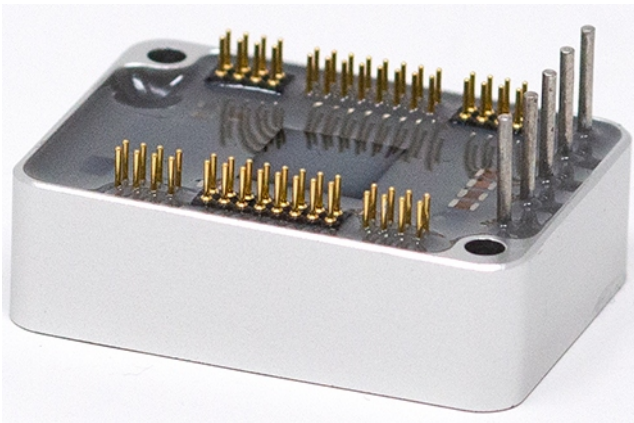


# TMCM-1617 CANopen® Firmware Manual

Firmware Version V1.08 | Document Revision V1.03 • 2021-12-17

The TMCM-1617 is a low-weight miniaturized single axis servo drive for 3-phase BLDC motors with up to 18A RMS motor current and +24V supply. With CAN, RS485, and EtherCAT® it offers various communication options. TMCM-1617 supports incremental encoders and digital hall sensors as position feedback. Customization and different housing options are possible.



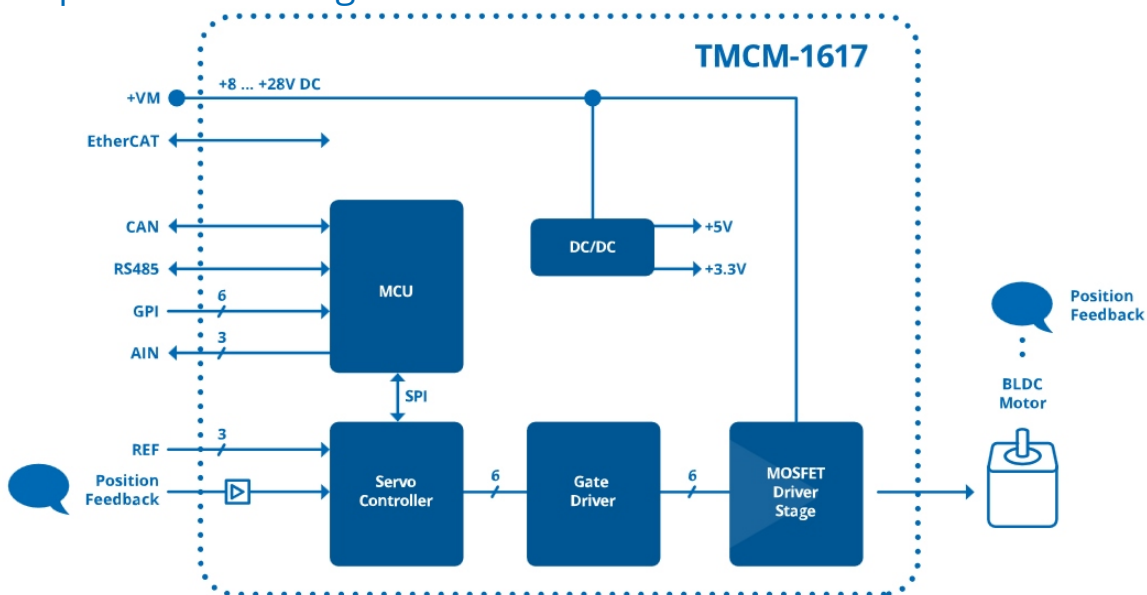
## Features

- Servo drive for BLDC motors
- +8...28V DC supply voltage
- Up to 18A RMS max. motor current
- RS485, CAN & EtherCAT® interfaces
- Incremental encoder feedback
- Digital HALL sensor feedback
- Reference switch inputs
- Cooling via aluminum housing
- L/W/H: 36.8mm x 26.8mm x 11.1mm
- Weight: ca. 24g
- Customization options

## Applications

- Robotics
- Laboratory Automation
- Manufacturing
- Factory Automation
- Servo Drives
- Low Inductance Motors

## Simplified Block Diagram



# Contents

<b>1</b>	<b>Preface</b>	<b>6</b>
1.1	General Features of this CANopen Implementation	6
1.2	Abbreviations used in this Manual	7
1.3	Firmware Update	7
<b>2</b>	<b>Communication</b>	<b>8</b>
2.1	Reference Model	8
2.2	NMT State Machine	10
2.3	Device Model	11
2.4	Object Dictionary	12
<b>3</b>	<b>Communication Area</b>	<b>14</b>
3.1	Detailed Object Specifications	14
3.1.1	Object 1000 <sub>h</sub> : Device Type	14
3.1.2	Object 1001 <sub>h</sub> : Error Register	14
3.1.3	Object 1005 <sub>h</sub> : COB-ID SYNC Message	15
3.1.4	Object 1008 <sub>h</sub> : Manufacturer Device Name	16
3.1.5	Object 1009 <sub>h</sub> : Manufacturer Hardware Version	16
3.1.6	Object 100A <sub>h</sub> : Manufacturer Software Version	16
3.1.7	Object 100C <sub>h</sub> : Guard Time	17
3.1.8	Object 100D <sub>h</sub> : Life Time Factor	17
3.1.9	Object 1010 <sub>h</sub> : Store Parameters	17
3.1.10	Object 1011 <sub>h</sub> : Restore Parameters	19
3.1.11	Object 1014 <sub>h</sub> : COB-ID Emergency Object	20
3.1.12	Object 1015 <sub>h</sub> : Inhibit Time EMCY	20
3.1.13	Object 1016 <sub>h</sub> : Consumer Heartbeat Time	21
3.1.14	Object 1017 <sub>h</sub> : Producer Heartbeat Time	22
3.1.15	Object 1018 <sub>h</sub> : Identity Object	22
3.1.16	Object 1029 <sub>h</sub> : Error Behaviour	23
3.1.17	Objects 1400 <sub>h</sub> – 1403 <sub>h</sub> : Receive PDO Communication Parameter	23
3.1.18	Objects 1600 <sub>h</sub> – 1603 <sub>h</sub> : Receive PDO Mapping Parameter	24
3.1.19	Objects 1800 <sub>h</sub> – 1803 <sub>h</sub> : Transmit PDO Communication Parameter	25
3.1.20	Objects 1A00 <sub>h</sub> – 1A03 <sub>h</sub> : Transmit PDO Mapping Parameter	26
<b>4</b>	<b>Manufacturer specific Area</b>	<b>28</b>
4.1	Detailed Object Specifications	28
4.1.1	Object 2000 <sub>h</sub> : Device Info	28
4.1.2	Object 2003 <sub>h</sub> : Maximum Current	28
4.1.3	Object 2004 <sub>h</sub> : Open Loop Current	29
4.1.4	Object 2005 <sub>h</sub> : Switch Parameters	29
4.1.5	Object 2006 <sub>h</sub> : Brake Chopper	30
4.1.6	Object 2041 <sub>h</sub> : Torque Mode Settings	31
4.1.7	Object 2042 <sub>h</sub> : Velocity Mode Settings	31
4.1.8	Object 2043 <sub>h</sub> : Position Mode Settings	32
4.1.9	Object 2050 <sub>h</sub> : Motor Type	33
4.1.10	Object 2055 <sub>h</sub> : Commutation Mode	33
4.1.11	Object 2056 <sub>h</sub> : Motor Pole Pairs	34
4.1.12	Object 2057 <sub>h</sub> : Motor Shaft Direction	35
4.1.13	Object 2058 <sub>h</sub> : Position Scaler	35
4.1.14	Object 2060 <sub>h</sub> : ADC Configuration	35
4.1.15	Object 2070 <sub>h</sub> : Hall Sensor Settings	36
4.1.16	Object 2080 <sub>h</sub> : ABN Encoder Settings	37

4.1.17	Object 2101 <sub>h</sub> : Motor Status Flags	38
4.1.18	Object 2102 <sub>h</sub> : Open Loop Commutation Angle	39
4.1.19	Object 2103 <sub>h</sub> : Encoder Commutation Angle	39
4.1.20	Object 2104 <sub>h</sub> : Hall Commutation Angle	40
4.1.21	Object 2140 <sub>h</sub> : Home Offset Display	40
4.1.22	Object 2702 <sub>h</sub> : Device Digital Inputs	41
4.1.23	Object 2703 <sub>h</sub> : Device Digital Outputs	41
4.1.24	Object 2704 <sub>h</sub> : CAN Bit Rate	42
4.1.25	Object 2705 <sub>h</sub> : Node-ID	42
4.1.26	Object 2707 <sub>h</sub> : CAN Bit Rate Load	43
4.1.27	Object 2708 <sub>h</sub> : Node-ID Load	43
4.1.28	Object 270E <sub>h</sub> : Device Analog Inputs	44
<b>5</b>	<b>Profile specific Area</b>	<b>46</b>
5.1	Detailed Object Specifications	46
5.1.1	Object 605A <sub>h</sub> : Quick Stop Option Code	46
5.1.2	Object 605B <sub>h</sub> : Shutdown Option Code	47
5.1.3	Object 605C <sub>h</sub> : Disable Operation Option Code	47
5.1.4	Object 605D <sub>h</sub> : Halt Option Code	48
5.1.5	Object 605E <sub>h</sub> : Fault Reaction Option Code	48
5.1.6	Object 6060 <sub>h</sub> : Modes of Operation	49
5.1.7	Object 6061 <sub>h</sub> : Modes of Operation Display	50
5.1.8	Object 60FD <sub>h</sub> : Digital Inputs	51
5.1.9	Object 6502 <sub>h</sub> : Supported Drive Modes	51
<b>6</b>	<b>Profile Position Mode</b>	<b>52</b>
6.1	Detailed Object Specifications	52
6.1.1	Object 6040 <sub>h</sub> : Control Word	53
6.1.2	Object 6041 <sub>h</sub> : Status Word	55
6.1.3	Object 6062 <sub>h</sub> : Position Demand Value	56
6.1.4	Object 6063 <sub>h</sub> : Position Actual Internal Value	57
6.1.5	Object 6064 <sub>h</sub> : Position Actual Value	57
6.1.6	Object 6067 <sub>h</sub> : Position Window	57
6.1.7	Object 6068 <sub>h</sub> : Position Window Time	58
6.1.8	Object 606C <sub>h</sub> : Velocity Actual Value	58
6.1.9	Object 607A <sub>h</sub> : Target Position	59
6.1.10	Object 607D <sub>h</sub> : Software Position Limit	59
6.1.11	Object 6081 <sub>h</sub> : Profile Velocity	60
6.1.12	Object 6083 <sub>h</sub> : Profile Acceleration	60
6.1.13	Object 6084 <sub>h</sub> : Profile Deceleration	61
6.1.14	Object 6085 <sub>h</sub> : Quick Stop Deceleration	61
6.2	How to move a Motor in pp Mode	62
<b>7</b>	<b>Profile Velocity Mode</b>	<b>63</b>
7.0.1	Object 6040 <sub>h</sub> : Control Word	63
7.0.2	Object 6041 <sub>h</sub> : Status Word	64
7.0.3	Object 6062 <sub>h</sub> : Position Demand Value	66
7.0.4	Object 6063 <sub>h</sub> : Position Actual Internal Value	66
7.0.5	Object 6064 <sub>h</sub> : Position Actual Value	67
7.0.6	Object 606C <sub>h</sub> : Velocity Actual Value	67
7.0.7	Object 607D <sub>h</sub> : Software Position Limit	67
7.0.8	Object 6083 <sub>h</sub> : Profile Acceleration	68
7.0.9	Object 6085 <sub>h</sub> : Quick Stop Deceleration	68
7.0.10	Object 60FF <sub>h</sub> : Target Velocity	69

7.1	How to move a Motor in pv Mode	69
<b>8</b>	<b>Homing Mode</b>	<b>70</b>
8.1	Homing Methods	71
8.1.1	Homing Method 17: Homing on negative Limit Switch	71
8.1.2	Homing Method 18: Homing on positive Limit Switch	71
8.1.3	Homing Method 19: Homing on positive Home Switch	72
8.1.4	Homing Method 21: Homing on negative Home Switch	72
8.1.5	Homing Method 35: Current Position as Home Position	73
8.2	Detailed Object Specifications	74
8.2.1	Object 6040 <sub>h</sub> : Control Word	74
8.2.2	Object 6041 <sub>h</sub> : Status Word	75
8.2.3	Object 606C <sub>h</sub> : Velocity Actual Value	76
8.2.4	Object 607C <sub>h</sub> : Home Offset	77
8.2.5	Object 6098 <sub>h</sub> : Homing Method	78
8.2.6	Object 6099 <sub>h</sub> : Homing Speeds	78
8.2.7	Object 609A <sub>h</sub> : Homing Acceleration	78
8.3	How to start a Homing in hm Mode	79
<b>9</b>	<b>Cyclic synchronous Position Mode</b>	<b>80</b>
9.1	Detailed Object Specifications	80
9.1.1	Object 6040 <sub>h</sub> : Control Word	80
9.1.2	Object 6041 <sub>h</sub> : Status Word	81
9.1.3	Object 6062 <sub>h</sub> : Position Demand Value	83
9.1.4	Object 6063 <sub>h</sub> : Position Actual Internal Value	83
9.1.5	Object 6064 <sub>h</sub> : Position Actual Value	84
9.1.6	Object 606C <sub>h</sub> : Velocity Actual Value	84
9.1.7	Object 607A <sub>h</sub> : Target Position	84
9.1.8	Object 607D <sub>h</sub> : Software Position Limit	85
9.1.9	Object 60B0 <sub>h</sub> : Position Offset	85
9.1.10	Object 60C2 <sub>h</sub> : Interpolation Time Period	86
<b>10</b>	<b>Cyclic synchronous Velocity Mode</b>	<b>87</b>
10.1	Detailed Object Specifications	87
10.1.1	Object 6040 <sub>h</sub> : Control Word	87
10.1.2	Object 6041 <sub>h</sub> : Status Word	88
10.1.3	Object 606C <sub>h</sub> : Velocity Actual Value	90
10.1.4	Object 60FF <sub>h</sub> : Target Velocity	90
10.1.5	Object 607D <sub>h</sub> : Software Position Limit	91
10.1.6	Object 60B1 <sub>h</sub> : Velocity Offset	91
10.1.7	Object 60C2 <sub>h</sub> : Interpolation Time Period	92
<b>11</b>	<b>Cyclic synchronous Torque Mode</b>	<b>93</b>
11.1	Detailed Object Specifications	93
11.1.1	Object 6040 <sub>h</sub> : Control Word	93
11.1.2	Object 6041 <sub>h</sub> : Status Word	94
11.1.3	Object 6062 <sub>h</sub> : Position Demand Value	95
11.1.4	Object 6063 <sub>h</sub> : Position Actual Internal Value	96
11.1.5	Object 6064 <sub>h</sub> : Position Actual Value	96
11.1.6	Object 6071 <sub>h</sub> : Target Torque	97
11.1.7	Object 6077 <sub>h</sub> : Torque actual Value	97
11.1.8	Object 607D <sub>h</sub> : Software Position Limit	97
11.1.9	Object 60B2 <sub>h</sub> : Torque Offset	98
11.1.10	Object 60C2 <sub>h</sub> : Interpolation Time Period	98

<b>12 Emergency Messages (EMCY)</b>	<b>100</b>
<b>13 SDO Abort Codes</b>	<b>102</b>
<b>14 Firmware Update</b>	<b>104</b>
<b>15 Figures Index</b>	<b>105</b>
<b>16 Tables Index</b>	<b>106</b>
<b>17 Supplemental Directives</b>	<b>109</b>
17.1 Producer Information . . . . .	109
17.2 Copyright . . . . .	109
17.3 Trademark Designations and Symbols . . . . .	109
17.4 Target User . . . . .	109
17.5 Disclaimer: Life Support Systems . . . . .	109
17.6 Disclaimer: Intended Use . . . . .	109
17.7 Collateral Documents & Tools . . . . .	110
<b>18 Revision History</b>	<b>111</b>
18.1 Firmware Revision . . . . .	111
18.2 Document Revision . . . . .	111

# 1 Preface

This document specifies objects and modes of operation of the Trinamic TMCM-1617 stepper motor control module with CANopen firmware. The CANopen firmware is designed to fulfill the CANopen DS402 and DS301 standards. This manual assumes that the reader is already familiar with the basics of the CANopen protocol, defined by the DS301 and DS402 standards of the CAN-CiA. If necessary it is always possible to turn the module into a TMCL module by loading the TMCM-1617 TMCL firmware again through the CAN interface or the RS485 interface, with the help of the firmware update function of the TMCL-IDE 3.0.

## 1.1 General Features of this CANopen Implementation

### Main Characteristics

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

### SDO Communication

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

### PDO Communication

- Producer
- Consumer
- RPDOs
  - Axis 0: 1, 2, 3, 4
  - Transmission modes: asynchronous.
  - Dynamic mapping with max. 3 mapping entries.
  - Default mappings: according to CiA-402 for first three PDOs of each axis, manufacturer specific for other PDOs of each axis.
- TPDOs
  - Axis 0: 1, 2, 3, 4
  - Transmission modes: asynchronous, asynchronous with event timer, synchronous.
  - Dynamic mapping with max. 3 mapping entries.
  - Default mappings: according to CiA-402 for first three PDOs of each axis, manufacturer specific for other PDOs of each axis.

### Further Characteristics

- SYNC: consumer (TPDOs 3 are synchronous PDOs)
- Emergency: producer
- RTR: supported only for node guarding/life guarding
- Heartbeat: consumer and producer

## 1.2 Abbreviations used in this Manual

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

*Table 1: Abbreviations used in this Manual*

## 1.3 Firmware Update

The software running on the microprocessor consists of two parts, a boot loader and the CANopen firmware itself. Whereas the boot loader is installed during production and testing at TRINAMIC and remains untouched throughout the whole lifetime, the CANopen firmware can easily be updated by the user. The new firmware can be loaded into the module via the firmware update function of the TMCL-IDE, using the CAN interface or the RS485 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 CAN Network 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.

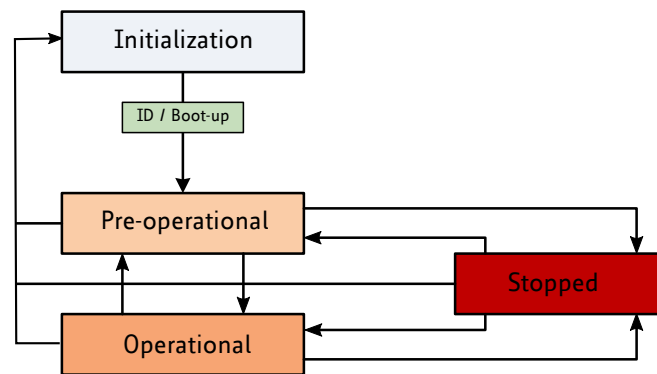


Figure 1: NMT State Machine

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

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

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

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

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

The NMT state machine (or DS301 state machine) is not to be confused with the DS402 state machine. There is only one NMT state machine for the entire device, but for each motor there is a DS402 state machine which controls the motor. There are no links between these state machines, with one exception: When the NMT state machine is being switched to the stopped state, all DS402 state machines that are in OPERATION\_ENABLED state will be switch to FAULT state.

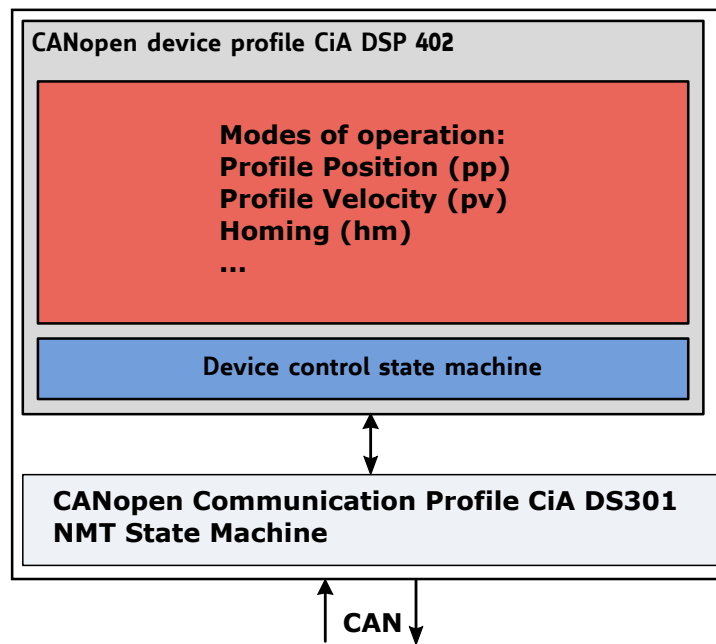


Figure 2: Communication Architecture

## 2.3 Device Model

A CANopen 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.

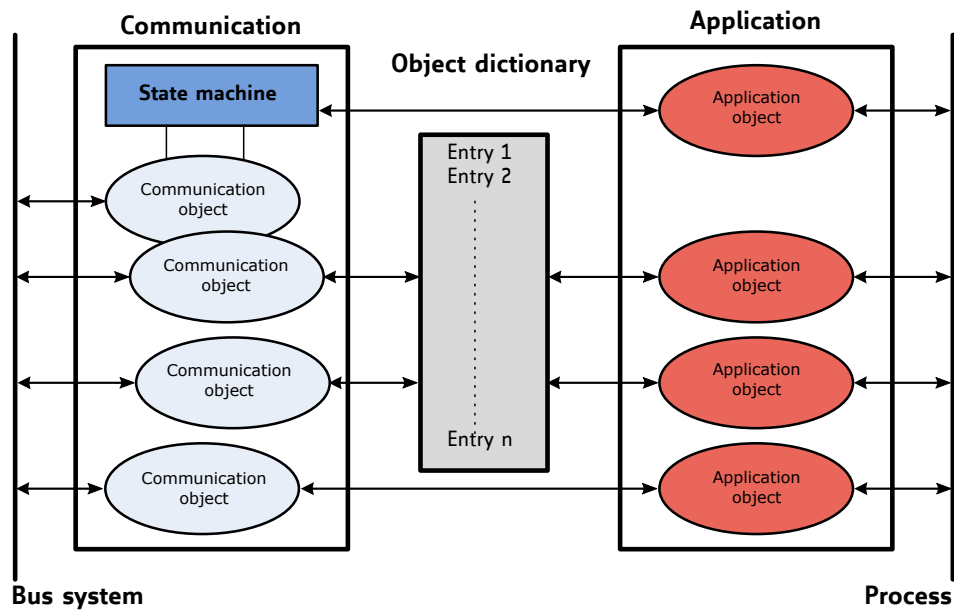


Figure 3: Device Model

## 2.4 Object Dictionary

The most important part of a device profile is the object dictionary description. The object dictionary is essentially a grouping of objects accessible via the network in an ordered pre-defined fashion. Each object within the dictionary is addressed using a 16-bit index. The overall layout of the standard object dictionary is 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 TMC-1617 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 CANopen device according to the DS301 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	FFFC0192 <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 1005<sub>h</sub>: COB-ID SYNC Message

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

Value Definition		
Bit	Name	Definition
30	Generate	0: Device does not generate SYNC message 1: Device generates SYNC message
29	Frame	Not supported, always set to 0.
28...11	29 bit ID	Not supported, always set to 0.
10...0	11 bit ID	11 bit COB-ID.

Table 10: Value Definition (1005<sub>h</sub>)

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

Table 11: Object Description (1005<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	UNSIGNED32	80 <sub>h</sub>

Table 12: Entry Description (1005<sub>h</sub>)

### 3.1.4 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 13: Object Description (1008<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	no	—	TMCM-1617

Table 14: Entry Description (1008<sub>h</sub>)

### 3.1.5 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 15: 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. 1.00

Table 16: Entry Description (1009<sub>h</sub>)

### 3.1.6 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 17: 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. 1.00.

Table 18: Entry Description (100A<sub>h</sub>)

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

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

Object Description			
Index	Name	Object Type	Data Type
100C <sub>h</sub>	Guard Time	Variable	UNSIGNED16

Table 19: Object Description (100C<sub>h</sub>)

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

Table 20: Entry Description (100C<sub>h</sub>)

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

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

Object Description			
Index	Name	Object Type	Data Type
100D <sub>h</sub>	Life Time Factor	Variable	UNSIGNED8

Table 21: Object Description (100D<sub>h</sub>)

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

Table 22: Entry Description (100D<sub>h</sub>)

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

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

The TMCM-1617 module supports saving of the following parameter groups:

- Sub-index 1: save all parameters.
- Sub-index 2: save communication parameters.
- Sub-index 3: save device profile parameters (not used).
- Sub-index 4: save motor 0 parameters.
- Sub-index 5: save device parameters (other non axis-related parameters).

---

**Note** In order to avoid storage of parameters by mistake, storage is only executed when a specific signature is written to the appropriate sub-Index. This signature is "save" (65766173<sub>h</sub>, see also table 23).

---

Save Signature			
e	v	a	s
65 <sub>h</sub>	76 <sub>h</sub>	61 <sub>h</sub>	73 <sub>h</sub>

Table 23: Save Signature

On reception of the correct signature in the appropriate sub-index the device stores the parameter and then confirms the SDO transmission (initiate download response). If the storing failed, the device responds with an abort SDO transfer (abort code: 06060000<sub>h</sub>). If a wrong signature is written, the device refuses to store and responds with abort SDO transfer (abort code: 0800002x<sub>h</sub>).

On read access, each sub-index provides information if it is possible to store the parameter group. It reads 1 if yes and 0 if no.

Object Description			
Index	Name	Object Type	Data Type
1010 <sub>h</sub>	Store Parameters	Array	UNSIGNED32

Table 24: Object Description (1010<sub>h</sub>)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
0	Highest supported sub-index	ro	no	UNSIGNED8	5
1	Save all parameters	rw	no	UNSIGNED32	1
2	Save communication parameters	rw	no	UNSIGNED32	1
3	Save device profile parameters	rw	no	UNSIGNED32	0
4	Save motor 0 parameters	rw	no	UNSIGNED32	1
5	Save device parameters	rw	no	UNSIGNED32	1

Table 25: Entry Description (1010<sub>h</sub>)

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

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

The TMC-1617 module supports restoring of the following parameter groups:

- Sub-index 1: restore all parameters.
- Sub-index 2: restore communication parameters.
- Sub-index 3: restore device profile parameters (not used).
- Sub-index 4: restore motor 0 parameters.
- Sub-index 5: restore device parameters (other non axis-related parameters).

#### Note

In order to avoid restoring the parameters by mistake, restoring is only executed when a specific signature is written to the appropriate sub-Index. This signature is "load" (64616F6C<sub>h</sub>, see also table 26).

Load Signature			
d	a	o	l
64 <sub>h</sub>	61 <sub>h</sub>	6F <sub>h</sub>	6C <sub>h</sub>

Table 26: Load Signature

On reception of the correct signature in the appropriate sub-index the device restores the parameter and then confirms the SDO transmission (initiate download response). If the restoring failed, the device responds with an abort SDO transfer (abort code: 06060000<sub>h</sub>). If a wrong signature is written, the device refuses to restore and responds with abort SDO transfer (abort code: 0800002x<sub>h</sub>).

On read access, each sub-index provides information if it is possible to restore the parameter group. It reads 1 if yes and 0 if no.

After the default values have been restored they will become active after the next rest or power cycle of the TMCM-1617.

Object Description			
Index	Name	Object Type	Data Type
1011 <sub>h</sub>	Restore parameters	Array	UNSIGNED32

Table 27: Object Description (1011<sub>h</sub>)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
0	Highest supported sub-index	ro	no	UNSIGNED8	5
1	Restore all parameters	rw	no	UNSIGNED32	1
2	Restore communication parameters	rw	no	UNSIGNED32	1
3	Restore device profile parameters	rw	no	UNSIGNED32	0
4	Restore motor 0 parameters	rw	no	UNSIGNED32	1
5	Restore device parameters	rw	no	UNSIGNED32	1

Table 28: Entry Description (1011<sub>h</sub>)

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

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

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

Table 29: Object Description (1014<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	UNSIGNED32	80 <sub>h</sub> + Node ID

Table 30: Entry Description (1014<sub>h</sub>)

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

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

Object Description			
Index	Name	Object Type	Data Type
1015 <sub>h</sub>	COB-ID emergency object	Variable	UNSIGNED16

Table 31: Object Description (1015<sub>h</sub>)

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

Table 32: Entry Description (1015<sub>h</sub>)

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

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

Value Definition		
Bits	Name	Definition
31...24	Reserved	—
23...16	Node ID	Heartbeat Producer Node ID
15...0	Heartbeat time	Time in 1ms

Table 33: Value Definition (1016<sub>h</sub>)

Object Description			
Index	Name	Object Type	Data Type
1016 <sub>h</sub>	Consumer heartbeat time	Array	UNSIGNED32

Table 34: Object Description (1016<sub>h</sub>)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
0	Number of entries	ro	no	UNSIGNED8	1
1	Consumer heartbeat time 1	rw	no	UNSIGNED32	0

Table 35: Entry Description (1016<sub>h</sub>)

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

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

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

Table 36: Object Description (1017<sub>h</sub>)

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

Table 37: Entry Description (1017<sub>h</sub>)

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

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

- The vendor ID (sub-index 01<sub>h</sub>) contains a unique value allocated to each manufacturer. The vendor ID of Trinamic is 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 38: Object Description (1018<sub>h</sub>)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
0	Number of entries	ro	no	0...3	3
1	Vendor ID	ro	no	UNSIGNED32	0286 <sub>h</sub>
2	Product code	ro	no	UNSIGNED32	1617
3	Revision number	ro	no	UNSIGNED32	e.g. 20003 <sub>h</sub> for version 2.3

Table 39: Entry Description (1018<sub>h</sub>)

### 3.1.16 Object 1029<sub>h</sub>: Error Behaviour

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

- Communication error
- Application error

Object Description			
Index	Name	Object Type	Data Type
1029 <sub>h</sub>	Error behaviour	Array	UNSIGNED8

Table 40: Object Description (1029<sub>h</sub>)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
0	Number of error classes	ro	no	—	2
1	Communication error	rw	no	UNSIGNED8	0 (enter stopped state)
2	Application error	rw	no	UNSIGNED8	1 (remain in current state)

Table 41: Entry Description (1029<sub>h</sub>)

### 3.1.17 Objects 1400<sub>h</sub> – 1403<sub>h</sub>: Receive PDO Communication Parameter

This object contains the communication parameters for the RPDOs which the device is able to receive. The sub-index 0 contains the number of valid entries within the communication record. Its value normally is 2, as this object consists of two other entries.

Sub-index 1 contains the COB-ID used by this PDO (in bits 10...0). Bit 30 (RTR bit) defines if this PDO uses RTRs. As RTRs are not supported for PDOs by this CANopen implementation, this bit must always be set in order to turn off RTR support for this PDO. Bit 31 defines if this PDO is active or not. If this bit is set, the PDO is inactive, and if this bit is clear, the PDO is active. Before making any changes to a PDO definition, set this bit to inactivate the PDO.

Sub-Index 2 contains the transmission type of the RPDO. This can be FF<sub>h</sub> or FE<sub>h</sub> for event-driven, or 1...240 for synchronous (1 means that the PDO will be processed with every SYNC message, and 4 for example means that the PDO will be processed with every 4<sup>th</sup> SYNC message). Other values are not supported.

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

Table 42: Object Description (1400<sub>h</sub>)

Entry Description				
Sub-index	Description	Access	Value Range	Default Value
0	Largest sub-index supported	ro	2	2
1	COB-ID used by PDO	rw	UNSIGNED32	Index 1400 <sub>h</sub> : 200 <sub>h</sub> + Node-ID Index 1401 <sub>h</sub> : 300 <sub>h</sub> + Node-ID Index 1402 <sub>h</sub> : 400 <sub>h</sub> + Node-ID Index 1403 <sub>h</sub> : 500 <sub>h</sub> + Node-ID
2	Transmission type	rw	UNSIGNED8	Index 1400 <sub>h</sub> : FF <sub>h</sub> Index 1401 <sub>h</sub> : FF <sub>h</sub> Index 1402 <sub>h</sub> : FF <sub>h</sub> Index 1403 <sub>h</sub> : FE <sub>h</sub>

Table 43: Entry Description (1400<sub>h</sub>)

### 3.1.18 Objects 1600<sub>h</sub> – 1603<sub>h</sub>: Receive PDO Mapping Parameter

These objects contain the mapping parameters for the RPDOs the device is able to receive. The sub-index 0 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 1 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> – 1603 <sub>h</sub>	Receive PDO mapping parameter	RECORD	PDO Mapping
1600 <sub>h</sub>	RPDO 1	RECORD	PDO Mapping
1601 <sub>h</sub>	RPDO 2	RECORD	PDO Mapping
1602 <sub>h</sub>	RPDO 3	RECORD	PDO Mapping
1603 <sub>h</sub>	RPDO 4	RECORD	PDO Mapping

Table 44: Object Description (1600<sub>h</sub>)



Entry Description				
Sub-index	Description	Access	Value Range	Default Value
0	Number of mapped application objects in PDO	rw	0...3	Index 1600 <sub>h</sub> : 1 Index 1601 <sub>h</sub> : 2 Index 1602 <sub>h</sub> : 2 Index 1603 <sub>h</sub> : 2
1	Mapping entry 1	rw	UNSIGNED32	Index 1600 <sub>h</sub> : 60400010 <sub>h</sub> Index 1601 <sub>h</sub> : 60400010 <sub>h</sub> Index 1602 <sub>h</sub> : 60400010 <sub>h</sub> Index 1603 <sub>h</sub> : 60400010 <sub>h</sub>
2	Mapping entry 2	rw	UNSIGNED32	Index 1600 <sub>h</sub> : 0 Index 1601 <sub>h</sub> : 60600008 <sub>h</sub> Index 1602 <sub>h</sub> : 607A0020 <sub>h</sub> Index 1603 <sub>h</sub> : 60FF0020 <sub>h</sub>
3	Mapping entry 3	rw	UNSIGNED32	Index 1600 <sub>h</sub> : 0 <sub>h</sub> Index 1601 <sub>h</sub> : 0 <sub>h</sub> Index 1602 <sub>h</sub> : 0 <sub>h</sub> Index 1603 <sub>h</sub> : 0 <sub>h</sub>

Table 45: Entry Description (1600<sub>h</sub>)

Before making changes to PDO definitions, first mark the PDO as inactive by setting bit 31 of its COB-ID (see section 3.1.17). Then, set its number of mapped PDO entries to zero (sub-index 0 of the appropriate PDO mapping object). Now, the mappings themselves can be changed. After that, set the number of map objects to the desired value, and finally activate the PDO by clearing bit 31 of its COB-ID.

### 3.1.19 Objects 1800<sub>h</sub> – 1803<sub>h</sub>: Transmit PDO Communication Parameter

This object contains the communication parameters for the TPDOs which the device is able to transmit. The sub-index 0 contains the number of valid entries within the communication record. Its value normally is 5, as this object consists of five other entries.

Sub-index 1 contains the COB-ID used by this PDO (in bits 10...0). Bit 30 (RTR bit) defines if this PDO uses RTRs. As RTRs are not supported for PDOs by this CANopen implementation, this bit must always be set in order to turn off RTR support for this PDO. Bit 31 defines if this PDO is active or not. If this bit is set, the PDO is inactive, and if this bit is clear, the PDO is active. Before making any changes to a PDO definition, set this bit to inactivate the PDO.

Sub-index 2 contains the transmission type of the RPDO. This can be FF<sub>h</sub> or FE<sub>h</sub> for event-driven or 1...240 for synchronous (1 means that the PDO will be sent with every SYNC message, and 4 for example means that the PDO will be sent with every 4<sup>th</sup> SYNC message). Other values are not supported.

Sub-index 3 contains the inhibit time, given in units of 0.1ms. After a TPDO has been sent, it will not be sent again before the inhibit time has elapsed.

Sub-index 4 is not used.

Sub-index 5 contains the event timer value in milliseconds. When this is set to a value greater than 0 the TPDO will be sent repeatedly each time the event timer has elapsed. It will also be sent when the value has changed before the event timer has elapsed, but not before the inhibit time has elapsed.

Object Description			
Index	Name	Object Type	Data Type
1800 <sub>h</sub> – 1803 <sub>h</sub>	Transmit PDO communication parameter	RECORD	TPDO CommPar
1800 <sub>h</sub>	TPDO 1	RECORD	TPDO CommPar
1801 <sub>h</sub>	TPDO 2	RECORD	TPDO CommPar
1802 <sub>h</sub>	TPDO 3	RECORD	TPDO CommPar
1803 <sub>h</sub>	TPDO 4	RECORD	TPDO CommPar

Table 46: Object Description (1800<sub>h</sub>)

Entry Description				
Sub-index	Description	Access	Value Range	Default Value
0	Largest sub-index supported	ro	5	5
1	COB-ID	rw	UNSIGNED32	Index 1800 <sub>h</sub> : 180 <sub>h</sub> + Node-ID Index 1801 <sub>h</sub> : 280 <sub>h</sub> + Node-ID Index 1802 <sub>h</sub> : 380 <sub>h</sub> + Node-ID Index 1803 <sub>h</sub> : 480 <sub>h</sub> + Node-ID
2	Transmission type	rw	UNSIGNED8	Index 1800 <sub>h</sub> : FF <sub>h</sub> Index 1801 <sub>h</sub> : FF <sub>h</sub> Index 1802 <sub>h</sub> : 01 <sub>h</sub> Index 1803 <sub>h</sub> : 01 <sub>h</sub>
3	Inhibit time	rw	UNSIGNED16	0
4	Compatibility entry	ro	UNSIGNED8	0
5	Event timer	rw	UNSIGNED16	0

Table 47: Entry Description (1800<sub>h</sub>)

### 3.1.20 Objects 1A00<sub>h</sub> – 1A03<sub>h</sub>: Transmit PDO Mapping Parameter

These objects contain the mapping parameters for the TPDOs the device is able to transmit. The sub-index 0 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 1 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> – 1A03 <sub>h</sub>	Transmit PDO mapping parameter	RECORD	PDO Mapping
1A00 <sub>h</sub>	TPDO 1	RECORD	PDO Mapping
1A01 <sub>h</sub>	TPDO 2	RECORD	PDO Mapping
1A02 <sub>h</sub>	TPDO 3	RECORD	PDO Mapping
1A03 <sub>h</sub>	TPDO 4	RECORD	PDO Mapping

Table 48: Object Description (1A00<sub>h</sub>)

Entry Description				
Sub-index	Description	Access	Value Range	Default Value
0	Number of mapped application objects in PDO	rw	0...3	Index 1A00 <sub>h</sub> : 1 Index 1A01 <sub>h</sub> : 2 Index 1A02 <sub>h</sub> : 2 Index 1A03 <sub>h</sub> : 2
1	Mapping entry 1	rw	UNSIGNED32	Index 1A00 <sub>h</sub> : 60410010 <sub>h</sub> Index 1A01 <sub>h</sub> : 60410010 <sub>h</sub> Index 1A02 <sub>h</sub> : 60410010 <sub>h</sub> Index 1A03 <sub>h</sub> : 60410010 <sub>h</sub>
2	Mapping entry 2	rw	UNSIGNED32	Index 1A00 <sub>h</sub> : 0 Index 1A01 <sub>h</sub> : 60610008 <sub>h</sub> Index 1A02 <sub>h</sub> : 60640020 <sub>h</sub> Index 1A03 <sub>h</sub> : 606C0020 <sub>h</sub>
3	Mapping entry 3	rw	UNSIGNED32	Index 1A00 <sub>h</sub> : 0 <sub>h</sub> Index 1A01 <sub>h</sub> : 0 <sub>h</sub> Index 1A02 <sub>h</sub> : 0 <sub>h</sub> Index 1A03 <sub>h</sub> : 0 <sub>h</sub>

Table 49: Entry Description (1A00<sub>h</sub>)

Before making changes to PDO definitions, first mark the PDO as inactive by setting bit 31 of its COB-ID (see section 3.1.19). Then, set its number of mapped PDO entries to zero (sub-index 0 of the appropriate PDO mapping object). Now, the mappings themselves can be changed. After that, set the number of mapped objects to the desired value, and finally activate the PDO by clearing bit 31 of its COB-ID.

## 4 Manufacturer specific Area

The manufacturer segment contains manufacturer specific objects. These objects control the special features of the Trinamic Motion Control device TMCM-1617.

### 4.1 Detailed Object Specifications

#### 4.1.1 Object 2000<sub>h</sub>: Device Info

This object provides version information about the motor controller chip used on this module.

Object Description			
Index	Name	Object Type	Data Type
2000 <sub>h</sub>	Device Info	Variable	Record

Table 50: Object Description (2000<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	MC_Type	no	0	4294967295	0	—	ro
2	MC_Version	no	0	4294967295	0	—	ro
3	MC_Date	no	0	4294967295	0	—	ro
4	MC_Time	no	0	4294967295	0	—	ro
5	MC_Variant	no	0	4294967295	0	—	ro

Table 51: Entry Description (2000<sub>h</sub>)

#### 4.1.2 Object 2003<sub>h</sub>: Maximum Current

This object limits the maximum current that is used to drive the motor. The value is given in mA.

Object Description			
Index	Name	Object Type	Data Type
2003 <sub>h</sub>	Maximum Current	Variable	UNSIGNED32

Table 52: Object Description (2003<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0...18000	0

Table 53: Entry Description (2003<sub>h</sub>)

#### 4.1.3 Object 2004<sub>h</sub>: Open Loop Current

This object controls the motor current used in open loop mode. The value is given in mA.

Object Description			
Index	Name	Object Type	Data Type
2004 <sub>h</sub>	Open Loop Current	Variable	UNSIGNED32

Table 54: Object Description (2004<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0...18000	0

Table 55: Entry Description (2004<sub>h</sub>)

#### 4.1.4 Object 2005<sub>h</sub>: Switch Parameters

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

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

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

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

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

Table 56: Object Description (2005<sub>h</sub>)

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

Table 57: 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 58: Bit Definitions (2005<sub>h</sub>)

#### 4.1.5 Object 2006<sub>h</sub>: Brake Chopper

With this object the behaviour of the brake chopper output can be set up.

Object Description			
Index	Name	Object Type	Data Type
2006 <sub>h</sub>	Device Info	Variable	Record

Table 59: Object Description (2006<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Enable	no	0	1	0	—	RW
2	Voltage	no	60	300	300	0.1 V	RW
3	Hysteresis	no	0	50	5	0.1 V	RW

Table 60: Entry Description (2006<sub>h</sub>)

#### 4.1.6 Object 2041<sub>h</sub>: Torque Mode Settings

Object Description			
Index	Name	Object Type	Data Type
2041 <sub>h</sub>	Torque Mode Settings	Variable	Record

Table 61: Object Description (2041<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Torque_P	no	0	65535	300		RW
2	Torque_I	no	0	65535	300		RW
3	PI_Torque_Error	no	-2147483648	2147483647	0	[mA]	R
4	PI_Torque_Error_Sum	no	-2147483648	2147483647	0		R
5	PI_Flux_Error	no	-2147483648	2147483647	0	[mA]	R
6	PI_Flux_Error_Sum	no	-2147483648	2147483647	0		R
7	PHI_E	no	-32678	32767	0		R

Table 62: Entry Description (2041<sub>h</sub>)

Torque\_P P parameter for the torque PI controller.

Torque\_I I parameter for the torque PI controller.

#### 4.1.7 Object 2042<sub>h</sub>: Velocity Mode Settings

Object Description			
Index	Name	Object Type	Data Type
2042 <sub>h</sub>	Velocity Mode Settings	Variable	Record

Table 63: Object Description (2042<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Velocity_P	no	0	65535	300		RW
2	Velocity_I	no	0	65535	100		RW
3	PI_Velocity_Error	no	-2147483648	2147483647	0		R
4	PI_Velocity_Error_Sum	no	-2147483648	2147483647	0		R
5	Sensor_Selection	no	0	1	0		RW
6	Velocity_Filter	no	0	8	3		RW
7	Velocity_Unit	no	0	1	0		RW
8	Motor_Halted_Velocity	no	0	200000	10		RW

Table 64: Entry Description (2042<sub>h</sub>)

- Velocity\_P                    P parameter for the velocity PI controller.
- Velocity\_I                    I parameter for the velocity PI controller.
- Sensor\_Selection            Select a commutation mode that fits best to your motor’s sensors.
  - 0: Same as commutation
  - 1: ABN encoder
- Velocity\_Filter              Moving average filter.
- Velocity\_Unit                Select mechanical or electrical velocity unit.
  - 0: Mechanical rpm
  - 1: Electrical rpm
- Motor\_Halted\_Velocity     If the actual velocity is below this value the motor halted flag will beset.

#### 4.1.8 Object 2043<sub>h</sub>: Position Mode Settings

Object Description			
Index	Name	Object Type	Data Type
2043 <sub>h</sub>	Position Mode Settings	Variable	Record

Table 65: Object Description (2043<sub>h</sub>)



Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Position_P	no	0	65535	50		RW
2	PI_Position_Error	no	-2147483648	2147483647	0		R
3	Sensor_Selection	no	0	1	0		RW

Table 66: Entry Description (2043<sub>h</sub>)

Position\_P P parameter for the velocity PI controller.

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

- 0: Same as commutation
- 1: ABN encoder

#### 4.1.9 Object 2050<sub>h</sub>: Motor Type

With this object the motor type connected to the module can be set. The following settings are possible:

- Mode 0: no motor
- Mode 1: single phase DC motor
- Mode 3: three phase BLDC motor

Object Description			
Index	Name	Object Type	Data Type
2050 <sub>h</sub>	Limits	Variable	UNSIGNED8

Table 67: Object Description (2050<sub>h</sub>)

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

Table 68: 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.

Commutation Modes	
0	FOC — disabled
1	FOC — open loop
2	FOC — digital hall
3	FOC — ABN encoder

Table 69: Commutation Modes

Object Description			
Index	Name	Object Type	Data Type
2055 <sub>h</sub>	Commutation Mode	Variable	UNSIGNED8

Table 70: Object Description (2055<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Commutation Mode	no	0	1	3		RW

Table 71: Entry Description (2055<sub>h</sub>)

#### 4.1.11 Object 2056<sub>h</sub>: Motor Pole Pairs

Set this object to the number of pole pairs your motor is equipped with.

Object Description			
Index	Name	Object Type	Data Type
2056 <sub>h</sub>	Motor Pole Pairs	Variable	UNSIGNED8

Table 72: Object Description (2056<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Motor Pole Pairs	no	1	12	4		RW

Table 73: Entry Description (2056<sub>h</sub>)

#### 4.1.12 Object 2057<sub>h</sub>: Motor Shaft Direction

Using this object the motor shaft direction can be reversed. Set it to 0 (default value) for normal shaft direction or 1 for reversed shaft direction.

Object Description			
Index	Name	Object Type	Data Type
2057 <sub>h</sub>	Motor Shaft Direction	Variable	UNSIGNED8

Table 74: Object Description (2057<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Motor Shaft Direction	no	0	1	0		RW

Table 75: Entry Description (2057<sub>h</sub>)

#### 4.1.13 Object 2058<sub>h</sub>: Position Scaler

Using this object all position values can be scaled. It defines the number of steps per mechanical rotation. With its default value of 65536, a move of 65536 steps leads to one mechanical rotation.

Object Description			
Index	Name	Object Type	Data Type
2058 <sub>h</sub>	Position Scaler	Variable	SIGNED32

Table 76: Object Description (2058<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Position Scaler	no	-2147483648	2147483647	65536		RW

Table 77: Entry Description (2058<sub>h</sub>)

#### 4.1.14 Object 2060<sub>h</sub>: ADC Configuration

Using this object the ADC offsets for the coil current measurement can be configured. This is necessary for each new motor type.

Object Description			
Index	Name	Object Type	Data Type
2060 <sub>h</sub>	ADC Configuration	Variable	Record

Table 78: Object Description (2060<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	ADC_I0_Raw	no	0	65535	0		RO
2	ADC_I1_Raw	no	0	65535	0		RO
3	ADC_I0_Offset	no	0	65535	33500		RW
4	ADC_I1_Offset	no	0	65535	33500		RW
5	ADC_I0	no	-32768	32767	0		RO
6	ADC_I1	no	-32768	32767	0		RO
7	ADC_I2	no	-32768	32767	0		RO

Table 79: Entry Description (2060<sub>h</sub>)

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

This object sets various parameters of the hall sensors. If the motor is equipped with hall sensors then set the necessary parameters here.

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

Table 80: Object Description (2070<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Polarity	no	0	1	0		RW
2	Direction	no	0	1	0		RW
3	Interpolation	no	0	1	1		RW
4	PHI_E_Offset	no	-32768	32767	0		RW

Table 81: Entry Description (2070<sub>h</sub>)

- Polarity      Hall sensor polarity.
  - 0: Standard
  - 1: Inverted
  
- Direction    Hall sensor direction.
  - 0: Standard
  - 1: Inverted
  
- Interpolation   Hall sensor interpolation.
  - 0: On
  - 1: Off
  
- PHI\_E\_Offset   Offset for electrical angle hall\_phi\_e of hall sensor.

**4.1.16 Object 2080<sub>h</sub>: ABN Encoder Settings**

Using this object all necessary encoder parameters can be set. Check and set these parameters if your motor is equipped with an encoder. It is then also possible to choose between different encoder initialization modes.

Encoder Initialization Modes	
0	Estimate offset
1	Use offset
2	Use hall

*Table 82: Encoder Initialization Modes*

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

*Table 83: Object Description (2080<sub>h</sub>)*

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Direction	no	0	1	0		RW
2	Steps	no	0	16777215	4096		RW
3	Init_Mode	no	0	2	0		RW

Table 84: Entry Description (2080<sub>h</sub>)

#### 4.1.17 Object 2101<sub>h</sub>: Motor Status Flags

This object provides motor status and error flags. This can be a combination of the bits described in table 85.

Motor Status Flags		
Bit	Name	Meaning
0	Overcurrent	Too high current detected.
1	Undervoltage	Supply voltage too low.
2	Overvoltage	Supply voltage too high.
3	Overtemperature	Maximum driver temperature exceeded.
4	Motor halted	Motor stopped.
5	Hall error	Hall sensor error.
6	Driver error	Motor driver error.
7	Init error	Motor initialization error.
8	Stop mode	Motor in stop mode.
9	Velocity mode	Motor operating in velocity mode.
10	Position mode	Motor operating in position mode.
11	Torque mode	Motor operating in torque mode.
12	Emergency stop	Emergency stop active.
14	Position end	Target position reached.
15	Module initialized	Module initialization complete.
17	IIT exceeded	IIT limit exceeded.
18	Brake active	Brake output active.

Table 85: Motor Status Flags (2101<sub>h</sub>)

Object Description			
Index	Name	Object Type	Data Type
2101 <sub>h</sub>	Device State	Variable	UNSIGNED32

Table 86: Object Description (2101<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	no	0...3FFFF <sub>h</sub>	0

Table 87: Entry Description (2101<sub>h</sub>)

#### 4.1.18 Object 2102<sub>h</sub>: Open Loop Commutation Angle

This object shows the open loop commutation angle. It is mainly used by the Trinamic motor tuning tools.

Object Description			
Index	Name	Object Type	Data Type
2102 <sub>h</sub>	Open Loop Commutation Angle	Variable	SIGNED16

Table 88: Object Description (2102<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Open Loop Commutation Angle	no	-32768	32767	0		RO

Table 89: Entry Description (2102<sub>h</sub>)

#### 4.1.19 Object 2103<sub>h</sub>: Encoder Commutation Angle

This object shows the encoder commutation angle. It is mainly used by the Trinamic motor tuning tools.

Object Description			
Index	Name	Object Type	Data Type
2103 <sub>h</sub>	Encoder Commutation Angle	Variable	SIGNED16

Table 90: Object Description (2103<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Encoder Commutation Angle	no	-32768	32767	0		RO

Table 91: Entry Description (2103<sub>h</sub>)

#### 4.1.20 Object 2104<sub>h</sub>: Hall Commutation Angle

This object shows the hall sensor commutation angle. It is mainly used by the Trinamic motor tuning tools.

Object Description			
Index	Name	Object Type	Data Type
2104 <sub>h</sub>	Hall Commutation Angle	Variable	SIGNED16

Table 92: Object Description (2104<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Hall Commutation Angle	no	-32768	32767	0		RO

Table 93: Entry Description (2104<sub>h</sub>)

#### 4.1.21 Object 2140<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
2140 <sub>h</sub>	Home Offset Display	Variable	SIGNED32

Table 94: Object Description (2140<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 95: Entry Description (2140<sub>h</sub>)



#### 4.1.22 Object 2702<sub>h</sub>: Device Digital Inputs

Bit Definitions	
Bit	Description
0	REF_R
1	REF_L
2	REF_H
3	GPIO0
4	GPIO1

Table 96: Bit Definitions (2702<sub>h</sub>)

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

Table 97: Object Description (2702<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Digital Inputs	no	0	3	0		R

Table 98: Entry Description (2702<sub>h</sub>)

#### 4.1.23 Object 2703<sub>h</sub>: Device Digital Outputs

With this object the digital outputs (general purpose outputs) can be set. Bits 0...3 of sub index 1 switch the outputs of the module. Bits 0...3 of sub index 2 determine which outputs can be switched. The number of available digital outputs depends on the module type.

Bit Definitions	
Bit	Description
0	GPO2
1	GPO3
2	GPO4
3	GPO5

Table 99: Bit Definitions (2703<sub>h</sub>)

Object Description			
Index	Name	Object Type	Data Type
2703 <sub>h</sub>	Device Digital Outputs	Variable	ARRAY

Table 100: Object Description (2703<sub>h</sub>)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
1	Physical Outputs	rw	yes	UNSIGNED32	0
2	Output Mask	rw	yes	UNSIGNED32	0

Table 101: Entry Description (2703<sub>h</sub>)

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

With this object it is possible to change the CAN bit rate.

To do this, first write the new value to this object. Then, store the new setting by writing the save signature to object [Object 1010<sub>h</sub>: Store Parameters](#) subindex 2. After that, reset the module. The new setting then becomes active.

(Available bit rates: 20, 50, 100, 125, 250, 500, 800, 1000)

Object Description			
Index	Name	Object Type	Data Type
2704 <sub>h</sub>	CAN Bit Rate	Variable	UNSIGNED16

Table 102: Object Description (2704<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	CAN Bit Rate	no	20	1000	1000		RW

Table 103: Entry Description (2704<sub>h</sub>)

#### 4.1.25 Object 2705<sub>h</sub>: Node-ID

On modules that do not have address switches the node ID can be selected using this object.

On modules with address switches the node ID is normally selected using the address switches.

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

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

Table 104: Object Description (2705<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Node ID	no	1	127	1		RW

Table 105: Entry Description (2705<sub>h</sub>)

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

This object shows the selected CAN bit rate.

Object Description			
Index	Name	Object Type	Data Type
2707 <sub>h</sub>	CAN Bit Rate Load	Variable	UNSIGNED8

Table 106: Object Description (2707<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	CAN Bit Rate Load	no	20	1000	1000		R

Table 107: Entry Description (2707<sub>h</sub>)

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

This object shows the selected node ID.

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

Table 108: Object Description (2708<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Node ID Load	no	1	127	1		R

Table 109: Entry Description (2708<sub>h</sub>)

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

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

Table 110: Object Description (270E<sub>h</sub>)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	AIN0	no	0	4095	0		R
2	AIN1	no	0	4095	0		R
3	ADC2	no	0	4095	0		R
4	ADC3	no	0	4095	0		R
5	ADC4	no	0	4095	0		R
6	ADC5	no	0	4095	0		R
7	ADC6	no	0	4095	0		R
8	ADC7	no	0	4095	0		R

Table 111: Entry Description (270E<sub>h</sub>)

ADC value of the analog input pins AIN0 to AIN2, as well as ADC values of some on-board voltages.

- AIN0 Analog input pin AIN0 ADC value.
- AIN1 Analog input pin AIN1 ADC value.
- AIN2 Analog input pin AIN2 ADC value.
- AIN3 ADC value of the on-board logic supply voltage measurement circuit.
- AIN4 ADC value of the on-board 5V supply voltage measurement circuit.
- AIN5 Driver ADC value with offset.
- AIN6 Driver ADC value without offset.
- AIN7 No voltage source connected.

## 5 Profile specific Area

The profile segment contains CiA-402 standard motion control objects. These objects control the motion control functions of the TMCM-1617. 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>h</sub>, please see 5.1.6) are implemented on the TMCM-1617:

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

### 5.1 Detailed Object Specifications

#### 5.1.1 Object 605A<sub>h</sub>: Quick Stop Option Code

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

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

Table 112: Value Description (605A<sub>h</sub>)

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

Table 113: Object Description (605A<sub>h</sub>)

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

Table 114: Entry Description (605A<sub>h</sub>)

### 5.1.2 Object 605B<sub>h</sub>: Shutdown Option Code

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

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

Table 115: Value Description (605B<sub>h</sub>)

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

Table 116: Object Description (605B<sub>h</sub>)

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

Table 117: Entry Description (605B<sub>h</sub>)

### 5.1.3 Object 605C<sub>h</sub>: Disable Operation Option Code

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

Value Definition	
Value	Definition
1	Slow down on slow down ramp

Table 118: Value Description (605C<sub>h</sub>)

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

Table 119: Object Description (605C<sub>h</sub>)

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

Table 120: Entry Description (605C<sub>h</sub>)

#### 5.1.4 Object 605D<sub>h</sub>: Halt Option Code

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

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

Table 121: Value Description (605D<sub>h</sub>)

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

Table 122: Object Description (605D<sub>h</sub>)

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

Table 123: Entry Description (605D<sub>h</sub>)

#### 5.1.5 Object 605E<sub>h</sub>: Fault Reaction Option Code

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



Value Definition	
Value	Definition
2	Slow down on quick stop ramp

Table 124: Value Description (605E<sub>h</sub>)

Object Description			
Index	Name	Object Type	Data Type
605E <sub>h</sub>	Fault reaction option code	Variable	UNSIGNED16

Table 125: Object Description (605E<sub>h</sub>)

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

Table 126: Entry Description (605E<sub>h</sub>)

### 5.1.6 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)
8	Cyclic synchronous position mode (csp)
9	Cyclic synchronous velocity mode (csv)
10	Cyclic synchronous torque mode (cst)

Table 127: 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 128: Object Description (6060<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	see table 127	0

Table 129: Entry Description (6060<sub>h</sub>)

### 5.1.7 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)
8	Cyclic synchronous position mode (csp)
9	Cyclic synchronous velocity mode (csv)
10	Cyclic synchronous torque mode (cst)

Table 130: 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 131: Object Description (6061<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	see table 130	0

Table 132: Entry Description (6061<sub>h</sub>)

### 5.1.8 Object 60FD<sub>h</sub>: Digital Inputs

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

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

Table 133: Object Description (60FD<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	UNSIGNED32	0

Table 134: Entry Description (60FD<sub>h</sub>)

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

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

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

Table 135: Value Definition (6502<sub>h</sub>)

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

Table 136: Object Description (6502<sub>h</sub>)

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

Table 137: 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 5.1.6) for information about how to choose an operation mode. Object 6061<sub>h</sub> (section 5.1.7) shows the operation mode that is set.

### 6.1 Detailed Object Specifications

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

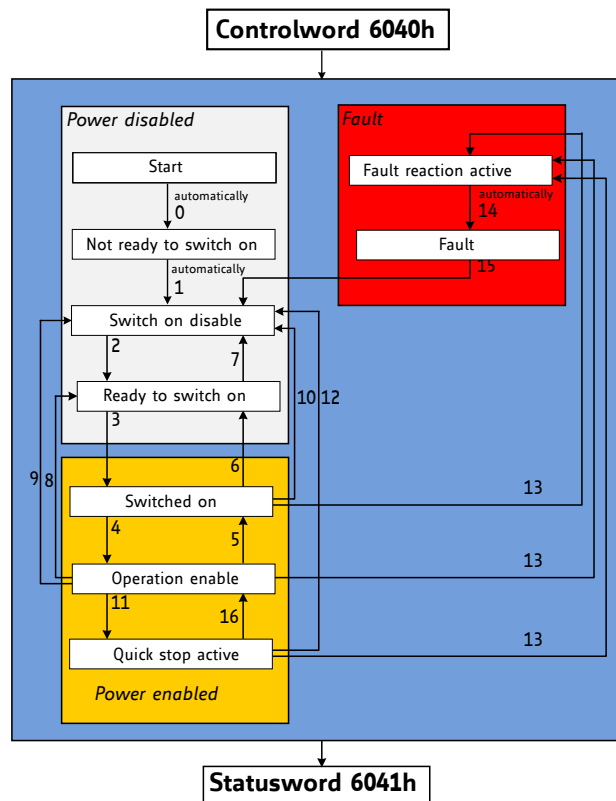


Figure 4: DS402 Finite State Machine

Notes on state transitions:

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

### 6.1.1 Object 6040<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 4 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 138: 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 139: 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 140: Command Coding

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

Table 141: Object Description (6040<sub>h</sub> in pp Mode)

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

Table 142: Entry Description (6040<sub>n</sub> in pp Mode)

### 6.1.2 Object 6041<sub>n</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 4 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 143: Structure of the Status Word in pp Mode

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

Table 144: 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 145: 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 146: State Coding

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

Table 147: Object Description (6041<sub>h</sub> in pp Mode)

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

Table 148: Entry Description (6041<sub>h</sub> in pp Mode)

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

This object provides the demanded position value. The value is given in PositionScaler steps (see [Object 2058<sub>h</sub>: Position Scaler](#)). Object 6062<sub>h</sub> indicates the actual position that the motor should have. It is not to be confused with objects 6063<sub>h</sub> and 6064<sub>h</sub>.

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

Table 149: Object Description (6062<sub>h</sub>)



Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 150: Entry Description (6062<sub>h</sub>)

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

This object provides the demanded position value. The value is given in PositionScaler steps (see [Object 2058<sub>h</sub>: Position Scaler](#)). It is the same as object 6062<sub>h</sub>.

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

Table 151: Object Description (6063<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 152: Entry Description (6063<sub>h</sub>)

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

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

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

Table 153: Object Description (6064<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 154: Entry Description (6064<sub>h</sub>)

### 6.1.6 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  $\text{FFFFFFF}_h$ , the position window control is switched off. If this object is set to zero, the target reached event will be signaled when the demand position ( $6062_h$ ) has reached the target position ( $6064_h$ ). When the position window is set to a value greater than zero, the target reached event will be signaled when the actual encoder position value ( $6064_h$ ) is within  $(\text{target\_position} - \text{position\_window})$  and  $(\text{target\_position} + \text{position\_window})$ .

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

Table 155: Object Description ( $6067_h$ )

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	UNSIGNED32	$\text{FFFFFFF}_h$

Table 156: Entry Description ( $6067_h$ )

### 6.1.7 Object $6068_h$ : Position Window Time

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

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

Table 157: Object Description ( $6068_h$ )

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

Table 158: Entry Description ( $6068_h$ )

### 6.1.8 Object $606C_h$ : Velocity Actual Value

This object shows the actual velocity value of the motor. The value is given in units of mechanical RPM or electrical RPM (depending on the setting made to  $2042_h$  sub index 7).

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

Table 159: Object Description (606C<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 160: Entry Description (606C<sub>h</sub>)

### 6.1.9 Object 607A<sub>h</sub>: Target Position

The target position is the position that the drive should move to in profile position mode using the current settings of motion control parameters (such as velocity, acceleration, deceleration, motion profile type etc.). The value of this object is interpreted as absolute or relative depending on the abs/rel flag in the controlword. It is given in PositionScaler steps (see [Object 2058<sub>h</sub>: Position Scaler](#)).

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

Table 161: Object Description (607A<sub>h</sub> in pp Mode)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	SIGNED32	0

Table 162: Entry Description (607A<sub>h</sub> in pp Mode)

### 6.1.10 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:

$$\text{Corrected\_min\_position\_limit} = \text{min\_position\_limit} - \text{home\_offset}$$

$$\text{Corrected\_max\_position\_limit} = \text{max\_position\_limit} - \text{home\_offset}$$

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

Table 163: Object Description (607D<sub>h</sub>)

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

Table 164: Entry Description (607D<sub>h</sub>)

### 6.1.11 Object 6081<sub>h</sub>: Profile Velocity

This object indicates the configured velocity normally attained at the end of the acceleration ramp during a profiled motion and is valid for both directions of motion. The profile velocity is the maximum velocity used when driving to a new position. It is given in units of RPM/s.

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

Table 165: Object Description (6081<sub>h</sub>)

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

Table 166: Entry Description (6081<sub>h</sub>)

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

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

This value is given using RPM/s units.

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

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

Table 167: Object Description (6083<sub>h</sub>)

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

Table 168: Entry Description (6083<sub>h</sub>)

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

This object indicates the configured deceleration. Object 6084<sub>h</sub> sets the maximum deceleration to be used in profile positioning mode. This value is given in units of RPM/s.

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

Table 169: Object Description (6084<sub>h</sub>)

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

Table 170: Entry Description (6084<sub>h</sub>)

### 6.1.14 Object 6085<sub>h</sub>: Quick Stop Deceleration

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

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

Table 171: Object Description (6085<sub>h</sub>)

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

Table 172: Entry Description (6085<sub>h</sub>)

## 6.2 How to move a Motor in pp Mode

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

- If you do not have any limit switches connected, first disable the limit switch inputs by writing 3 to object 2005<sub>h</sub>.
- Select pp mode by writing 1 to object 6060<sub>h</sub>.
- Write 6 to object 6040<sub>h</sub> to switch to READY\_TO\_SWITCH\_ON state.
- Write 7 to object 6040<sub>h</sub> to switch to SWITCHED\_ON state.
- Write 15 to object 6040<sub>h</sub> to switch to OPERATION\_ENABLED state.
- Write the desired target position (e.g. 500000) to object 607A<sub>h</sub>.
- Mark the new target position as active by writing 31 to object 6040<sub>h</sub>. The motor starts moving now.
- Reset the activation by writing 15 to object 6040<sub>h</sub> (this can be done while the motor is still moving).

## 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

### 7.0.1 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 4 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 173: 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 174: Command Coding

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

Table 175: Object Description (6040<sub>h</sub> in pv Mode)

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

Table 176: Entry Description (6040<sub>h</sub> in pv Mode)

### 7.0.2 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 4 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 177: 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 178: 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 179: 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 180: State Coding

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

Table 181: Object Description (6041<sub>h</sub> in pv Mode)

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

Table 182: Entry Description (6041<sub>h</sub> in pv Mode)

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

This object provides the demanded position value. The value is given in PositionScaler steps (see [Object 2058<sub>h</sub>: Position Scaler](#)). Object 6062<sub>h</sub> indicates the actual position that the motor should have. It is not to be confused with objects 6063<sub>h</sub> and 6064<sub>h</sub>.

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

Table 183: Object Description (6062<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 184: Entry Description (6062<sub>h</sub>)

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

This object provides the demanded position value. The value is given in PositionScaler steps (see [Object 2058<sub>h</sub>: Position Scaler](#)). It is the same as object 6062<sub>h</sub>.

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

Table 185: Object Description (6063<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 186: Entry Description (6063<sub>h</sub>)

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

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

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

Table 187: Object Description (6064<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 188: Entry Description (6064<sub>h</sub>)

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

This object shows the actual velocity value of the motor. The value is given in units of mechanical RPM or electrical RPM (depending on the setting made to 2042<sub>h</sub> sub index 7).

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

Table 189: Object Description (606C<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 190: Entry Description (606C<sub>h</sub>)

### 7.0.7 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:

$$\text{Corrected\_min\_position\_limit} = \text{min\_position\_limit} - \text{home\_offset}$$

$$\text{Corrected\_max\_position\_limit} = \text{max\_position\_limit} - \text{home\_offset}$$

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

Table 191: Object Description (607D<sub>h</sub>)

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

Table 192: Entry Description (607D<sub>h</sub>)

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

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

This value is given using RPM/s units.

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

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

Table 193: Object Description (6083<sub>h</sub>)

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

Table 194: Entry Description (6083<sub>h</sub>)

### 7.0.9 Object 6085<sub>h</sub>: Quick Stop Deceleration

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

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

Table 195: Object Description (6085<sub>h</sub>)

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

Table 196: Entry Description (6085<sub>h</sub>)

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

This object indicates the configured target velocity and is used as input for the trajectory generator. Object 60FF<sub>h</sub> sets the target velocity when using profile velocity mode. The drive then accelerates or decelerates to that velocity using the acceleration and deceleration set by objects 6083<sub>h</sub> and 6084<sub>h</sub>. The values are given in mechanical RPM or electrical RPM (depending on the setting made to 2042<sub>h</sub> sub index 7) units.

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

Table 197: Object Description (60FF<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	SIGNED32	0

Table 198: Entry Description (60FF<sub>h</sub>)

## 7.1 How to move a Motor in pv Mode

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

- If you do not have any limit switches connected, first disable the limit switch inputs by writing 3 to object 2005<sub>h</sub>.
- Select pv mode by writing 3 to object 6060<sub>h</sub>.
- Write 6 to object 6040<sub>h</sub> to switch to READY\_TO\_SWITCH\_ON state.
- Write 7 to object 6040<sub>h</sub> to switch to SWITCHED\_ON state.
- Write 15 to object 6040<sub>h</sub> to switch to OPERATION\_ENABLED state.
- Write the desired target speed (e.g. 100000) to object 60FF<sub>h</sub>. The motor now accelerates to that speed.
- Stop the motor by writing 0 to object 60FF<sub>h</sub>.

## 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 5 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.

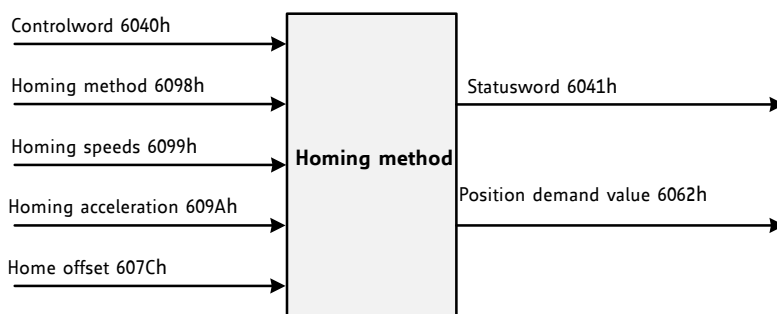


Figure 5: Homing Mode Function

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).

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

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

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

## 8.1 Homing Methods

The TMCM-1617 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.
19	Search the positive edge of the home switch.
21	Search the negative edge of the home 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 199: 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>, section 4.1.4).

### 8.1.1 Homing Method 17: Homing on negative Limit Switch

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

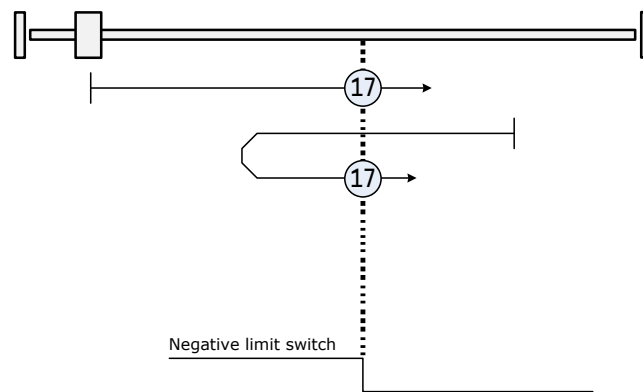


Figure 6: Homing Method 17

### 8.1.2 Homing Method 18: Homing on positive Limit Switch

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

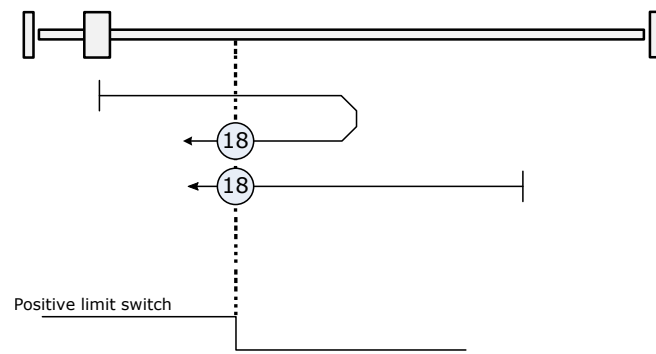


Figure 7: Homing Method 18

### 8.1.3 Homing Method 19: Homing on positive Home Switch

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

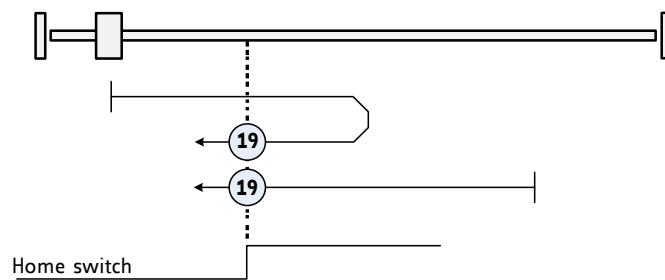


Figure 8: Homing Method 19

### 8.1.4 Homing Method 21: Homing on negative Home Switch

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

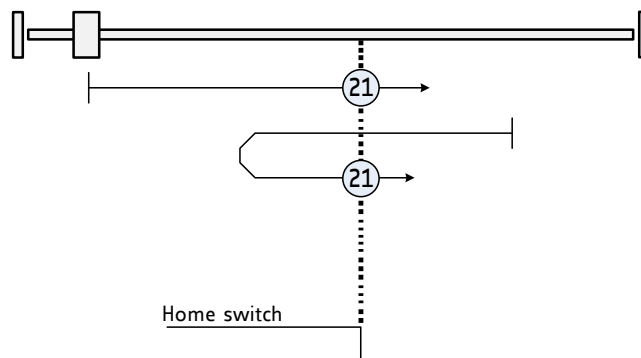


Figure 9: Homing Method 21



### 8.1.5 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 4 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 200: 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 201: 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 202: Command Coding

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

Table 203: Object Description (6040<sub>h</sub> in hm Mode)

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

Table 204: 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 4 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 205: 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 206: 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 207: 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 208: State Coding

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

Table 209: Object Description (6041<sub>h</sub> in hm Mode)

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

Table 210: 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 of the motor. The value is given in units of mechanical RPM or electrical RPM (depending on the setting made to 2042<sub>h</sub> sub index 7).

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

Table 211: Object Description (606C<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 212: 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. The value of this object is given in PositionScaler steps (see [Object 2058<sub>h</sub>: Position Scaler](#)). Negative values indicate the opposite direction.

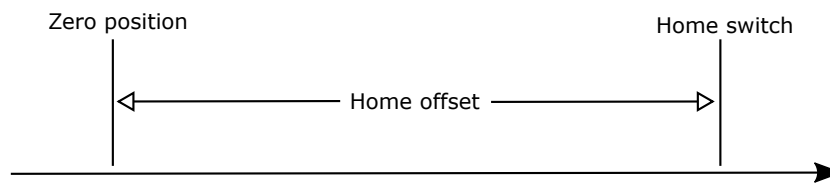


Figure 10: Home Offset

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

Table 213: Object Description (607C<sub>h</sub>)

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

Table 214: Entry Description (607C<sub>h</sub>)

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

The homing method to be used can be selected by writing to this object. Please see table 199 for a list of homing methods supported by the current version of the TMCM-1617 CANopen firmware.

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

Table 215: Object Description (6098<sub>h</sub>)

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

Table 216: Entry Description (6098<sub>h</sub>)

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

This object indicates the configured speeds used during homing procedure. The values are given in mechanical RPM or electrical RPM (depending on the setting made to 2042<sub>h</sub> sub index 7) units. Using object 6099<sub>h</sub> a fast and a slow homing speed can be set. In most homing modes, the home switch is searched with the fast speed first. When the home switch has been found, the motor will be decelerated to the slow speed (using the homing acceleration, object 609A<sub>h</sub>) 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 217: Object Description (6099<sub>h</sub>)

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

Table 218: 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. This object used RPM/s units.

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

Table 219: Object Description (609A<sub>h</sub>)

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

Table 220: Entry Description (609A<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 19 by writing 19 to object 6098<sub>h</sub>.
- Set the homing speeds by writing e.g. 50000 to object 6099<sub>h</sub> sub index 1 and e.g. 10000 to object 6099<sub>h</sub> sub index 2.
- Write 31 to object 6040<sub>h</sub> to start the homing process.
- Press and release the home switch.
- When homing has finished, write 15 to object 6040<sub>h</sub> again.

## 9 Cyclic synchronous Position Mode

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

The main control parameters are the target position (object 607A<sub>h</sub>, see section 9.1.7) and the interpolation time period (object 60C2<sub>h</sub>, see section 9.1.10). The drive automatically sets the velocity in such a manner that the next target position is reached within the interpolation time period. Acceleration and deceleration ramps are not used in this mode.

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

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

### 9.1 Detailed Object Specifications

#### 9.1.1 Object 6040<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 4 for detailed information. The cyclic synchronous position mode does not use any mode specific bits of the control word.

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

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

Table 221: Structure of the Control Word in csp Mode



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

Table 222: Command Coding

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

Table 223: Object Description (6040<sub>h</sub> in csp Mode)

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

Table 224: Entry Description (6040<sub>h</sub> in csp 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 4 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 225: Structure of the Status Word in csp Mode

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

Table 226: Trinamic Specific Bits

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

Table 227: Operation Mode specific Bits in csp 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 228: State Coding

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

Table 229: Object Description (6041<sub>h</sub> in csp Mode)

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

Table 230: Entry Description (6041<sub>h</sub> in csp Mode)

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

This object provides the demanded position value. The value is given in PositionScaler steps (see [Object 2058<sub>h</sub>: Position Scaler](#)). Object 6062<sub>h</sub> indicates the actual position that the motor should have. It is not to be confused with objects 6063<sub>h</sub> and 6064<sub>h</sub>.

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

Table 231: Object Description (6062<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 232: Entry Description (6062<sub>h</sub>)

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

This object provides the demanded position value. The value is given in PositionScaler steps (see [Object 2058<sub>h</sub>: Position Scaler](#)). It is the same as object 6062<sub>h</sub>.

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

Table 233: Object Description (6063<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 234: Entry Description (6063<sub>h</sub>)

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

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

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

Table 235: Object Description (6064<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 236: Entry Description (6064<sub>h</sub>)

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

This object shows the actual velocity value of the motor. The value is given in units of mechanical RPM or electrical RPM (depending on the setting made to 2042<sub>h</sub> sub index 7).

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

Table 237: Object Description (606C<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 238: Entry Description (606C<sub>h</sub>)

### 9.1.7 Object 607A<sub>h</sub>: Target Position

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

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

Table 239: Object Description (607A<sub>h</sub> in csp Mode)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	SIGNED32	0

Table 240: Entry Description (607A<sub>h</sub> in csp Mode)

### 9.1.8 Object 607D<sub>h</sub>: Software Position Limit

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

$$\text{Corrected\_min\_position\_limit} = \text{min\_position\_limit} - \text{home\_offset}$$

$$\text{Corrected\_max\_position\_limit} = \text{max\_position\_limit} - \text{home\_offset}$$

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

Table 241: Object Description (607D<sub>h</sub>)

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

Table 242: Entry Description (607D<sub>h</sub>)

### 9.1.9 Object 60B0<sub>h</sub>: Position Offset

This object provides an offset to the target position (object 607A<sub>h</sub>, see section 9.1.7)). The value is given in PositionScaler steps (see Object 2058<sub>h</sub>: Position Scaler) and will be added to the target position.

Object Description			
Index	Name	Object Type	Data Type
60B0 <sub>h</sub>	Position Offset	Variable	SIGNED32

Table 243: Object Description (60B0<sub>h</sub>)

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

Table 244: Entry Description (60B0<sub>h</sub>)

### 9.1.10 Object 60C2<sub>h</sub>: Interpolation Time Period

This object indicates the interpolation cycle time. The interpolation time period (sub-index 01<sub>h</sub>) is given in  $10^{\text{interpolation\_time\_index}}$  s. The interpolation time index (sub-index 02<sub>h</sub>) is dimensionless.

Object Description			
Index	Name	Object Type	Data Type
60C2 <sub>h</sub>	Interpolation Time Period	Record	Interpolation time period record (0080 <sub>h</sub> )

Table 245: Object Description (60C2<sub>h</sub>)

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

Table 246: Entry Description (60C2<sub>h</sub>)

## 10 Cyclic synchronous Velocity Mode

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

The main control parameters are the target velocity (object 60FF<sub>h</sub>, see section 10.1.4) and the interpolation time period (object 60C2<sub>h</sub>, see section 10.1.7). The drive automatically sets the acceleration in such a manner that the next target velocity is reached within the interpolation time period. Acceleration and deceleration ramps are not used in this mode.

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

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

### 10.1 Detailed Object Specifications

#### 10.1.1 Object 6040<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 4 for detailed information. The cyclic synchronous velocity mode does not use any mode specific bits of the control word.

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

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

Table 247: Structure of the Control Word in csv Mode

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

Table 248: Command Coding

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

Table 249: Object Description (6040<sub>h</sub> in csv Mode)

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

Table 250: Entry Description (6040<sub>h</sub> in csv Mode)

### 10.1.2 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 4 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 251: Structure of the Status Word in csv Mode



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

Table 252: Trinamic Specific Bits

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

Table 253: Operation Mode specific Bits in csv 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 254: State Coding

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

Table 255: Object Description (6041<sub>h</sub> in csv Mode)

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

Table 256: Entry Description (6041<sub>h</sub> in csv Mode)

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

This object shows the actual velocity value of the motor. The value is given in units of mechanical RPM or electrical RPM (depending on the setting made to 2042<sub>h</sub> sub index 7).

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

Table 257: Object Description (606C<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 258: Entry Description (606C<sub>h</sub>)

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

In csv mode the target velocity specifies the velocity that is to be reached within the interpolation time period. The values are given in mechanical RPM or electrical RPM (depending on the setting made to 2042<sub>h</sub> sub index 7) units.

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

Table 259: Object Description (60FF<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	SIGNED32	0

Table 260: Entry Description (60FF<sub>h</sub>)

### 10.1.5 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:

$$\begin{aligned} \text{Corrected\_min\_position\_limit} &= \text{min\_position\_limit} - \text{home\_offset} \\ \text{Corrected\_max\_position\_limit} &= \text{max\_position\_limit} - \text{home\_offset} \end{aligned}$$

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

Table 261: Object Description (607D<sub>h</sub>)

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

Table 262: Entry Description (607D<sub>h</sub>)

### 10.1.6 Object 60B1<sub>h</sub>: Velocity Offset

This object provides an offset to the target velocity (object 60FF<sub>h</sub>, see section 10.1.4)). The value will be added to the target velocity.

Object Description			
Index	Name	Object Type	Data Type
60B1 <sub>h</sub>	Velocity Offset	Variable	SIGNED32

Table 263: Object Description (60B1<sub>h</sub>)

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

Table 264: Entry Description (60B1<sub>h</sub>)

### 10.1.7 Object 60C2<sub>h</sub>: Interpolation Time Period

This object indicates the interpolation cycle time. The interpolation time period (sub-index 01<sub>h</sub>) is given in  $10^{\text{interpolation\_time\_index}}$  s. The interpolation time index (sub-index 02<sub>h</sub>) is dimensionless.

Object Description			
Index	Name	Object Type	Data Type
60C2 <sub>h</sub>	Interpolation Time Period	Record	Interpolation time period record (0080 <sub>h</sub> )

Table 265: Object Description (60C2<sub>h</sub>)

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

Table 266: Entry Description (60C2<sub>h</sub>)

## 11 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.

### 11.1 Detailed Object Specifications

#### 11.1.1 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 4 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 267: 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 268: Command Coding

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

Table 269: Object Description (6040<sub>h</sub> in cst Mode)

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

Table 270: Entry Description (6040<sub>h</sub> in cst Mode)

### 11.1.2 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 4 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 271: 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 272: 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 273: 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 274: State Coding

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

Table 275: Object Description (6041<sub>h</sub> in cst Mode)

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

Table 276: Entry Description (6041<sub>h</sub> in cst Mode)

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

This object provides the demanded position value. The value is given in PositionScaler steps (see [Object 2058<sub>h</sub>: Position Scaler](#)). Object 6062<sub>h</sub> indicates the actual position that the motor should have. It is not to be confused with objects 6063<sub>h</sub> and 6064<sub>h</sub>.

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

Table 277: Object Description (6062<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 278: Entry Description (6062<sub>h</sub>)

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

This object provides the demanded position value. The value is given in PositionScaler steps (see [Object 2058<sub>h</sub>: Position Scaler](#)). It is the same as object 6062<sub>h</sub>.

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

Table 279: Object Description (6063<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 280: Entry Description (6063<sub>h</sub>)

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

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

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

Table 281: Object Description (6064<sub>h</sub>)



Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 282: Entry Description (6064<sub>h</sub>)

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

This object sets the desired torque value. The value is given in mA.

Object Description			
Index	Name	Object Type	Data Type
6071 <sub>h</sub>	Target torque	Variable	INTEGER16

Table 283: Object Description (6071<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	-32768...32767	0

Table 284: Entry Description (6071<sub>h</sub>)

### 11.1.7 Object 6077<sub>h</sub>: Torque actual Value

This object provides the actual torque value. The value is given in mA.

Object Description			
Index	Name	Object Type	Data Type
6077 <sub>h</sub>	Torque actual Value	Variable	INTEGER16

Table 285: Object Description (6077<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	-32768...32767	0

Table 286: Entry Description (6077<sub>h</sub>)

### 11.1.8 Object 607D<sub>h</sub>: Software Position Limit

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

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

$$\begin{aligned} \text{Corrected\_min\_position\_limit} &= \text{min\_position\_limit} - \text{home\_offset} \\ \text{Corrected\_max\_position\_limit} &= \text{max\_position\_limit} - \text{home\_offset} \end{aligned}$$

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

Table 287: Object Description (607D<sub>h</sub>)

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

Table 288: Entry Description (607D<sub>h</sub>)

### 11.1.9 Object 60B2<sub>h</sub>: Torque Offset

This object provides an offset to the torque value. It will be added to the target torque (object 6071<sub>h</sub>, see section 11.1.6).

Object Description			
Index	Name	Object Type	Data Type
60B2 <sub>h</sub>	Torque Offset	Variable	SIGNED16

Table 289: Object Description (60B2<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	-32768...32767	0

Table 290: Entry Description (60B2<sub>h</sub>)

### 11.1.10 Object 60C2<sub>h</sub>: Interpolation Time Period

This object indicates the interpolation cycle time. The interpolation time period (sub-index 01<sub>h</sub>) is given in  $10^{\text{interpolation\_time\_index}}$  s. The interpolation time index (sub-index 02<sub>h</sub>) is dimensionless.

Object Description			
Index	Name	Object Type	Data Type
60C2 <sub>h</sub>	Interpolation Time Period	Record	Interpolation time period record (0080 <sub>h</sub> )

*Table 291: Object Description (60C2<sub>h</sub>)*

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

*Table 292: Entry Description (60C2<sub>h</sub>)*

## 12 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 TMC-1617						
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.
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.
8100 <sub>h</sub>	0	255	0	0	0	<b>Communication error</b> General CAN bus communication error.
8110 <sub>h</sub>	1	255	0	0	0	<b>CAN controller overflow</b> The receive message buffer of the CAN controller hardware is full and some CAN messages are lost.
8110 <sub>h</sub>	2	255	0	0	0	<b>CAN Tx buffer overflow</b> The software CAN transmit buffer is full and thus some CAN messages are lost.
8110 <sub>h</sub>	3	255	0	0	0	<b>CAN Rx buffer overflow</b> The software CAN receive buffer is full and so some CAN messages are lost.
8120 <sub>h</sub>	0	255	0	0	0	<b>CAN error passive</b> The CAN controller has detected communication errors and has entered the CAN Error passive state.
8130 <sub>h</sub>	0	255	0	0	0	<b>Heartbeat or lifeguard error</b> The module did not receive a heartbeat or lifeguard message in time.
8140 <sub>h</sub>	0	255	0	0	0	<b>CAN controller recovered from bus-off state</b> The CAN controller has detected too many errors and has changed into the bus-off state. The drive has been stopped and disabled. This message is sent after the CAN controller has recovered from bus-off state and is bus-on again.
8210 <sub>h</sub>	0	255	0	0	0	<b>PDO not processed due to length error</b> A PDO sent to the module could not be processed because too few bytes were supplied.

Error code	Additional byte					Description
	1	2	3	4	5	
8220 <sub>h</sub>	0	255	0	0	0	<b>PDO length exceeded</b> A PDO sent to the module could not be processed because too many bytes were supplied.
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 293: Emergency Messages (EMCY)

## 13 SDO Abort Codes

Trying to access an object via SDO read or SDO write may result in an error. In such a case an SDO abort transfer message containing an abort code will be sent. The following table lists all SDO abort codes defined by the CiA-301 standard. Not all of these are used by the TMCM-1617 module.

SDO Abort Codes	
Abort code	Description
05030000 <sub>h</sub>	Toggle bit not alternated.
05040000 <sub>h</sub>	SDO protocol timed out.
05040001 <sub>h</sub>	Client/server command specifier not valid or unknown.
05040002 <sub>h</sub>	Invalid block size.
05040003 <sub>h</sub>	Invalid sequence number.
05040004 <sub>h</sub>	CRC error.
05040005 <sub>h</sub>	Out of memory.
06010000 <sub>h</sub>	Unsupported access to an object.
06010001 <sub>h</sub>	Attempt to read a write only object.
06010002 <sub>h</sub>	Attempt to write a read only object.
06020000 <sub>h</sub>	Object does not exist in object dictionary.
06040041 <sub>h</sub>	Object cannot be mapped to the PDO.
06040042 <sub>h</sub>	The number and length of the objects to be mapped would exceed the PDO length.
06040043 <sub>h</sub>	General parameter incompatibility reason.
06040047 <sub>h</sub>	General internal incompatibility in the device.
06060000 <sub>h</sub>	Access failed due to a hardware error.
06070010 <sub>h</sub>	Data type does not match, length of service parameter does not match.
06070012 <sub>h</sub>	Data type does not match, length of service parameter too high.
06070013 <sub>h</sub>	Data type does not match, length of service parameter too low.
06090011 <sub>h</sub>	Sub-index does not exist.
06090030 <sub>h</sub>	Invalid value for parameter.
06090031 <sub>h</sub>	Value of parameter too high.
06090032 <sub>h</sub>	Value of parameter too low.
06090036 <sub>h</sub>	Maximum value is less than minimum value.
060A0023 <sub>h</sub>	Resource not available.
08000000 <sub>h</sub>	General error.
08000020 <sub>h</sub>	Data cannot be transferred or stored to the application.
08000021 <sub>h</sub>	Data cannot be transferred or stored to the application because of local control.

Abort code	Description
08000022 <sub>h</sub>	Data cannot be transferred or stored to the application because of the present device state.
08000023 <sub>h</sub>	Object dictionary dynamic generation failed or no object dictionary is present.
08000024 <sub>h</sub>	No data available.

*Table 294: SDO Abort Codes*

## 14 Firmware Update

For an update of the TMCM-1617's firmware the TMCL-IDE <sup>1</sup> is used.

The required update files can be downloaded from the TMCM-1617's product page <sup>2</sup>.

An update over CAN is only possible via the Trinamic TMCL CAN protocol, because of that, prior to starting the upload, the TMCM-1617 must be put into boot mode by writing the value 12345678<sub>h</sub> into object 5FFF<sub>h</sub>.

---

**Note** Make sure to update only CANopen firmware files.

---

---

<sup>1</sup>TMCL-IDE: <https://www.trinamic.com/support/software/tmcl-ide>

<sup>2</sup>TMCM-1617 product page: <https://www.trinamic.com/products/modules/details/tmcm-1617/>



## 15 Figures Index

1	NMT State Machine . . . . .	10	6	Homing Method 17 . . . . .	71
2	Communication Architecture . . . . .	11	7	Homing Method 18 . . . . .	72
3	Device Model . . . . .	12	8	Homing Method 19 . . . . .	72
4	DS402 Finite State Machine . . . . .	53	9	Homing Method 21 . . . . .	72
5	Homing Mode Function . . . . .	70	10	Home Offset . . . . .	77

## 16 Tables Index

1	Abbreviations used in this Manual . . .	7	53	Entry Description (2003 <sub>h</sub> ) . . . . .	29
2	Service Primitives . . . . .	8	54	Object Description (2004 <sub>h</sub> ) . . . . .	29
3	Service Types . . . . .	9	55	Entry Description (2004 <sub>h</sub> ) . . . . .	29
4	Object Dictionary . . . . .	12	56	Object Description (2005 <sub>h</sub> ) . . . . .	29
5	Object Description (1000 <sub>h</sub> ) . . . . .	14	57	Entry Description (2005 <sub>h</sub> ) . . . . .	30
6	Entry Description (1000 <sub>h</sub> ) . . . . .	14	58	Bit Definitions (2005 <sub>h</sub> ) . . . . .	30
7	Object Description (1001 <sub>h</sub> ) . . . . .	14	59	Object Description (2006 <sub>h</sub> ) . . . . .	30
8	Entry Description (1001 <sub>h</sub> ) . . . . .	14	60	Entry Description (2006 <sub>h</sub> ) . . . . .	30
9	Error Register Bits . . . . .	15	61	Object Description (2041 <sub>h</sub> ) . . . . .	31
10	Value Definition (1005 <sub>h</sub> ) . . . . .	15	62	Entry Description (2041 <sub>h</sub> ) . . . . .	31
11	Object Description (1005 <sub>h</sub> ) . . . . .	15	63	Object Description (2042 <sub>h</sub> ) . . . . .	31
12	Entry Description (1005 <sub>h</sub> ) . . . . .	15	64	Entry Description (2042 <sub>h</sub> ) . . . . .	32
13	Object Description (1008 <sub>h</sub> ) . . . . .	16	65	Object Description (2043 <sub>h</sub> ) . . . . .	32
14	Entry Description (1008 <sub>h</sub> ) . . . . .	16	66	Entry Description (2043 <sub>h</sub> ) . . . . .	33
15	Object Description (1009 <sub>h</sub> ) . . . . .	16	67	Object Description (2050 <sub>h</sub> ) . . . . .	33
16	Entry Description (1009 <sub>h</sub> ) . . . . .	16	68	Entry Description (2050 <sub>h</sub> ) . . . . .	33
17	Object Description (100A <sub>h</sub> ) . . . . .	16	69	Commutation Modes . . . . .	34
18	Entry Description (100A <sub>h</sub> ) . . . . .	17	70	Object Description (2055 <sub>h</sub> ) . . . . .	34
19	Object Description (100C <sub>h</sub> ) . . . . .	17	71	Entry Description (2055 <sub>h</sub> ) . . . . .	34
20	Entry Description (100C <sub>h</sub> ) . . . . .	17	72	Object Description (2056 <sub>h</sub> ) . . . . .	34
21	Object Description (100D <sub>h</sub> ) . . . . .	17	73	Entry Description (2056 <sub>h</sub> ) . . . . .	34
22	Entry Description (100D <sub>h</sub> ) . . . . .	17	74	Object Description (2057 <sub>h</sub> ) . . . . .	35
23	Save Signature . . . . .	18	75	Entry Description (2057 <sub>h</sub> ) . . . . .	35
24	Object Description (1010 <sub>h</sub> ) . . . . .	18	76	Object Description (2058 <sub>h</sub> ) . . . . .	35
25	Entry Description (1010 <sub>h</sub> ) . . . . .	19	77	Entry Description (2058 <sub>h</sub> ) . . . . .	35
26	Load Signature . . . . .	19	78	Object Description (2060 <sub>h</sub> ) . . . . .	36
27	Object Description (1011 <sub>h</sub> ) . . . . .	20	79	Entry Description (2060 <sub>h</sub> ) . . . . .	36
28	Entry Description (1011 <sub>h</sub> ) . . . . .	20	80	Object Description (2070 <sub>h</sub> ) . . . . .	36
29	Object Description (1014 <sub>h</sub> ) . . . . .	20	81	Entry Description (2070 <sub>h</sub> ) . . . . .	36
30	Entry Description (1014 <sub>h</sub> ) . . . . .	20	82	Encoder Initialization Modes . . . . .	37
31	Object Description (1015 <sub>h</sub> ) . . . . .	21	83	Object Description (2080 <sub>h</sub> ) . . . . .	37
32	Entry Description (1015 <sub>h</sub> ) . . . . .	21	84	Entry Description (2080 <sub>h</sub> ) . . . . .	38
33	Value Definition (1016 <sub>h</sub> ) . . . . .	21	85	Motor Status Flags (2101 <sub>h</sub> ) . . . . .	38
34	Object Description (1016 <sub>h</sub> ) . . . . .	21	86	Object Description (2101 <sub>h</sub> ) . . . . .	39
35	Entry Description (1016 <sub>h</sub> ) . . . . .	21	87	Entry Description (2101 <sub>h</sub> ) . . . . .	39
36	Object Description (1017 <sub>h</sub> ) . . . . .	22	88	Object Description (2102 <sub>h</sub> ) . . . . .	39
37	Entry Description (1017 <sub>h</sub> ) . . . . .	22	89	Entry Description (2102 <sub>h</sub> ) . . . . .	39
38	Object Description (1018 <sub>h</sub> ) . . . . .	22	90	Object Description (2103 <sub>h</sub> ) . . . . .	39
39	Entry Description (1018 <sub>h</sub> ) . . . . .	22	91	Entry Description (2103 <sub>h</sub> ) . . . . .	40
40	Object Description (1029 <sub>h</sub> ) . . . . .	23	92	Object Description (2104 <sub>h</sub> ) . . . . .	40
41	Entry Description (1029 <sub>h</sub> ) . . . . .	23	93	Entry Description (2104 <sub>h</sub> ) . . . . .	40
42	Object Description (1400 <sub>h</sub> ) . . . . .	24	94	Object Description (2140 <sub>h</sub> ) . . . . .	40
43	Entry Description (1400 <sub>h</sub> ) . . . . .	24	95	Entry Description (2140 <sub>h</sub> ) . . . . .	40
44	Object Description (1600 <sub>h</sub> ) . . . . .	24	96	Bit Definitions (2702 <sub>h</sub> ) . . . . .	41
45	Entry Description (1600 <sub>h</sub> ) . . . . .	25	97	Object Description (2702 <sub>h</sub> ) . . . . .	41
46	Object Description (1800 <sub>h</sub> ) . . . . .	26	98	Entry Description (2702 <sub>h</sub> ) . . . . .	41
47	Entry Description (1800 <sub>h</sub> ) . . . . .	26	99	Bit Definitions (2703 <sub>h</sub> ) . . . . .	41
48	Object Description (1A00 <sub>h</sub> ) . . . . .	27	100	Object Description (2703 <sub>h</sub> ) . . . . .	42
49	Entry Description (1A00 <sub>h</sub> ) . . . . .	27	101	Entry Description (2703 <sub>h</sub> ) . . . . .	42
50	Object Description (2000 <sub>h</sub> ) . . . . .	28	102	Object Description (2704 <sub>h</sub> ) . . . . .	42
51	Entry Description (2000 <sub>h</sub> ) . . . . .	28	103	Entry Description (2704 <sub>h</sub> ) . . . . .	42
52	Object Description (2003 <sub>h</sub> ) . . . . .	28	104	Object Description (2705 <sub>h</sub> ) . . . . .	43

105	Entry Description (2705 <sub>h</sub> )	43	156	Entry Description (6067 <sub>h</sub> )	58
106	Object Description (2707 <sub>h</sub> )	43	157	Object Description (6068 <sub>h</sub> )	58
107	Entry Description (2707 <sub>h</sub> )	43	158	Entry Description (6068 <sub>h</sub> )	58
108	Object Description (2708 <sub>h</sub> )	43	159	Object Description (606C <sub>h</sub> )	59
109	Entry Description (2708 <sub>h</sub> )	44	160	Entry Description (606C <sub>h</sub> )	59
110	Object Description (270E <sub>h</sub> )	44	161	Object Description (607A <sub>h</sub> in pp Mode)	59
111	Entry Description (270E <sub>h</sub> )	44	162	Entry Description (607A <sub>h</sub> in pp Mode)	59
112	Value Description (605A <sub>h</sub> )	46	163	Object Description (607D <sub>h</sub> )	60
113	Object Description (605A <sub>h</sub> )	46	164	Entry Description (607D <sub>h</sub> )	60
114	Entry Description (605A <sub>h</sub> )	47	165	Object Description (6081 <sub>h</sub> )	60
115	Value Description (605B <sub>h</sub> )	47	166	Entry Description (6081 <sub>h</sub> )	60
116	Object Description (605B <sub>h</sub> )	47	167	Object Description (6083 <sub>h</sub> )	60
117	Entry Description (605B <sub>h</sub> )	47	168	Entry Description (6083 <sub>h</sub> )	61
118	Value Description (605C <sub>h</sub> )	47	169	Object Description (6084 <sub>h</sub> )	61
119	Object Description (605C <sub>h</sub> )	48	170	Entry Description (6084 <sub>h</sub> )	61
120	Entry Description (605C <sub>h</sub> )	48	171	Object Description (6085 <sub>h</sub> )	61
121	Value Description (605D <sub>h</sub> )	48	172	Entry Description (6085 <sub>h</sub> )	61
122	Object Description (605D <sub>h</sub> )	48	173	Structure of the Control Word in pv Mode	63
123	Entry Description (605D <sub>h</sub> )	48	174	Command Coding	64
124	Value Description (605E <sub>h</sub> )	49	175	Object Description (6040 <sub>h</sub> in pv Mode)	64
125	Object Description (605E <sub>h</sub> )	49	176	Entry Description (6040 <sub>h</sub> in pv Mode)	64
126	Entry Description (605E <sub>h</sub> )	49	177	Structure of the Status Word in pv Mode	64
127	Value Description (6060 <sub>h</sub> )	49	178	Trinamic Specific Bits	65
128	Object Description (6060 <sub>h</sub> )	50	179	Operation Mode specific Bits in pv Mode	65
129	Entry Description (6060 <sub>h</sub> )	50	180	State Coding	65
130	Value Description (6061 <sub>h</sub> )	50	181	Object Description (6041 <sub>h</sub> in pv Mode)	65
131	Object Description (6061 <sub>h</sub> )	50	182	Entry Description (6041 <sub>h</sub> in pv Mode)	66
132	Entry Description (6061 <sub>h</sub> )	51	183	Object Description (6062 <sub>h</sub> )	66
133	Object Description (60FD <sub>h</sub> )	51	184	Entry Description (6062 <sub>h</sub> )	66
134	Entry Description (60FD <sub>h</sub> )	51	185	Object Description (6063 <sub>h</sub> )	66
135	Value Definition (6502 <sub>h</sub> )	52	186	Entry Description (6063 <sub>h</sub> )	66
136	Object Description (6502 <sub>h</sub> )	52	187	Object Description (6064 <sub>h</sub> )	67
137	Entry Description (6502 <sub>h</sub> )	52	188	Entry Description (6064 <sub>h</sub> )	67
138	Structure of the Control Word in pp Mode	54	189	Object Description (606C <sub>h</sub> )	67
139	Operation Mode specific Bits in pp Mode	54	190	Entry Description (606C <sub>h</sub> )	67
140	Command Coding	54	191	Object Description (607D <sub>h</sub> )	68
141	Object Description (6040 <sub>h</sub> in pp Mode)	54	192	Entry Description (607D <sub>h</sub> )	68
142	Entry Description (6040 <sub>h</sub> in pp Mode)	55	193	Object Description (6083 <sub>h</sub> )	68
143	Structure of the Staus Word in pp Mode	55	194	Entry Description (6083 <sub>h</sub> )	68
144	Trinamic Specific Bits	55	195	Object Description (6085 <sub>h</sub> )	68
145	Operation Mode specific Bits in pp Mode	55	196	Entry Description (6085 <sub>h</sub> )	69
146	State Coding	56	197	Object Description (60FF <sub>h</sub> )	69
147	Object Description (6041 <sub>h</sub> in pp Mode)	56	198	Entry Description (60FF <sub>h</sub> )	69
148	Entry Description (6041 <sub>h</sub> in pp Mode)	56	199	Supported CANopen Homing Methods	71
149	Object Description (6062 <sub>h</sub> )	56	200	Structure of the Control Word in hm Mode	74
150	Entry Description (6062 <sub>h</sub> )	57	201	Operation Mode specific Bits in hm Mode	74
151	Object Description (6063 <sub>h</sub> )	57	202	Command Coding	74
152	Entry Description (6063 <sub>h</sub> )	57	203	Object Description (6040 <sub>h</sub> in hm Mode)	75
153	Object Description (6064 <sub>h</sub> )	57	204	Entry Description (6040 <sub>h</sub> in hm Mode)	75
154	Entry Description (6064 <sub>h</sub> )	57	205	Structure of the Status Word in hm Mode	75
155	Object Description (6067 <sub>h</sub> )	58			

206	Trinamic Specific Bits . . . . .	75	251	Structure of the Status Word in csv Mode . . . . .	88
207	Operation Mode specific Bits in hm Mode . . . . .	76	252	Trinamic Specific Bits . . . . .	89
208	State Coding . . . . .	76	253	Operation Mode specific Bits in csv Mode . . . . .	89
209	Object Description (6041 <sub>h</sub> in hm Mode)	76	254	State Coding . . . . .	89
210	Entry Description (6041 <sub>h</sub> in hm Mode)	76	255	Object Description (6041 <sub>h</sub> in csv Mode)	89
211	Object Description (606C <sub>h</sub> ) . . . . .	77	256	Entry Description (6041 <sub>h</sub> in csv Mode)	90
212	Entry Description (606C <sub>h</sub> ) . . . . .	77	257	Object Description (606C <sub>h</sub> ) . . . . .	90
213	Object Description (607C <sub>h</sub> ) . . . . .	77	258	Entry Description (606C <sub>h</sub> ) . . . . .	90
214	Entry Description (607C <sub>h</sub> ) . . . . .	77	259	Object Description (60FF <sub>h</sub> ) . . . . .	90
215	Object Description (6098 <sub>h</sub> ) . . . . .	78	260	Entry Description (60FF <sub>h</sub> ) . . . . .	90
216	Entry Description (6098 <sub>h</sub> ) . . . . .	78	261	Object Description (607D <sub>h</sub> ) . . . . .	91
217	Object Description (6099 <sub>h</sub> ) . . . . .	78	262	Entry Description (607D <sub>h</sub> ) . . . . .	91
218	Entry Description (6099 <sub>h</sub> ) . . . . .	78	263	Object Description (60B1 <sub>h</sub> ) . . . . .	91
219	Object Description (609A <sub>h</sub> ) . . . . .	79	264	Entry Description (60B1 <sub>h</sub> ) . . . . .	91
220	Entry Description (609A <sub>h</sub> ) . . . . .	79	265	Object Description (60C2 <sub>h</sub> ) . . . . .	92
221	Structure of the Control Word in csp Mode . . . . .	80	266	Entry Description (60C2 <sub>h</sub> ) . . . . .	92
222	Command Coding . . . . .	81	267	Structure of the Control Word in cst Mode . . . . .	93
223	Object Description (6040 <sub>h</sub> in csp Mode)	81	268	Command Coding . . . . .	93
224	Entry Description (6040 <sub>h</sub> in csp Mode)	81	269	Object Description (6040 <sub>h</sub> in cst Mode)	94
225	Structure of the Status Word in csp Mode . . . . .	81	270	Entry Description (6040 <sub>h</sub> in cst Mode)	94
226	Trinamic Specific Bits . . . . .	82	271	Structure of the Status Word in cst Mode	94
227	Operation Mode specific Bits in csp Mode . . . . .	82	272	Trinamic Specific Bits . . . . .	94
228	State Coding . . . . .	82	273	Operation Mode specific Bits in cst Mode . . . . .	95
229	Object Description (6041 <sub>h</sub> in csp Mode)	82	274	State Coding . . . . .	95
230	Entry Description (6041 <sub>h</sub> in csp Mode)	83	275	Object Description (6041 <sub>h</sub> in cst Mode)	95
231	Object Description (6062 <sub>h</sub> ) . . . . .	83	276	Entry Description (6041 <sub>h</sub> in cst Mode)	95
232	Entry Description (6062 <sub>h</sub> ) . . . . .	83	277	Object Description (6062 <sub>h</sub> ) . . . . .	96
233	Object Description (6063 <sub>h</sub> ) . . . . .	83	278	Entry Description (6062 <sub>h</sub> ) . . . . .	96
234	Entry Description (6063 <sub>h</sub> ) . . . . .	83	279	Object Description (6063 <sub>h</sub> ) . . . . .	96
235	Object Description (6064 <sub>h</sub> ) . . . . .	84	280	Entry Description (6063 <sub>h</sub> ) . . . . .	96
236	Entry Description (6064 <sub>h</sub> ) . . . . .	84	281	Object Description (6064 <sub>h</sub> ) . . . . .	96
237	Object Description (606C <sub>h</sub> ) . . . . .	84	282	Entry Description (6064 <sub>h</sub> ) . . . . .	97
238	Entry Description (606C <sub>h</sub> ) . . . . .	84	283	Object Description (6071 <sub>h</sub> ) . . . . .	97
239	Object Description (607A <sub>h</sub> in csp Mode)	84	284	Entry Description (6071 <sub>h</sub> ) . . . . .	97
240	Entry Description (607A <sub>h</sub> in csp Mode)	85	285	Object Description (6077 <sub>h</sub> ) . . . . .	97
241	Object Description (607D <sub>h</sub> ) . . . . .	85	286	Entry Description (6077 <sub>h</sub> ) . . . . .	97
242	Entry Description (607D <sub>h</sub> ) . . . . .	85	287	Object Description (607D <sub>h</sub> ) . . . . .	98
243	Object Description (60B0 <sub>h</sub> ) . . . . .	85	288	Entry Description (607D <sub>h</sub> ) . . . . .	98
244	Entry Description (60B0 <sub>h</sub> ) . . . . .	86	289	Object Description (60B2 <sub>h</sub> ) . . . . .	98
245	Object Description (60C2 <sub>h</sub> ) . . . . .	86	290	Entry Description (60B2 <sub>h</sub> ) . . . . .	98
246	Entry Description (60C2 <sub>h</sub> ) . . . . .	86	291	Object Description (60C2 <sub>h</sub> ) . . . . .	99
247	Structure of the Control Word in csv Mode . . . . .	87	292	Entry Description (60C2 <sub>h</sub> ) . . . . .	99
248	Command Coding . . . . .	88	293	Emergency Messages (EMCY) . . . . .	101
249	Object Description (6040 <sub>h</sub> in csv Mode)	88	294	SDO Abort Codes . . . . .	103
250	Entry Description (6040 <sub>h</sub> in csv Mode)	88	295	Firmware Revision . . . . .	111
			296	Document Revision . . . . .	112

## 17 Supplemental Directives

### 17.1 Producer Information

### 17.2 Copyright

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

Redistribution of sources or derived formats (for example, Portable Document Format or Hypertext Markup Language) must retain the above copyright notice, and the complete data sheet, user manual, and documentation of this product including associated application notes; and a reference to other available product-related documentation.

### 17.3 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 CANopen® 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.

### 17.4 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.

### 17.5 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.

### 17.6 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.

## 17.7 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).

## 18 Revision History

### 18.1 Firmware Revision

Version	Date	Author	Description
V1.06	2020-03-26	ED/OK	First release.
V1.08	2021-12-17	BP	<p>Update release.</p> <ul style="list-style-type: none"> <li>• The following object entries were added: <ul style="list-style-type: none"> <li>- Within <a href="#">Object 2042<sub>h</sub>: Velocity Mode Settings</a>: <ul style="list-style-type: none"> <li>* Sensor_Selection (subindex 5)</li> <li>* Velocity_Filter (subindex 6)</li> <li>* Velocity_Unit (subindex 7)</li> <li>* Motor_Halted_Velocity (subindex 8)</li> </ul> </li> <li>- Sensor_Selection (subindex 5) within <a href="#">Object 2043<sub>h</sub>: Position Mode Settings</a></li> </ul> </li> <li>• The data type of Steps (subindex 2) within <a href="#">Object 2080<sub>h</sub>: ABN Encoder Settings</a> changed from UNSIGNED16 to UNSIGNED32</li> <li>• Default parameters were changed: <ul style="list-style-type: none"> <li>- <a href="#">Object 2050<sub>h</sub>: Motor Type</a> from 0 (no motor) to 3 (Three phase BLDC motor)</li> <li>- P_ Parameter (subindex 1) of the <a href="#">Object 2043<sub>h</sub>: Position Mode Settings</a> changed from 300 to 50</li> </ul> </li> <li>• Rework of the implementation of <a href="#">Object 1010<sub>h</sub>: Store Parameters</a> and <a href="#">Object 1011<sub>h</sub>: Restore Parameters</a>. PDO configuration can now be stored as part of the "Communication Parameters".</li> <li>• The internal scaling of the PI regulators P parameter changed.</li> <li>• The Overcurrent flag and Driver Error flag of the <a href="#">Object 2101<sub>h</sub>: Motor Status Flags</a> got implemented.</li> <li>• The error LED output handling got implemented.</li> <li>• Fix: Bad ramp at first position move after power-on reset.</li> </ul>

Table 295: Firmware Revision

### 18.2 Document Revision

Version	Date	Author	Description
V1.00	2020-03-26	OK	First release.
V1.01	2021-02-28	SK	Removed analog encoder option.
V1.02	2021-03-22	OK	Object 2702 <sub>h</sub> corrected.

Version	Date	Author	Description
V1.03	2021-12-17	BP/OK	<p>New firmware Release (V1.08). The changes to the firmware and EDS file are integrated into this document, see the Firmware Revision History.</p> <ul style="list-style-type: none"><li>• Removed object 6065<sub>h</sub> and 60F2<sub>h</sub> as they are not available in the firmware.</li><li>• Object 6084<sub>h</sub> corrected - access is "ro" instead of "rw".</li><li>• Added section SDO Abort Codes.</li><li>• Added section Firmware Update.</li></ul>

*Table 296: Document Revision*