . .	55
105	Object Description (6081 <sub>h</sub> ) . . . . .	41	143	State Coding . . . . .	55
106	Entry Description (6081 <sub>h</sub> ) . . . . .	41	144	Object Description (6041 <sub>h</sub> in hm Mode)	55
107	Object Description (6082 <sub>h</sub> ) . . . . .	41	145	Entry Description (6041 <sub>h</sub> in hm Mode)	55
108	Entry Description (6082 <sub>h</sub> ) . . . . .	42	146	Object Description (606C <sub>h</sub> ) . . . . .	56
109	Object Description (6083 <sub>h</sub> ) . . . . .	42	147	Entry Description (606C <sub>h</sub> ) . . . . .	56
110	Entry Description (6083 <sub>h</sub> ) . . . . .	42	148	Object Description (607C <sub>h</sub> ) . . . . .	56
111	Object Description (6084 <sub>h</sub> ) . . . . .	42	149	Entry Description (607C <sub>h</sub> ) . . . . .	56
112	Entry Description (6084 <sub>h</sub> ) . . . . .	42	150	Object Description (6098 <sub>h</sub> ) . . . . .	56
113	Object Description (6085 <sub>h</sub> ) . . . . .	43	151	Entry Description (6098 <sub>h</sub> ) . . . . .	57
114	Entry Description (6085 <sub>h</sub> ) . . . . .	43	152	Object Description (6099 <sub>h</sub> ) . . . . .	57
115	Structure of the Control Word in pv Mode . . . . .	45	153	Entry Description (6099 <sub>h</sub> ) . . . . .	57
116	Command Coding . . . . .	46	154	Object Description (609A <sub>h</sub> ) . . . . .	57
117	Object Description (6040 <sub>h</sub> in pv Mode)	46	155	Entry Description (609A <sub>h</sub> ) . . . . .	58
118	Entry Description (6040 <sub>h</sub> in pv Mode)	46	156	Object Description (2100 <sub>h</sub> ) . . . . .	58
119	Structure of the Status Word in pv Mode	46	157	Entry Description (2100 <sub>h</sub> ) . . . . .	58
120	Trinamic Specific Bits . . . . .	47	158	Structure of the Control Word in cst Mode . . . . .	60
121	Operation Mode specific Bits in pv Mode	47	159	Command Coding . . . . .	60
122	State Coding . . . . .	47	160	Object Description (6040 <sub>h</sub> in cst Mode)	61
123	Object Description (6041 <sub>h</sub> in pv Mode)	47	161	Entry Description (6040 <sub>h</sub> in cst Mode)	61
124	Entry Description (6041 <sub>h</sub> in pv Mode)	48	162	Structure of the Status Word in cst Mode	61
125	Object Description (606B <sub>h</sub> ) . . . . .	48	163	Trinamic Specific Bits . . . . .	61
126	Entry Description (606B <sub>h</sub> ) . . . . .	48	164	Operation Mode specific Bits in cst Mode	62
127	Object Description (606C <sub>h</sub> ) . . . . .	48	165	State Coding . . . . .	62
128	Entry Description (606C <sub>h</sub> ) . . . . .	48	166	Object Description (6041 <sub>h</sub> in cst Mode)	62
129	Object Description (6083 <sub>h</sub> ) . . . . .	49	167	Entry Description (6041 <sub>h</sub> in cst Mode)	62
130	Entry Description (6083 <sub>h</sub> ) . . . . .	49	168	Object Description (6071 <sub>h</sub> ) . . . . .	63
131	Object Description (60FF <sub>h</sub> ) . . . . .	49	169	Entry Description (6071 <sub>h</sub> ) . . . . .	63
132	Entry Description (60FF <sub>h</sub> ) . . . . .	49	170	Object Description (6072 <sub>h</sub> ) . . . . .	63
133	Supported CANopen Homing Methods	52	171	Entry Description (6072 <sub>h</sub> ) . . . . .	63
134	Homing Methods 17 – 21 . . . . .	52	172	Object Description (6074 <sub>h</sub> ) . . . . .	63
135	Structure of the Control Word in hm Mode . . . . .	53	173	Entry Description (6074 <sub>h</sub> ) . . . . .	64
136	Operation Mode specific Bits in hm Mode . . . . .	53	174	Object Description (6077 <sub>h</sub> ) . . . . .	64
137	Command Coding . . . . .	53	175	Entry Description (6077 <sub>h</sub> ) . . . . .	64
138	Object Description (6040 <sub>h</sub> in hm Mode)	54	176	Object Description (60B2 <sub>h</sub> ) . . . . .	64
139	Entry Description (6040 <sub>h</sub> in hm Mode)	54	177	Entry Description (60B2 <sub>h</sub> ) . . . . .	64
140	Structure of the Status Word in hm Mode	54	178	Emergency Messages (EMCY) . . . . .	67
141	Trinamic Specific Bits . . . . .	54	179	Firmware Revision . . . . .	73
			180	Document Revision . . . . .	73



## 13 Supplemental Directives

### 13.1 Producer Information

### 13.2 Copyright

TRINAMIC owns the content of this user manual in its entirety, including but not limited to pictures, logos, trademarks, and resources. © Copyright 2017 TRINAMIC. All rights reserved. Electronically published by TRINAMIC, Germany.

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

### 13.3 Trademark Designations and Symbols

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

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

### 13.4 Target User

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

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

### 13.5 Disclaimer: Life Support Systems

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

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

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

### 13.6 Disclaimer: Intended Use

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



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

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

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

### 13.7 Collateral Documents & Tools

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





## 14 Revision History

### 14.1 Firmware Revision

Version	Date	Author	Description
4.00	02.10.2017	ED	First release.

*Table 179: Firmware Revision*

### 14.2 Document Revision

Version	Date	Author	Description
1.00	25.10.2017	ED	First release.

*Table 180: Document Revision*

