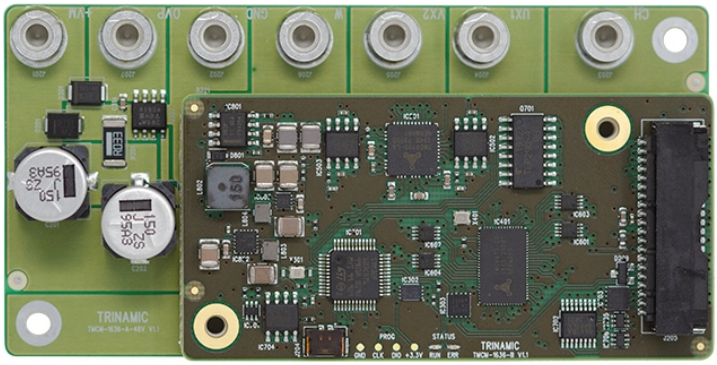
TMCM-1636 CANopen® Firmware Manual

Firmware Version V1.13 | Document Revision V1.04 • 2022-MAY-30

The TMCM-1636 is a single axis servo drive for 3-phase BLDC and DC motors with up to ca. 1000W running at +24V or +48V. It offers a CAN interface with either TMCL or CANopen protocol for communication. While in stop mode a UART interface is available to configure the module via the TMCL protocol. The TMCM-1636 supports various position feedback options: incremental quadrature encoder, digital HALL sensor, absolute SPI- and SSI-based encoder. Customization of firmware and hardware is possible.



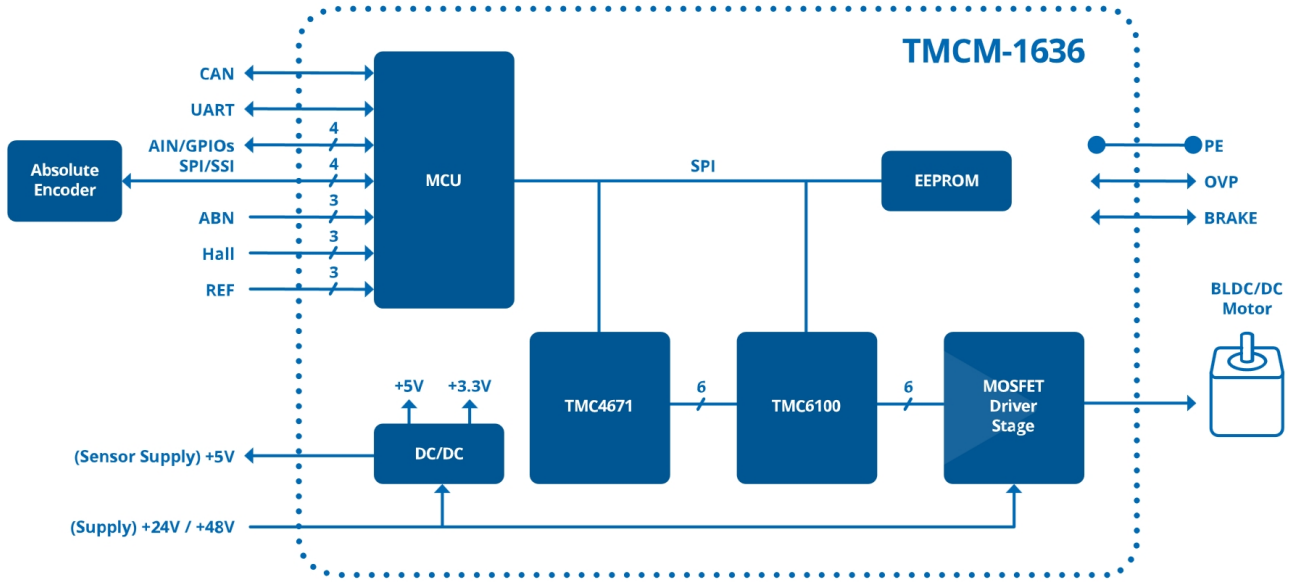
Features

- Servo Drive for BLDC and DC Motor
- +24V and +48V Supply Version
- Up to 1000W continuous
- Up to 60A RMS phase current max.
- CAN interface
- Incremental encoder
- Digital HALL sensor
- Absolute SPI & SSI-based encoder support
- Various GPIOs
- Motor brake control and overvoltage protection

Applications

- Robotics
- Laboratory Automation
- Manufacturing
- Factory Automation
- Servo Drives
- Motorized Tables and Chairs
- Industrial BLDC & DC Motor Drives

Simplified Block Diagram



ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™

ADI Trinamic™

Read entire documentation.

Contents

1 Preface **3**
1.1 General Features of this CANopen Implementation . . . 3
1.2 Abbreviations used in this Manual . . . 4
1.3 Firmware Update . . . 4

2 Communication **5**
2.1 Reference Model . . . 5
2.2 NMT State Machine . . . 7
2.3 Device Model . . . 8
2.4 Object Dictionary . . . 9

3 Communication Area **11**
3.1 Detailed Object Specifications . . . 11
3.1.1 Object 1000_h: Device Type . . . 11
3.1.2 Object 1001_h: Error Register . . . 11
3.1.3 Object 1005_h: COB-ID SYNC Message . . . 12
3.1.4 Object 1008_h: Manufacturer Device Name . . . 12
3.1.5 Object 1009_h: Manufacturer Hardware Version . . . 12
3.1.6 Object $100A_h$: Manufacturer Software Version . . . 13
3.1.7 Object $100C_h$: Guard Time . . . 13
3.1.8 Object $100D_h$: Life Time Factor . . . 13
3.1.9 Object 1010_h: Store Parameters . . . 13
3.1.10 Object 1011_h: Restore Parameters . . . 14
3.1.11 Object 1014_h: COB-ID Emergency Object . . . 16
3.1.12 Object 1015_h: Inhibit Time EMCY . . . 16
3.1.13 Object 1016_h: Consumer Heartbeat Time . . . 16
3.1.14 Object 1017_h: Producer Heartbeat Time . . . 17
3.1.15 Object 1018_h: Identity Object . . . 17
3.1.16 Object 1029_h: Error Behaviour . . . 18
3.1.17 Objects 1400_h – 1403_h: Receive PDO Communication Parameter . . . 18
3.1.18 Objects 1600_h – 1603_h: Receive PDO Mapping Parameter . . . 19
3.1.19 Objects 1800_h – 1803_h: Transmit PDO Communication Parameter . . . 20
3.1.20 Objects $1A00_h$ – $1A03_h$: Transmit PDO Mapping Parameter . . . 22

4 Manufacturer specific Area **23**
4.1 Detailed Object Specifications . . . 23
4.1.1 Object 2003_h: Maximum Current . . . 23
4.1.2 Object 2004_h: Open Loop Current . . . 23
4.1.3 Object 2005_h: Switch Parameters . . . 23
4.1.4 Object 2006_h: Brake Chopper . . . 24
4.1.5 Object 2041_h: Torque Mode Settings . . . 25
4.1.6 Object 2042_h: Velocity Mode Settings . . . 25
4.1.7 Object 2043_h: Position Mode Settings . . . 26
4.1.8 Object 2050_h: Motor Type . . . 27
4.1.9 Object 2055_h: Commutation Mode . . . 27
4.1.10 Object 2056_h: Motor Pole Pairs . . . 28
4.1.11 Object 2057_h: Motor Shaft Direction . . . 28
4.1.12 Object 2058_h: Position Scaler . . . 28
4.1.13 Object 2060_h: ADC Configuration . . . 29
4.1.14 Object 2070_h: Hall Sensor Settings . . . 29
4.1.15 Object 2080_h: ABN Encoder Settings . . . 30
4.1.16 Object 2085_h: ABS Encoder Settings . . . 31

4.1.17 Object 2101_h: Motor Status Flags . . . 32
4.1.18 Object 2102_h: Open Loop Commutation Angle . . . 32
4.1.19 Object 2103_h: Encoder Commutation Angle . . . 33
4.1.20 Object 2104_h: Hall Commutation Angle . . . 33
4.1.21 Object 2140_h: Home Offset Display . . . 33
4.1.22 Object 2702_h: Device Digital Inputs . . . 34
4.1.23 Object 2703_h: Device Digital Outputs . . . 34
4.1.24 Object 2704_h: CAN Bit Rate . . . 35
4.1.25 Object 2705_h: Node-ID . . . 35
4.1.26 Object 2707_h: CAN Bit Rate Load . . . 35
4.1.27 Object 2708_h: Node-ID Load . . . 36
4.1.28 Object $270E_h$: Analog Inputs . . . 36
4.1.29 Object $5FFF_h$: Bootloader mode . . . 37

5 Profile specific Area **38**
5.1 Detailed Object Specifications . . . 38
5.1.1 Object $605A_h$: Quick Stop Option Code . . . 38
5.1.2 Object $605B_h$: Shutdown Option Code . . . 39
5.1.3 Object $605C_h$: Disable Operation Option Code . . . 39
5.1.4 Object $605D_h$: Halt Option Code . . . 39
5.1.5 Object $605E_h$: Fault Reaction Option Code . . . 40
5.1.6 Object 6060_h: Modes of Operation . . . 40
5.1.7 Object 6061_h: Modes of Operation Display . . . 41
5.1.8 Object $60FD_h$: Digital Inputs . . . 42
5.1.9 Object 6502_h: Supported Drive Modes . . . 42
5.1.10 Object $67FF_h$: Single Device Type . . . 43

6 Profile Position Mode **43**
6.1 Detailed Object Specifications . . . 44
6.1.1 Object 6040_h: Controlword . . . 44
6.1.2 Object 6041_h: Statusword . . . 46
6.1.3 Object 6062_h: Position Demand Value . . . 47
6.1.4 Object 6063_h: Position Actual Internal Value . . . 47
6.1.5 Object 6064_h: Position Actual Value . . . 48
6.1.6 Object 6067_h: Position Window . . . 48
6.1.7 Object 6068_h: Position Window Time . . . 48
6.1.8 Object $606B_h$: Velocity Demand Value . . . 49
6.1.9 Object $606C_h$: Velocity Actual Value . . . 49
6.1.10 Object $607A_h$: Target Position . . . 49
6.1.11 Object $607D_h$: Software Position Limit . . . 50
6.1.12 Object 6081_h: Profile Velocity . . . 50
6.1.13 Object 6083_h: Profile Acceleration . . . 50
6.1.14 Object 6084_h: Profile Deceleration . . . 51
6.1.15 Object 6085_h: Quick Stop Deceleration . . . 51
6.2 How to move a Motor in pp Mode . . . 51

7 Profile Velocity Mode **53**
7.0.1 Object 6040_h: Controlword . . . 53
7.0.2 Object 6041_h: Statusword . . . 54
7.0.3 Object 6062_h: Position Demand Value . . . 56
7.0.4 Object 6063_h: Position Actual Internal Value . . . 56
7.0.5 Object 6064_h: Position Actual Value . . . 56
7.0.6 Object $606C_h$: Velocity Actual Value . . . 56
7.0.7 Object $607D_h$: Software Position Limit . . . 57

7.0.8 Object 6083_h: Profile Acceleration . . . 57
7.0.9 Object 6084_h: Profile Deceleration . . . 58
7.0.10 Object 6085_h: Quick Stop Deceleration . . . 58
7.0.11 Object $60FF_h$: Target Velocity . . . 58
7.1 How to move a Motor in pv Mode . . . 58

8 Homing Mode **60**
8.1 Homing Methods . . . 61
8.1.1 Homing Method 17: Homing on negative Limit Switch . . . 61
8.1.2 Homing Method 18: Homing on positive Limit Switch . . . 61
8.1.3 Homing Method 19: Homing on positive Home Switch . . . 62
8.1.4 Homing Method 21: Homing on negative Home Switch . . . 62
8.1.5 Homing Method 35: Current Position as Home Position . . . 63
8.2 Detailed Object Specifications . . . 64
8.2.1 Object 6040_h: Controlword . . . 64
8.2.2 Object 6041_h: Statusword . . . 65
8.2.3 Object $606C_h$: Velocity Actual Value . . . 66
8.2.4 Object $607C_h$: Home Offset . . . 66
8.2.5 Object 6098_h: Homing Method . . . 67
8.2.6 Object 6099_h: Homing Speeds . . . 67
8.2.7 Object $609A_h$: Homing Acceleration . . . 68
8.3 How to start a Homing in hm Mode . . . 68

9 Cyclic synchronous Position Mode **69**
9.1 Detailed Object Specifications . . . 69
9.1.1 Object 6040_h: Controlword . . . 69
9.1.2 Object 6041_h: Statusword . . . 70
9.1.3 Object 6062_h: Position Demand Value . . . 72
9.1.4 Object 6063_h: Position Actual Internal Value . . . 72
9.1.5 Object 6064_h: Position Actual Value . . . 72
9.1.6 Object $606C_h$: Velocity Actual Value . . . 72
9.1.7 Object $607A_h$: Target Position . . . 73
9.1.8 Object $607D_h$: Software Position Limit . . . 73
9.1.9 Object $60B0_h$: Position Offset . . . 74
9.1.10 Object $60C2_h$: Interpolation Time Period . . . 74

10 Cyclic synchronous Velocity Mode **75**
10.1 Detailed Object Specifications . . . 75
10.1.1 Object 6040_h: Controlword . . . 75
10.1.2 Object 6041_h: Statusword . . . 76
10.1.3 Object $606C_h$: Velocity Actual Value . . . 78
10.1.4 Object $60FF_h$: Target Velocity . . . 78
10.1.5 Object $607D_h$: Software Position Limit . . . 78
10.1.6 Object $60B1_h$: Velocity Offset . . . 79
10.1.7 Object $60C2_h$: Interpolation Time Period . . . 79

11 Cyclic synchronous Torque Mode **80**
11.1 Detailed Object Specifications . . . 80
11.1.1 Object 6040_h: Controlword . . . 80
11.1.2 Object 6041_h: Statusword . . . 81
11.1.3 Object 6062_h: Position Demand Value . . . 82
11.1.4 Object 6063_h: Position Actual Internal Value . . . 82
11.1.5 Object 6064_h: Position Actual Value . . . 83
11.1.6 Object 6071_h: Target Torque . . . 83
11.1.7 Object 6077_h: Torque Actual Value . . . 83

11.1.8 Object $607D_h$: Software Position Limit . . . 83
11.1.9 Object $60B2_h$: Torque Offset . . . 84
11.1.10 Object $60C2_h$: Interpolation Time Period . . . 84

12 Emergency Messages (EMCY) 86

13 SDO Abort Codes 88

14 Figures Index 90

15 Tables Index 91

16 Supplemental Directives 92
16.1 Producer Information . . . 92
16.2 Copyright . . . 92
16.3 Trademark Designations and Symbols . . . 92
16.4 Target User . . . 92
16.5 Disclaimer: Life Support Systems . . . 92
16.6 Disclaimer: Intended Use . . . 92
16.7 Collateral Documents & Tools . . . 93

17 Revision History 94
17.1 Firmware Revision . . . 94
17.2 Document Revision . . . 94

1 Preface

This document specifies objects and modes of operation of the Trinamic TMCM-1636 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-1636 TMCL firmware again through the CAN 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.

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.

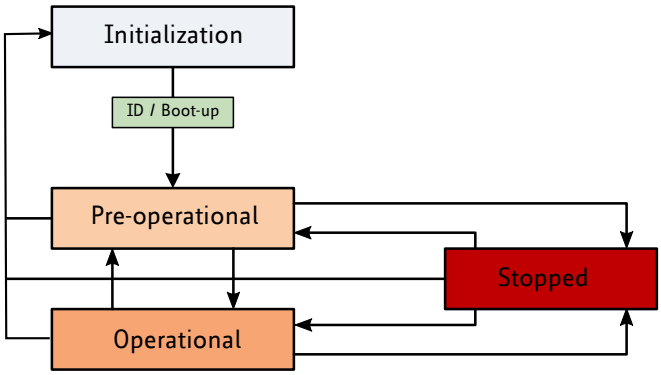


Figure 1: NMT State Machine

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.

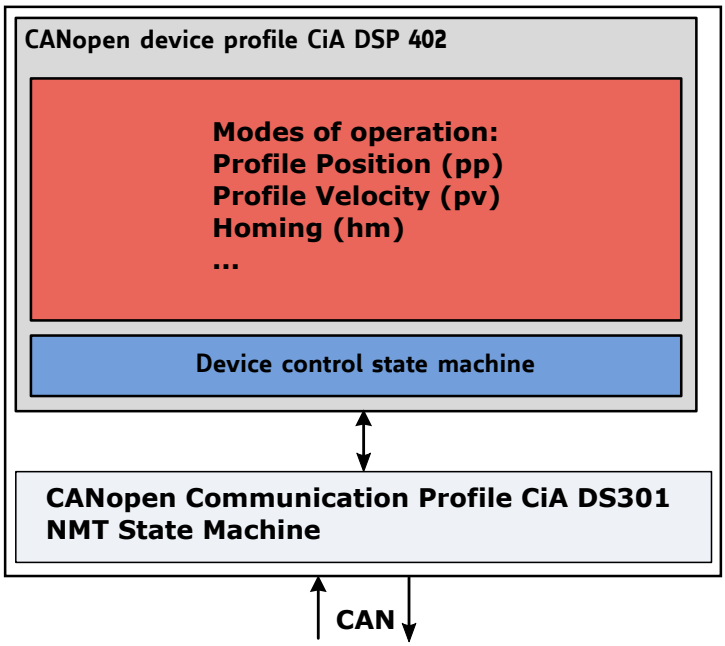


Figure 2: Communication Architecture

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.

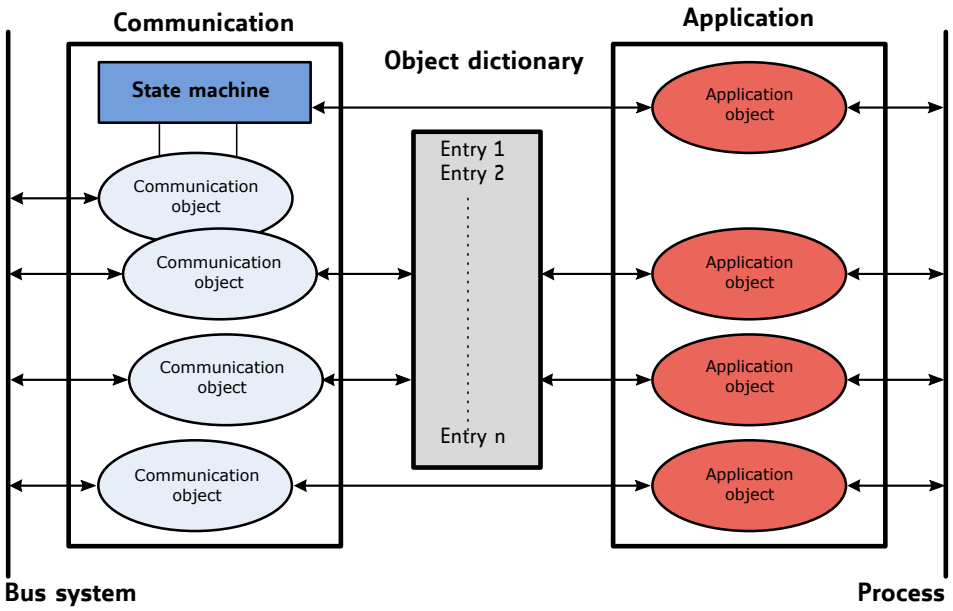


Figure 3: Device Model

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_h	Not used.
0001_h – $001F_h$	Static data types.
0020_h – $003F_h$	Complex data types.
0040_h – $005F_h$	Manufacturer specific complex data types.
0060_h – $007F_h$	Device profile specific static data types.
0080_h – $009F_h$	Device profile specific complex data types.
$00A0_h$ – $0FFF_h$	Reserved for further use.
1000_h – $1FFF_h$	Communication profile area.
2000_h – $5FFF_h$	Manufacturer specific profile area.
6000_h – $9FFF_h$	Standardized device profile area.
$A000_h$ – $BFFF_h$	Standardized interface profile area.
$C000_h$ – $FFFF_h$	Reserved for further use.

Table 4: Object Dictionary

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

The manufacturer segment at indices 2000_h through $5FFF_h$ contains manufacturer specific objects. These objects control the special features of the Trinamic TMCM-1636 motion control device.

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

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_h: Device Type

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

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
1000_h	Device Type	ro	no	UNSIGNED32	$0…2^{32}$-1	0	—

Table 5: Object Description (1000_h)

3.1.2 Object 1001_h: Error Register

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

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
1001_h	Error Register	ro	no	UNSIGNED8	$0…2^{8}$-1	—	—

Table 6: Object Description (1001_h)

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

Table 7: Error Register Bits

3.1.3 Object 1005_h: 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 8: Value Definition (1005_h)

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
1005_h	COB-ID SYNC Message	ro	no	UNSIGNED32	$0…2^{32}$-1	80_h	—

Table 9: Object Description (1005_h)

3.1.4 Object 1008_h: Manufacturer Device Name

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

Object Description					
Index	Name	Access	PDO Mapping	Data Type	Default Value
1008_h	Manufacturer Device Name	ro	no	Visible String	TMCM-1636

Table 10: Object Description (1008_h)

3.1.5 Object 1009_h: Manufacturer Hardware Version

This object contains the hardware version description.

Object Description					
Index	Name	Access	PDO Mapping	Data Type	Default Value
1009_h	Manufacturer Hardware Version	ro	no	Visible String	Depends on device, e.g. 1.00

Table 11: Object Description (1009_h)

3.1.6 Object $100A_h$: Manufacturer Software Version

This object contains the software version description.

Object Description					
Index	Name	Access	PDO Mapping	Data Type	Default Value
$100A_h$	Manufacturer Software Version	ro	no	Visible String	Depends on device, e.g. 1.00

Table 12: Object Description ($100A_h$)

3.1.7 Object $100C_h$: Guard Time

The objects at index $100C_h$ and $100D_h$ 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	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$100C_h$	Guard Time	rw	no	UNSIGNED16	$0…2^{16}$-1	0	—

Table 13: Object Description ($100C_h$)

3.1.8 Object $100D_h$: 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	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$100D_h$	Life Time Factor	rw	no	UNSIGNED8	$0…2^{8}$-1	0	—

Table 14: Object Description ($100D_h$)

3.1.9 Object 1010_h: 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-1636 module supports saving of the following parameter groups:

- Sub-index 1: save all parameters.
- Sub-index 2: save communication parameters 2704_h and 2705_h.
- Sub-index 3: save device profile parameters (not used).
- Sub-index 4: save motor 0 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_h, see also table 15).

Save Signature			
e	v	a	s
65_h	76_h	61_h	73_h

Table 15: 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_h). If a wrong signature is written, the device refuses to store and responds with abort SDO transfer (abort code: $0800002x_h$).

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_h	Store Parameters	Array	UNSIGNED32

Table 16: Object Description (1010_h)

Entry Description						
Sub-index	Name	Access	PDO Mapping	Value Range	Default Value	Unit
1	Save all parameters	rw	no	$0…2^{32}$-1	1	—
2	Save communication parameters	rw	no	$0…2^{32}$-1	1	—
3	Save device profile parameters	rw	no	$0…2^{32}$-1	0	—
4	Save motor 0 parameters	rw	no	$0…2^{32}$-1	1	—

Table 17: Entry Description (1010_h)

3.1.10 Object 1011_h: 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 TMCM-1636 module supports restoring of the following parameter groups:

- Sub-index 1: restore all parameters.

- Sub-index 2: no function.
- Sub-index 3: restore device profile parameters (not used).
- Sub-index 4: restore motor 0 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" ($64616F6C_h$, see also table 18).

Load Signature			
d	a	o	l
64_h	61_h	$6F_h$	$6C_h$

Table 18: 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_h). If a wrong signature is written, the device refuses to restore and responds with abort SDO transfer (abort code: $0800002x_h$).

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-1636.

Object Description			
Index	Name	Object Type	Data Type
1011_h	Store Parameters	Array	UNSIGNED32

Table 19: Object Description (1011_h)

Entry Description						
Sub-index	Name	Access	PDO Mapping	Value Range	Default Value	Unit
1	Restore all parameters	rw	no	$0…2^{32}$-1	—	—
2	Restore communication parameters	rw	no	$0…2^{32}$-1	—	—
3	Restore device profile parameters	rw	no	$0…2^{32}$-1	—	—
4	Restore motor 0 parameters	rw	no	$0…2^{32}$-1	—	—

Table 20: Entry Description (1011_h)

3.1.11 Object 1014_h: COB-ID Emergency Object

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

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
1014_h	COB-ID Emergency Object	rw	no	UNSIGNED32	$0\ldots2^{32}$-1	80_h + Node ID	—

Table 21: Object Description (1014_h)

3.1.12 Object 1015_h: 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	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
1015_h	Inhibit Time EMCY	rw	no	UNSIGNED16	$0\ldots2^{16}$-1	0	100µs

Table 22: Object Description (1015_h)

3.1.13 Object 1016_h: 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 23: Value Definition (1016_h)

Object Description			
Index	Name	Object Type	Data Type
1016_h	Consumer Heartbeat Time	Array	UNSIGNED32

Table 24: Object Description (1016_h)

Entry Description						
Sub-index	Name	Access	PDO Mapping	Value Range	Default Value	Unit
1	Consumer Heartbeat Time 1	rw	no	0…2^{32}-1	0	1ms

Table 25: Entry Description (1016_h)

3.1.14 Object 1017_h: 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	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
1017_h	Producer Heartbeat Time 1	rw	no	UNSIGNED16	0…2^{16}-1	0	1ms

Table 26: Object Description (1017_h)

3.1.15 Object 1018_h: Identity Object

The object 1018_h contains general information about the device:

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

Object Description			
Index	Name	Object Type	Data Type
1018_h	Identity Object	Record	Identity Object Record

Table 27: Object Description (1018_h)

Entry Description							
Sub-index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
1	Vendor ID	ro	no	UNSIGNED32	0…2^{32}-1	0286_h	—
2	Product Code	ro	no	UNSIGNED32	0…2^{32}-1	1636	—
3	Revision Number	ro	no	UNSIGNED32	0…2^{32}-1	e.g. 20003_h for version 2.3	—

Table 28: Entry Description (1018_h)

3.1.16 Object 1029_h: 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_h	Error Behaviour	Array	UNSIGNED8

Table 29: Object Description (1029_h)

Entry Description						
Sub-index	Name	Access	PDO Mapping	Value Range	Default Value	Unit
1	Communication Error	rw	no	0…2^8-1	0 (enter stopped state)	—
2	Application Error	rw	no	0…2^8-1	1 (remain in current state)	—

Table 30: Entry Description (1029_h)

3.1.17 Objects 1400_h – 1403_h: 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_h or FE_h 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 4th SYNC message). Other values are not supported.

Object Description			
Index	Name	Object Type	Data Type
1400_h – 1403_h	Receive PDO parameter	RECORD	RPDO CommPar
1400_h	RPDO 1	RECORD	RPDO CommPar
1401_h	RPDO 2	RECORD	RPDO CommPar
1402_h	RPDO 3	RECORD	RPDO CommPar
1403_h	RPDO 4	RECORD	RPDO CommPar

Table 31: Object Description (1400_h)

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_h: 200_h + Node-ID Index 1401_h: 300_h + Node-ID Index 1402_h: 400_h + Node-ID Index 1403_h: 500_h + Node-ID
2	Transmission type	rw	UNSIGNED8	Index 1400_h: FF_h Index 1401_h: FF_h Index 1402_h: FF_h Index 1403_h: FE_h

Table 32: Entry Description (1400_h)

3.1.18 Objects 1600_h – 1603_h: 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_h – 1603_h	Receive PDO mapping parameter	RECORD	PDO Mapping
1600_h	RPDO 1	RECORD	PDO Mapping
1601_h	RPDO 2	RECORD	PDO Mapping
1602_h	RPDO 3	RECORD	PDO Mapping
1603_h	RPDO 4	RECORD	PDO Mapping

Table 33: Object Description (1600_h)

Entry Description				
Sub-index	Description	Access	Value Range	Default Value
0	Number of mapped application objects in PDO	rw	0…3	Index 1600_h: 1 Index 1601_h: 2 Index 1602_h: 2 Index 1603_h: 2
1	Mapping entry 1	rw	UNSIGNED32	Index 1600_h: 60400010_h Index 1601_h: 60400010_h Index 1602_h: 60400010_h Index 1603_h: 60400010_h
2	Mapping entry 2	rw	UNSIGNED32	Index 1600_h: 0 Index 1601_h: 60600008_h Index 1602_h: $607A0020_h$ Index 1603_h: $60FF0020_h$
3	Mapping entry 3	rw	UNSIGNED32	Index 1600_h: 0_h Index 1601_h: 0_h Index 1602_h: 0_h Index 1603_h: 0_h

Table 34: Entry Description (1600_h)

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 mapppings themself 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_h – 1803_h: 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_h or FE_h 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 4th 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_h – 1803_h	Transmit PDO communication parameter	RECORD	TPDO CommPar
1800_h	TPDO 1	RECORD	TPDO CommPar
1801_h	TPDO 2	RECORD	TPDO CommPar
1802_h	TPDO 3	RECORD	TPDO CommPar
1803_h	TPDO 4	RECORD	TPDO CommPar

Table 35: Object Description (1800_h)

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_h: 180_h + Node-ID Index 1801_h: 280_h + Node-ID Index 1802_h: 380_h + Node-ID Index 1803_h: 480_h + Node-ID
2	Transmission type	rw	UNSIGNED8	Index 1800_h: FF_h Index 1801_h: FF_h Index 1802_h: 01_h Index 1803_h: 01_h
3	Inhibit time	rw	UNSIGNED16	0
4	Compatibility entry	ro	UNSIGNED8	0
5	Event timer	rw	UNSIGNED16	0

Table 36: Entry Description (1800_h)

3.1.20 Objects $1A00_h$ – $1A03_h$: 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
$1A00_h$ – $1A03_h$	Transmit PDO mapping parameter	RECORD	PDO Mapping
$1A00_h$	TPDO 1	RECORD	PDO Mapping
$1A01_h$	TPDO 2	RECORD	PDO Mapping
$1A02_h$	TPDO 3	RECORD	PDO Mapping
$1A03_h$	TPDO 4	RECORD	PDO Mapping

Table 37: Object Description ($1A00_h$)

Entry Description				
Sub-index	Description	Access	Value Range	Default Value
0	Number of mapped aapli-cation objects in PDO	rw	0…3	Index $1A00_h$: 1 Index $1A01_h$: 2 Index $1A02_h$: 2 Index $1A03_h$: 2
1	Mapping entry 1	rw	UNSIGNED32	Index $1A00_h$: 60410010_h Index $1A01_h$: 60410010_h Index $1A02_h$: 60410010_h Index $1A03_h$: 60410010_h
2	Mapping entry 2	rw	UNSIGNED32	Index $1A00_h$: 0 Index $1A01_h$: 60610008_h Index $1A02_h$: 60640020_h Index $1A03_h$: $606C0020_h$
3	Mapping entry 3	rw	UNSIGNED32	Index $1A00_h$: 0_h Index $1A01_h$: 0_h Index $1A02_h$: 0_h Index $1A03_h$: 0_h

Table 38: Entry Description ($1A00_h$)

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 mapppings themself 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.

4 Manufacturer specific Area

The manufacturer segment contains manufacturer specific objects. These objects control the special features of the Trinamic Motion Control device TMCM-1636.

4.1 Detailed Object Specifications

4.1.1 Object 2003_h: Maximum Current

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

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
2003_h	Maximum Current	rw	no	UNSIGNED32	0...60000	4000	mA

Table 39: Object Description (2003_h)

4.1.2 Object 2004_h: Open Loop Current

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

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
2004_h	Open Loop Current	rw	no	UNSIGNED32	0...60000	2000	mA

Table 40: Object Description (2004_h)

4.1.3 Object 2005_h: 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	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
2005_h	Switch Parameters	rw	no	UNSIGNED32	0…63	0	—

Table 41: Object Description (2005_h)

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

Table 42: Bit Definitions (2005_h)

4.1.4 Object 2006_h: 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_h	Brake Chopper	Record	Brake Chopper Record

Table 43: Object Description (2006_h)

Entry Description							
Sub-index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
1	Enable	rw	no	UNSIGNED8	0…1	0	—
2	Voltage	rw	no	UNSIGNED32	50…1000	1000	1/10 V
3	Hysteresis	rw	no	UNSIGNED16	0…50	0	1/10 V

Table 44: Entry Description (2006_h)

4.1.5 Object 2041_h: Torque Mode Settings

Object Description			
Index	Name	Object Type	Data Type
2041_h	Torque Mode Settings	Record	Torque Mode Settings Record

Table 45: Object Description (2041_h)

Entry Description							
Sub-index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
1	Torque_P	rw	no	UNSIGNED16	$0…2^{16}$-1	500	—
2	Torque_I	rw	no	UNSIGNED16	$0…2^{16}$-1	500	—
3	PI_Torque_Error	ro	no	SIGNED32	$-2^{31}…2^{31}$-1	—	mA
4	PI_Torque_Error_Sum	ro	no	SIGNED32	$-2^{31}…2^{31}$-1	—	—
5	PI_Flux_Error	ro	no	SIGNED32	$-2^{31}…2^{31}$-1	—	mA
6	PI_Flux_Error_Sum	ro	no	SIGNED32	$-2^{31}…2^{31}$-1	—	—
7	PHI_E	ro	no	SIGNED16	$-2^{15}…2^{15}$-1	—	—

Table 46: Entry Description (2041_h)

Torque_P P parameter for the torque PI controller.

Torque_I I parameter for the torque PI controller.

4.1.6 Object 2042_h: Velocity Mode Settings

Object Description			
Index	Name	Object Type	Data Type
2042_h	Velocity Mode Settings	Record	Velocity Mode Settings Record

Table 47: Object Description (2042_h)

Entry Description							
Sub-index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
1	Velocity_P	rw	no	UNSIGNED16	0…2^{16}-1	100	—
2	Velocity_I	rw	no	UNSIGNED16	0…2^{16}-1	100	—
3	PI_Velocity_Error	ro	no	SIGNED32	-2^{31}…2^{31}-1	—	—
4	PI_Velocity_Error_Sum	ro	no	SIGNED32	-2^{31}…2^{31}-1	—	—
5	Sensor_Selection	rw	no	UNSIGNED8	0…2	0	—
6	Velocity_Filter	rw	no	UNSIGNED8	0…8	3	—
7	Velocity_Unit	rw	no	UNSIGNED8	0…1	0	—
8	Motor_Halted_Velocity	rw	no	UNSIGNED8	0…200000	10	—

Table 48: Entry Description (2042_h)

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
- 2: Absolut 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.

4.1.7 Object 2043_h: Position Mode Settings

Object Description			
Index	Name	Object Type	Data Type
2043_h	Position Mode Settings	Record	Position Mode Settings Record

Table 49: Object Description (2043_h)

Entry Description							
Sub-index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
1	Position_P	rw	no	UNSIGNED16	0…2^{16}-1	100	—
2	PI_Position_Error	ro	no	SIGNED32	-2^{31}…2^{31}-1	—	—
3	Sensor_Selection	rw	no	UNSIGNED8	0…2	0	—

Table 50: Entry Description (2043_h)

Position_P P parameter for the velocity PI controller.

PI_Position_Error Error of the position PI controller.

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

- 0: Same as commutation
- 1: ABN encoder
- 2: Absolut encoder

4.1.8 Object 2050_h: Motor Type

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

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

Mode 2 (stepper motor) is not supported by this module.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
2050_h	Motor Type	rw	no	UNSIGNED8	0…3	3	—

Table 51: Object Description (2050_h)

4.1.9 Object 2055_h: Commutation Mode

Uisng this object the commutation mode can be set. Select a commutation mode that fits best with your motor's sensors.

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

Table 52: Commutation Modes

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
2055_h	Commutation Mode	rw	no	UNSIGNED8	0…3	0	—

Table 53: Object Description (2055_h)

4.1.10 Object 2056_h: Motor Pole Pairs

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

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
2056_h	Motor Pole Pairs	rw	no	UNSIGNED8	1…255	4	—

Table 54: Object Description (2056_h)

4.1.11 Object 2057_h: Motor Shaft Direction

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

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
2057_h	Motor Shaft Direction	rw	no	UNSIGNED8	0…1	1	—

Table 55: Object Description (2057_h)

4.1.12 Object 2058_h: 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	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
2058_h	Position Scaler	rw	no	SIGNED32	6…2147483647	65536	—

Table 56: Object Description (2058_h)

4.1.13 Object 2060_h: 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_h	ADC Configuration	Record	ADC Configuration Record

Table 57: Object Description (2060_h)

Entry Description							
Sub-index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
1	ADC_I0_Raw	ro	no	UNSIGNED16	$0…2^{16}$-1	—	—
2	ADC_I1_Raw	ro	no	UNSIGNED16	$0…2^{16}$-1	—	—
3	ADC_I0_Offset	rw	no	UNSIGNED16	$0…2^{16}$-1	33000	—
4	ADC_I1_Offset	rw	no	UNSIGNED16	$0…2^{16}$-1	33000	—
5	ADC_I0	ro	no	SIGNED16	$-2^{15}…2^{15}$-1	—	—
6	ADC_I1	ro	no	SIGNED16	$-2^{15}…2^{15}$-1	—	—
7	ADC_I2	ro	no	SIGNED16	$-2^{15}…2^{15}$-1	—	—

Table 58: Entry Description (2060_h)

4.1.14 Object 2070_h: 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_h	Hall Sensor Setting	Record	Hall Sensor Setting Record

Table 59: Object Description (2070_h)

Entry Description							
Sub-index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
1	Polarity	rw	no	UNSIGNED8	0…1	0	—
2	Direction	rw	no	UNSIGNED8	0…1	0	—
3	Interpolation	rw	no	UNSIGNED8	0…1	0	—
4	PHI_E_Offset	rw	no	SIGNED16	-2^{15}…2^{15}-1	0	—

Table 60: Entry Description (2070_h)

4.1.15 Object 2080_h: 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 61: Encoder Initialization Modes

Object Description			
Index	Name	Object Type	Data Type
2080_h	ABN Encoder Settings	Record	ABN Encoder Settings Record

Table 62: Object Description (2080_h)

Entry Description							
Sub-index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
1	Direction	rw	no	UNSIGNED8	0…1	0	—
2	Steps	rw	no	UNSIGNED32	0…2^{24}-1	4096	—
3	Init_Mode	rw	no	UNSIGNED8	0…2	0	—
4	Encoder_Value	ro	no	UNSIGNED32	-2^{31}…2^{31}-1	—	—

Table 63: Entry Description (2080_h)

4.1.16 Object 2085_h: ABS Encoder Settings

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

Encoder Types	
0	disabled
1	AMT20
2	AMT23
3	MBD01_18B
4	MBD01_19B
200	GENERIC_SSI

Table 64: Encoder Types

Encoder Initialization Modes	
0	Estimate offset
1	Use offset

Table 65: Encoder Initialization Modes

Object Description			
Index	Name	Object Type	Data Type
2085_h	ABS Encoder Settings	Record	ABS Encoder Settings Record

Table 66: Object Description (2085_h)

Entry Description							
Sub-index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
1	Type	rw	no	UNSIGNED8	0...200	0	—
2	Direction	rw	no	UNSIGNED8	0...1	0	—
3	Init_Mode	rw	no	UNSIGNED8	0...1	0	—
4	Offset	rw	no	UNSIGNED32	$0...2^{16}$-1	0	—
5	Data_Length	rw	no	UNSIGNED8	0...255	0	—
6	Position_Start	rw	no	UNSIGNED8	0...255	0	—
7	Position_Length	rw	no	UNSIGNED8	0...255	0	—

Table 67: Entry Description (2085_h)

4.1.17 Object 2101_h: Motor Status Flags

This object provides motor status and error flags. This can be a combination of the bits described in table 68.

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 68: Motor Status Flags (2101_h)

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
2101_h	Motor Status Flags	ro	no	UNSIGNED32	0…$3FFFF_h$	0	—

Table 69: Object Description (2101_h)

4.1.18 Object 2102_h: 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	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
2102_h	Open Loop Commutation Angle	ro	no	SIGNED16	$-2^{15}...2^{15}-1$	—	—

Table 70: Object Description (2102_h)

4.1.19 Object 2103_h: Encoder Commutation Angle

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

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
2103_h	Encoder Commutation Angle	ro	no	SIGNED16	$-2^{15}...2^{15}-1$	—	—

Table 71: Object Description (2103_h)

4.1.20 Object 2104_h: 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	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
2104_h	Hall Commutation Angle	ro	no	SIGNED16	$-2^{15}...2^{15}-1$	—	—

Table 72: Object Description (2104_h)

4.1.21 Object 2140_h: Home Offset Display

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

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
2140_h	Home Offset Display	ro	no	SIGNED32	$-2^{31}...2^{31}-1$	—	—

Table 73: Object Description (2140_h)

4.1.22 Object 2702_h: Device Digital Inputs

Bit Definitions	
Bit	Description
0	REF_R
1	REF_L
2	REF_H
3	GPI0
4	GPI1
5	GPI2
6	GPI3

Table 74: Bit Definitions (2702_h)

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
2702_h	Device Digital Inputs	ro	no	UNSIGNED32	0…127	—	—

Table 75: Object Description (2702_h)

4.1.23 Object 2703_h: Device Digital Outputs

With this object the digital outputs (general purpose outputs) can be set. The bits in sub index 1 (Physical outputs) switch the outputs of the module. The bits in sub index 2 (Output mask) determine which outputs can be switched.

Bit Definitions	
Bit	Description
0	GPO0
1	GPO1

Table 76: Bit Definitions (2703_h)

Object Description			
Index	Name	Object Type	Data Type
2703_h	Device Digital Outputs	Array	UNSIGNED32

Table 77: Object Description (2703_h)

Entry Description						
Sub-index	Name	Access	PDO Mapping	Value Range	Default Value	Unit
1	Physical Outputs	rw	yes	0…2^{32}-1	0	—
2	Output Mask	rw	yes	0…2^{32}-1	0	—

Table 78: Entry Description (2703_h)

4.1.24 Object 2704_h: 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 2706h. 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	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
2704_h	CAN Bit Rate	rw	no	UNSIGNED16	20…1000	1000	kBits/s

Table 79: Object Description (2704_h)

4.1.25 Object 2705_h: Node-ID

The node ID can be selected using this object. 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 Object 2705_h: Node-ID. After that, reset the module. The new setting then becomes active.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
2705_h	Node-ID	rw	no	UNSIGNED8	1…127	1	—

Table 80: Object Description (2705_h)

4.1.26 Object 2707_h: CAN Bit Rate Load

This object shows the selected CAN bit rate.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
2707_h	CAN Bit Rate Load	ro	no	UNSIGNED16	20…1000	—	kBits/s

Table 81: Object Description (2707_h)

4.1.27 Object 2708_h: Node-ID Load

This object shows the selected node ID.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
2708_h	Node-ID Load	ro	no	UNSIGNED8	1…127	—	—

Table 82: Object Description (2708_h)

4.1.28 Object $270E_h$: Analog Inputs

ADC value of the analog input pins AIN_0 to AIN_1, as well as ADC values of some on-board voltages.

Object Description			
Index	Name	Object Type	Data Type
$270E_h$	Analog Inputs	Array	UNSIGNED32

Table 83: Object Description ($270E_h$)

Entry Description						
Sub-index	Name	Access	PDO Mapping	Value Range	Default Value	Unit
1	AIN_0	ro	no	0…4095	—	—
2	AIN_1	ro	no	0…4095	—	—
3	AIN_VM	ro	no	0…65535	—	—
4	AIN_TEMP	ro	no	0…65535	—	—

Table 84: Entry Description (2703_h)

AIN_0	Analog input pin AIN_0 ADC value.
AIN_1	Analog input pin AIN_1 ADC value.
AIN_VM	ADC value of the on-board motor supply voltage measurement circuit.
AIN_TEMP	ADC value of the temperature measurement circuit.

4.1.29 Object $5FFF_h$: Bootloader mode

Writing the magic number 12345678_h to this object switches the module to bootloader mode. This is only necessary for performing a firmware update.
This object always reads as zero.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$5FFF_h$	Bootloader mode	rw	no	UNSIGNED32	$0…2^{32}$-1	0	—

Table 85: Object Description ($5FFF_h$)

5 Profile specific Area

The profile segment contains CiA-402 standard motion control objects. These objects control the motion control functions of the TMCM-1636. Since it is not possible to operate the modes in parallel, the user is able to activate the required function by selecting a mode of operation. The control device writes to the modes of operation object in order to select the operation mode. The drive device provides the modes of operation display object to indicate the actual activated operation mode. Controlword, statusword, and set-points are used mode-specific. This implies the responsibility of the control device to avoid inconsistencies and erroneous behavior.
The following operating modes (selectable via object 6060_h, please see 5.1.6) are implemented on the TMCM-1636:

- 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 $605A_h$: Quick Stop Option Code

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

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

Table 86: Value Description ($605A_h$)

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$605A_h$	Quick Stop Option Code	rw	no	SIGNED16	1, 2, 5, 6	2	—

Table 87: Object Description ($605A_h$)

5.1.2 Object $605B_h$: Shutdown Option Code

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

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

Table 88: Value Description ($605B_h$)

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$605B_h$	Shutdown Option Code	rw	no	UNSIGNED8	0	0	—

Table 89: Object Description ($605B_h$)

5.1.3 Object $605C_h$: Disable Operation Option Code

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

Value Definition	
Value	Definition
1	Slow down on slow down ramp

Table 90: Value Description ($605C_h$)

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$605C_h$	Disable Operation Option Code	rw	no	UNSIGNED8	1	1	—

Table 91: Object Description ($605C_h$)

5.1.4 Object $605D_h$: Halt Option Code

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

Value Definition	
Value	Definition
1	Slow down on slow down ramp and stay in *operation enabled*

Table 92: Value Description ($605D_h$)

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$605D_h$	Halt Option Code	rw	no	UNSIGNED8	1	1	—

Table 93: Object Description ($605D_h$)

5.1.5 Object $605E_h$: Fault Reaction Option Code

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

Value Definition	
Value	Definition
2	Slow down on quick stop ramp

Table 94: Value Description ($605E_h$)

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$605E_h$	Fault Reaction Option Code	rw	no	UNSIGNED8	2	2	—

Table 95: Object Description ($605E_h$)

5.1.6 Object 6060_h: Modes of Operation

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

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

Table 96: Value Description (6060_h)

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

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6060_h	Modes of Operation	rw	yes	SIGNED8	see table 96	0	—

Table 97: Object Description (6060_h)

5.1.7 Object 6061_h: Modes of Operation Display

This object shows the operating mode that is currently set.

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

Table 98: Value Description (6061_h)

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

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6061_h	Modes of Operation Display	ro	yes	SIGNED8	see table 98	—	—

Table 99: Object Description (6061_h)

5.1.8 Object $60FD_h$: Digital Inputs

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

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$60FD_h$	Digital Inputs	ro	yes	UNSIGNED32	$0…2^{32}$-1	—	—

Table 100: Object Description ($60FD_h$)

5.1.9 Object 6502_h: Supported Drive Modes

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

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

Table 101: Value Definition (6502_h)

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6502_h	Supported Drive Modes	ro	yes	UNSIGNED32	see table 101	-	—

Table 102: Object Description (6502_h)

5.1.10 Object $67FF_h$: Single Device Type

This object provides information on the device profile used for the individual axis. Its structure is similar to object 1000_h. The lower sixteen bits contain the device profile number which is always 402 (0192_h) with Trinamic motion control modules. The upper sixteen bits contain more information about the drive profile.

Value Definition	
Bit	Meaning
0…15	Device profile number
16	Frequency converter (0=no, 1=yes)
17	Servo drive (0=no, 1=yes)
18	Stepper motor (0=no, 1=yes)
19…21	Reserved
22	PDO set for generic drive device (0=yes, 1=no)
23	Multi device module (0=no, 1=yes)
24…31	Reserved

Table 103: Value Definition ($67FF_h$)

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$67ff_h$	Single Device Type	ro	yes	UNSIGNED32	see table 103	Depends on individual axis.	—

Table 104: Object Description ($67ff_h$)

6 Profile Position Mode

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

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

6.1 Detailed Object Specifications

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

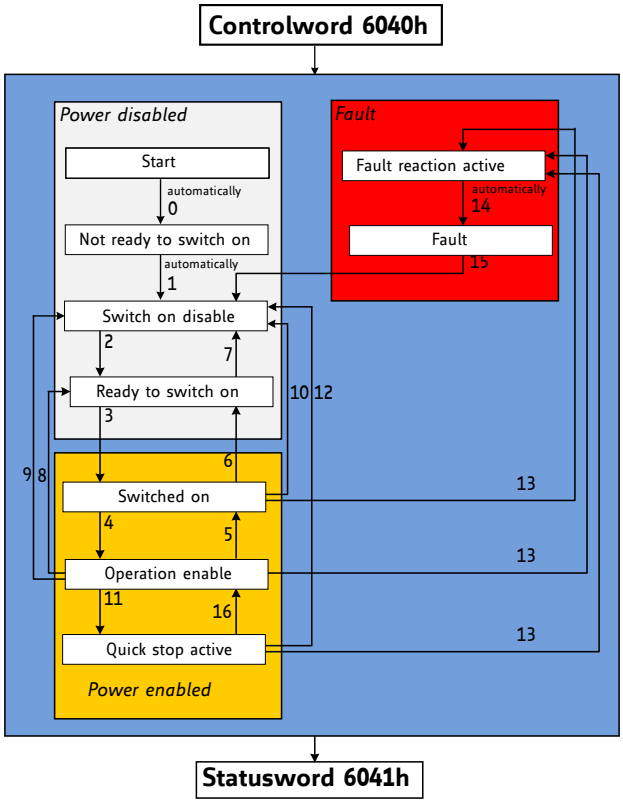


Figure 4: DS402 Finite State Machine

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_h: Controlword

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 Controlword											
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 105: Structure of the Controlword 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 106: Operation Mode specific Bits in pp Mode

Command Coding						
Command	Bits of Controlword					Transitions
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
Shutdown	0	x	1	1	0	2,6,8
Switch on	0	0	1	1	1	3
Switch on & enable operation	0	1	1	1	1	3, 4
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 107: Command Coding

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6040_h	Controlword	rw	yes	UNSIGNED16	See command coding above.	0	—

Table 108: Object Description (6040_h)

6.1.2 Object 6041_h: Statusword

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 Statusword															
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 109: Structure of the Staus Word in pp Mode

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

Table 110: 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 111: Operation Mode specific Bits in pp Mode

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

Table 112: State Coding

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6041$_h$	Statusword	ro	yes	UNSIGNED16	See state coding above.	—	—

Table 113: Object Description (6041$_h$)

6.1.3 Object 6062$_h$: Position Demand Value

This object provides the demanded position value. The value is given in PositionScaler steps (see Object 2058$_h$: Position Scaler). Object 6062$_h$ indicates the actual position that the motor should have. It is not to be confused with objects 6063$_h$ and 6064$_h$.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6062$_h$	Position Demand Value	ro	yes	SIGNED32	$-2^{31}\ldots2^{31}-1$	—	—

Table 114: Object Description (6062$_h$)

6.1.4 Object 6063$_h$: Position Actual Internal Value

This object provides the demanded position value. The value is given in PositionScaler steps (see Object 2058$_h$: Position Scaler). It is the same as object 6062$_h$.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6063_h	Position Actual Internal Value	ro	yes	SIGNED32	$-2^{31}\ldots2^{31}-1$	—	—

Table 115: Object Description (6063_h)

6.1.5 Object 6064_h: Position Actual Value

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

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6064_h	Position Actual Value	ro	yes	SIGNED32	$-2^{31}\ldots2^{31}-1$	—	—

Table 116: Object Description (6064_h)

6.1.6 Object 6067_h: Position Window

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

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6067_h	Position Window	rw	no	UNSIGNED32	$0\ldots2^{32}-1$	$FFFFFFFF_h$	—

Table 117: Object 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	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6068_h	Position Window Time	rw	no	UNSIGNED16	$0...2^{16}$-1	0	ms

Table 118: Object Description (6068_h)

6.1.8 Object $606B_h$: Velocity Demand Value

This object provides the output value of the trajectory generator.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$606B_h$	Velocity Demand Value	ro	yes	SIGNED32	$-2^{31}...2^{31}$-1	—	—

Table 119: Object Description ($606B_h$)

6.1.9 Object $606C_h$: Velocity Actual Value

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

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$606C_h$	Velocity Actual Value	ro	yes	SIGNED32	$-2^{31}...2^{31}$-1	—	—

Table 120: Object Description ($606C_h$)

6.1.10 Object $607A_h$: Target Position

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

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$607A_h$	Target Position	rw	yes	SIGNED32	$-2^{31}...2^{31}$-1	0	—

Table 121: Object Description ($607A_h$)

6.1.11 Object $607D_h$: Software Position Limit

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

$$Corrected\_min\_position\_limit = min\_position\_limit - home\_offset$$
$$Corrected\_max\_position\_limit = max\_position\_limit - home\_offset$$

Object Description			
Index	Name	Object Type	Data Type
$607D_h$	Software Position Limit	Array	SIGNED32

Table 122: Object Description ($607D_h$)

Entry Description						
Sub-index	Name	Access	PDO Mapping	Value Range	Default Value	Unit
1	Minimum Position Limit	rw	no	$-2^{31}\ldots2^{31}-1$	-2^{31}	—
2	Maximum Position Limit	rw	no	$-2^{31}\ldots2^{31}-1$	$2^{31}-1$	—

Table 123: Entry Description ($607D_h$)

6.1.12 Object 6081_h: Profile Velocity

This object indicates the configured velocity normally attained at the end of the acceleration ramp during a profiled motion and is valid for both directions of motion. The profile velocity is the maximum velocity used when driving to a new position. It is given in units of mechanical RPM or electrical RPM (depending on the setting made to 2042_h sub index 7).

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6081_h	Profile Velocity	rw	yes	UNSIGNED32	$0\ldots2^{32}-1$	0	—

Table 124: Object Description (6081_h)

6.1.13 Object 6083_h: Profile Acceleration

This object indicates the configured acceleration. Object 6083_h sets the maximum acceleration to be used in profile position and in profile velocity mode. This value is given using RPM/s units.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6083_h	Profile Acceleration	rw	yes	UNSIGNED32	$0…2^{32}$-1	0	—

Table 125: Object Description (6083_h)

6.1.14 Object 6084_h: Profile Deceleration

This object indicates the configured deceleration. On the TMCM-1636 module the deceleration ramp is always the same as the acceleration ramp. For this reason this object is a read-only object and always reads the same value as object 6083_h. This value is given in units of RPM/s.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6084_h	Profile Deceleration	ro	yes	UNSIGNED32	$0…2^{32}$-1	0	—

Table 126: Object Description (6084_h)

6.1.15 Object 6085_h: Quick Stop Deceleration

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

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6085_h	Quick Stop Deceleration	ro	yes	UNSIGNED32	$0…2^{32}$-1	0	—

Table 127: Object Description (6085_h)

6.2 How to move a Motor in pp Mode

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

- If you do not have any limit switches connected, first disable the limit switch inputs by writing 3 to object 2005_h.
- Select pp mode by writing 1 to object 6060_h.
- Write 6 to object 6040_h to switch to READY_TO_SWITCH_ON state.
- Write 7 to object 6040_h to switch to SWITCHED_ON state.

- Write 15 to object 6040_h to switch to OPERATION_ENABLED state.
- Write the desired target position (e.g. 500000) to object $607A_h$.
- Mark the new target position as active by writing 31 to object 6040_h. The motor starts moving now.
- Reset the activation by writing 15 to object 6040_h (this can be done while the motor is still moving).

7 Profile Velocity Mode

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

The profile velocity mode covers the following sub-functions:

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

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

- Profile velocity
- Profile acceleration
- Motion profile type

7.0.1 Object 6040_h: Controlword

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 Controlword does not contain any operation mode specific bits.

Structure of the Controlword											
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 128: Structure of the Controlword in pv Mode

Command Coding						
Command	Bits of Controlword					Transitions
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
Shutdown	0	x	1	1	0	2,6,8
Switch on	0	0	1	1	1	3
Switch on & enable operation	0	1	1	1	1	3, 4
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 129: Command Coding

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6040_h	Controlword	rw	yes	UNSIGNED16	See command coding above.	0	—

Table 130: Object Description (6040_h)

7.0.2 Object 6041_h: Statusword

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 Statusword															
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 131: Structure of the Statusword 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 132: 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 133: Operation Mode specific Bits in pv Mode

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

Table 134: State Coding

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6041_h	Statusword	ro	yes	UNSIGNED16	See state coding above.	—	—

Table 135: Object Description (6041_h)

7.0.3 Object 6062_h: Position Demand Value

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

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6062_h	Position Demand Value	ro	yes	SIGNED32	$-2^{31}\ldots2^{31}-1$	—	—

Table 136: Object Description (6062_h)

7.0.4 Object 6063_h: Position Actual Internal Value

This object provides the demanded position value. The value is given in PositionScaler steps (see Object 2058_h: Position Scaler). It is the same as object 6062_h.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6063_h	Position Actual Internal Value	ro	yes	SIGNED32	$-2^{31}\ldots2^{31}-1$	—	—

Table 137: Object Description (6063_h)

7.0.5 Object 6064_h: Position Actual Value

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

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6064_h	Position Actual Value	ro	yes	SIGNED32	$-2^{31}\ldots2^{31}-1$	—	—

Table 138: Object Description (6064_h)

7.0.6 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	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$606C_h$	Velocity Actual Value	ro	yes	SIGNED32	$-2^{31}\ldots2^{31}-1$	—	—

Table 139: Object Description ($606C_h$)

7.0.7 Object $607D_h$: Software Position Limit

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

$$Corrected\_min\_position\_limit = min\_position\_limit - home\_offset$$
$$Corrected\_max\_position\_limit = max\_position\_limit - home\_offset$$

Object Description			
Index	Name	Object Type	Data Type
$607D_h$	Software Position Limit	Array	SIGNED32

Table 140: Object Description ($607D_h$)

Entry Description						
Sub-index	Name	Access	PDO Mapping	Value Range	Default Value	Unit
1	Minimum Position Limit	rw	no	$-2^{31}\ldots2^{31}-1$	-2^{31}	—
2	Maximum Position Limit	rw	no	$-2^{31}\ldots2^{31}-1$	$2^{31}-1$	—

Table 141: Entry Description ($607D_h$)

7.0.8 Object 6083_h: Profile Acceleration

This object indicates the configured acceleration. Object 6083_h sets the maximum acceleration to be used in profile position and in profile velocity mode. This value is given using RPM/s units.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6083_h	Profile Acceleration	rw	yes	UNSIGNED32	$0\ldots2^{32}-1$	0	—

Table 142: Object Description (6083_h)

7.0.9 Object 6084_h: Profile Deceleration

This object indicates the configured deceleration. On the TMCM-1636 module the deceleration ramp is always the same as the acceleration ramp. For this reason this object is a read-only object and always reads the same value as object 6083_h. This value is given in units of RPM/s.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6084_h	Profile Deceleration	ro	yes	UNSIGNED32	$0 \ldots 2^{32}-1$	0	—

Table 143: Object Description (6084_h)

7.0.10 Object 6085_h: Quick Stop Deceleration

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

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6085_h	Quick Stop Deceleration	ro	yes	UNSIGNED32	$0 \ldots 2^{32}-1$	0	—

Table 144: Object Description (6085_h)

7.0.11 Object $60FF_h$: Target Velocity

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

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$60FF_h$	Target Velocity	rw	yes	SIGNED32	$-2^{31} \ldots 2^{31}-1$	0	—

Table 145: Object Description ($60FF_h$)

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_h.
- Select pv mode by writing 3 to object 6060_h.
- Write 6 to object 6040_h to switch to READY_TO_SWITCH_ON state.
- Write 7 to object 6040_h to switch to SWITCHED_ON state.
- Write 15 to object 6040_h to switch to OPERATION_ENABLED state.
- Write the desired target speed (e.g. 100000) to object $60FF_h$. The motor now accelerates to that speed.
- Stop the motor by writing 0 to object $60FF_h$.

8 Homing Mode

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

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

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

Figure 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 $607C_h$ allows displacing the zero in point the coordinate system for the home position.

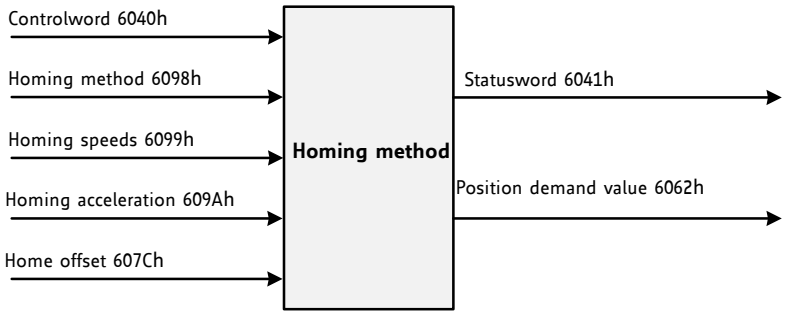


Figure 5: Homing Mode Function

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 $607C_h$, section 8.2.4).

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

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

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

8.1 Homing Methods

The TMCM-1636 supports a subset of different standard CANopen homing methods. The homing method that is to be used can be chosen via object 6098_h (section 8.2.5).

Supported Homing Methods	
Method	Description
0	No homing (default value for object 6098_h).
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 146: Supported CANopen Homing Methods

When using homing methods that need end switch inputs or home switch inputs please take care of their configuration (object 2005_h, section 4.1.3).

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.

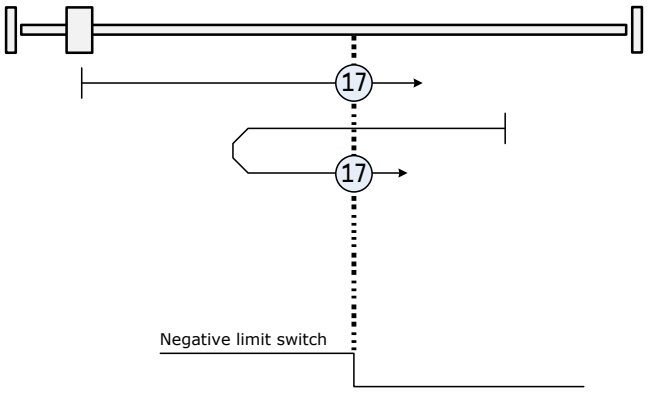


Figure 6: Homing Method 17

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.

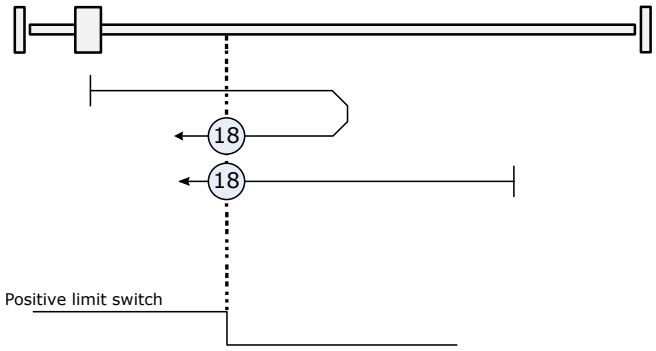


Figure 7: Homing Method 18

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.

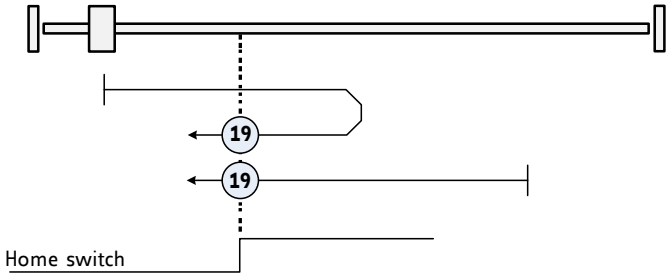


Figure 8: Homing Method 19

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.

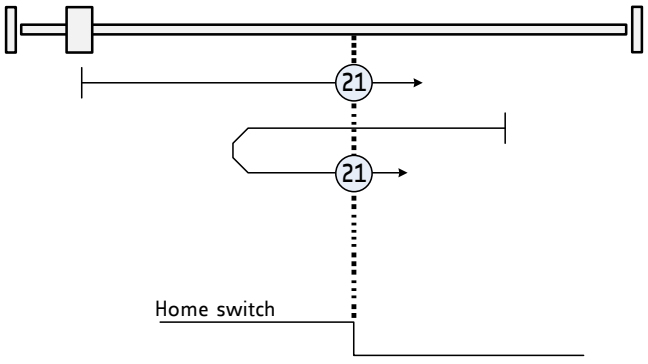


Figure 9: Homing Method 21

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_h: Controlword

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 Controlword										
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 147: Structure of the Controlword 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 148: Operation Mode specific Bits in hm Mode

Command Coding						
Command	Bits of Controlword					Transitions
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
Shutdown	0	x	1	1	0	2,6,8
Switch on	0	0	1	1	1	3
Switch on & enable operation	0	1	1	1	1	3, 4
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 149: Command Coding

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6040_h	Controlword	rw	yes	UNSIGNED16	See command coding above.	0	—

Table 150: Object Description (6040_h)

8.2.2 Object 6041_h: Statusword

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 Statusword															
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 151: Structure of the Statusword 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 152: 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 153: Operation Mode specific Bits in hm Mode

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

Table 154: State Coding

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6041$_h$	Statusword	ro	yes	UNSIGNED16	See state coding above.	—	—

Table 155: Object Description (6041$_h$)

8.2.3 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	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
606C$_h$	Velocity Actual Value	ro	yes	SIGNED32	$-2^{31}\ldots2^{31}-1$	—	—

Table 156: Object Description (606C$_h$)

8.2.4 Object 607C$_h$: Home Offset

This object indicates the configured difference between the zero position for the application and the machine home position/home switch (found during homing). While homing, the machine home position is found and once the homing is completed, the zero position is offset from the home position by adding the home offset to the home position. The effect of setting the home position to a non-zero value depends on the selected homing method. The value of this object is given in PositionScaler steps (see Object 2058$_h$: Position Scaler). Negative values indicate the opposite direction.

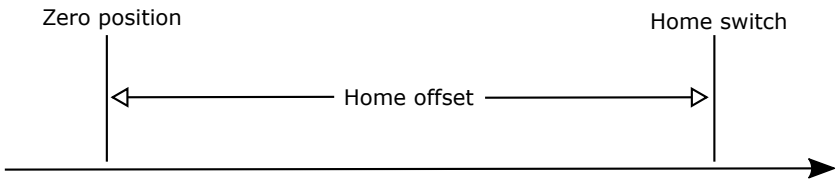


Figure 10: Home Offset

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$607C_h$	Home Offset	rw	no	SIGNED32	$-2^{31}...2^{31}-1$	0	—

Table 157: Object Description ($607C_h$)

8.2.5 Object 6098_h: Homing Method

The homing method to be used can be selected by writing to this object. Please see table 146 for a list of homing methods supported by the current version of the TMCM-1636 CANopen firmware.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6098_h	Homing Method	rw	yes	SIGNED8	See table 146	0	—

Table 158: Object Description (6098_h)

8.2.6 Object 6099_h: Homing Speeds

This object indicates the configured speeds used during homing procedure. The values are given in mechanical RPM or electrical RPM (depending on the setting made to 2042_h sub index 7) units. Using object 6099_h a fast and a slow homing speed can be set. In most homing modes, the home switch is searched with the fast speed first. When the home switch has been found, the motor will be decelerated to the slow speed (using the homing acceleration, object $609A_h$) to search for the exact switch point. When the switch point has been found the motor will be stopped at that point.

Object Description			
Index	Name	Object Type	Data Type
6099_h	Homing Speeds	Array	UNSIGNED32

Table 159: Object Description (6099_h)

Entry Description						
Sub-index	Name	Access	PDO Mapping	Value Range	Default Value	Unit
1	Fast Homing Speed 1	rw	no	0…2^{32}-1	0	—
2	Slow Homing Speed	rw	no	0…2^{32}-1	0	—

Table 160: Entry Description (6099_h)

8.2.7 Object $609A_h$: Homing Acceleration

This object indicates the configured acceleration and deceleration to be used during homing operation. This object used RPM/s units.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$609A_h$	Homing Acceleration	ro	no	UNSIGNED32	0…2^{32}-1	0	—

Table 161: Object Description ($609A_h$)

8.3 How to start a Homing in hm Mode

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

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

9 Cyclic synchronous Position Mode

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

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

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

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

9.1 Detailed Object Specifications

9.1.1 Object 6040_h: Controlword

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 Controlword.

Structure of the Controlword									
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 162: Structure of the Controlword in csp Mode

Command Coding						
Command	Bits of Controlword					Transitions
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
Shutdown	0	x	1	1	0	2,6,8
Switch on	0	0	1	1	1	3
Switch on & enable operation	0	1	1	1	1	3, 4
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 163: Command Coding

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6040_h	Controlword	rw	yes	UNSIGNED16	See command coding above.	0	—

Table 164: Object Description (6040_h)

9.1.2 Object 6041_h: Statusword

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 Statusword															
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 165: Structure of the Statusword 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 166: 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 167: Operation Mode specific Bits in csp Mode

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

Table 168: State Coding

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6041_h	Statusword	ro	yes	UNSIGNED16	See state coding above.	—	—

Table 169: Object Description (6041_h)

9.1.3 Object 6062_h: Position Demand Value

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

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6062_h	Position Demand Value	ro	yes	SIGNED32	$-2^{31}\ldots2^{31}-1$	—	—

Table 170: Object Description (6062_h)

9.1.4 Object 6063_h: Position Actual Internal Value

This object provides the demanded position value. The value is given in PositionScaler steps (see Object 2058_h: Position Scaler). It is the same as object 6062_h.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6063_h	Position Actual Internal Value	ro	yes	SIGNED32	$-2^{31}\ldots2^{31}-1$	—	—

Table 171: Object Description (6063_h)

9.1.5 Object 6064_h: Position Actual Value

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

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6064_h	Position Actual Value	ro	yes	SIGNED32	$-2^{31}\ldots2^{31}-1$	—	—

Table 172: Object Description (6064_h)

9.1.6 Object $606C_h$: Velocity Actual Value

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

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$606C_h$	Velocity Actual Value	ro	yes	SIGNED32	$-2^{31}\ldots2^{31}-1$	—	—

Table 173: Object Description ($606C_h$)

9.1.7 Object $607A_h$: Target Position

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

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$607A_h$	Target Position	rw	yes	SIGNED32	$-2^{31}\ldots2^{31}-1$	0	—

Table 174: Object Description ($607A_h$)

9.1.8 Object $607D_h$: Software Position Limit

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

$$Corrected\_min\_position\_limit = min\_position\_limit - home\_offset$$
$$Corrected\_max\_position\_limit = max\_position\_limit - home\_offset$$

Object Description			
Index	Name	Object Type	Data Type
$607D_h$	Software Position Limit	Array	SIGNED32

Table 175: Object Description ($607D_h$)

Entry Description						
Sub-index	Name	Access	PDO Mapping	Value Range	Default Value	Unit
1	Minimum Position Limit	rw	no	$-2^{31}\ldots2^{31}-1$	-2^{31}	—
2	Maximum Position Limit	rw	no	$-2^{31}\ldots2^{31}-1$	$2^{31}-1$	—

Table 176: Entry Description ($607D_h$)

9.1.9 Object $60B0_h$: Position Offset

This object provides an offset to the target position (object $607A_h$, see section 9.1.7)). The value is given in PositionScaler steps (see Object 2058_h: Position Scaler) and will be added to the target position.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$60B0_h$	Position Offset	rw	yes	SIGNED32	$-2^{31}\ldots2^{31}-1$	0	—

Table 177: Object Description ($60B0_h$)

9.1.10 Object $60C2_h$: Interpolation Time Period

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

Object Description			
Index	Name	Object Type	Data Type
$60C2_h$	Interpolation Time Period	Record	Interpolation Time Period Record (0080_h)

Table 178: Object Description ($60C2_h$)

Entry Description							
Sub-index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
1	Interpolation time period value	rw	no	UNSIGNED8	$0\ldots2^8-1$	1	—
2	Interpolation time index	rw	no	SIGNED8	-3…3	-1	—

Table 179: Entry Description (2043_h)

10 Cyclic synchronous Velocity Mode

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

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

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

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

10.1 Detailed Object Specifications

10.1.1 Object 6040_h: Controlword

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 Controlword.

Structure of the Controlword									
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 180: Structure of the Controlword in csv Mode

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

Table 181: Command Coding

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6040_h	Controlword	rw	yes	UNSIGNED16	See command coding above.	0	—

Table 182: Object Description (6040_h)

10.1.2 Object 6041_h: Statusword

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 Statusword															
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 183: Structure of the Statusword 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 184: 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 185: Operation Mode specific Bits in csv Mode

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

Table 186: State Coding

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6041_h	Statusword	ro	yes	UNSIGNED16	See state coding above.	—	—

Table 187: Object Description (6041_h)

10.1.3 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	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$606C_h$	Velocity Actual Value	ro	yes	SIGNED32	$-2^{31}\ldots2^{31}-1$	—	—

Table 188: Object Description ($606C_h$)

10.1.4 Object $60FF_h$: Target Velocity

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

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$60FF_h$	Target Velocity	rw	yes	SIGNED32	$-2^{31}\ldots2^{31}-1$	0	—

Table 189: Object Description ($60FF_h$)

10.1.5 Object $607D_h$: Software Position Limit

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

$$Corrected\_min\_position\_limit = min\_position\_limit - home\_offset$$
$$Corrected\_max\_position\_limit = max\_position\_limit - home\_offset$$

Object Description			
Index	Name	Object Type	Data Type
$607D_h$	Software Position Limit	Array	SIGNED32

Table 190: Object Description ($607D_h$)

Entry Description						
Sub-index	Name	Access	PDO Mapping	Value Range	Default Value	Unit
1	Minimum Position Limit	rw	no	$-2^{31}\ldots2^{31}-1$	-2^{31}	—
2	Maximum Position Limit	rw	no	$-2^{31}\ldots2^{31}-1$	$2^{31}-1$	—

Table 191: Entry Description ($607D_h$)

10.1.6 Object $60B1_h$: Velocity Offset

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

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$60B1_h$	Target Offset	rw	yes	SIGNED32	$-2^{31}\ldots2^{31}-1$	0	—

Table 192: Object Description ($60B1_h$)

10.1.7 Object $60C2_h$: Interpolation Time Period

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

Object Description			
Index	Name	Object Type	Data Type
$60C2_h$	Interpolation Time Period	Record	Interpolation Time Period Record (0080_h)

Table 193: Object Description ($60C2_h$)

Entry Description							
Sub-index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
1	Interpolation time period value	rw	no	UNSIGNED8	$0\ldots2^{8}-1$	1	—
2	Interpolation time index	rw	no	SIGNED8	-3…3	-1	—

Table 194: Entry Description (2043_h)

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_h: Controlword

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 Controlword.

Structure of the Controlword									
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 195: Structure of the Controlword in cst Mode

Command Coding						
Command	Bits of Controlword					Transitions
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
Shutdown	0	x	1	1	0	2,6,8
Switch on	0	0	1	1	1	3
Switch on & enable operation	0	1	1	1	1	3, 4
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 196: Command Coding

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6040_h	Controlword	rw	yes	UNSIGNED16	See command coding above.	0	—

Table 197: Object Description (6040_h)

11.1.2 Object 6041_h: Statusword

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 Statusword															
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 198: Structure of the Statusword 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 199: 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 200: Operation Mode specific Bits in cst Mode

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

Table 201: State Coding

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6041$_h$	Statusword	ro	yes	UNSIGNED16	See state coding above.	—	—

Table 202: Object Description (6041$_h$)

11.1.3 Object 6062$_h$: Position Demand Value

This object provides the demanded position value. The value is given in PositionScaler steps (see Object 2058$_h$: Position Scaler). Object 6062$_h$ indicates the actual position that the motor should have. It is not to be confused with objects 6063$_h$ and 6064$_h$.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6062$_h$	Position Demand Value	ro	yes	SIGNED32	$-2^{31}\ldots2^{31}-1$	—	—

Table 203: Object Description (6062$_h$)

11.1.4 Object 6063$_h$: Position Actual Internal Value

This object provides the demanded position value. The value is given in PositionScaler steps (see Object 2058$_h$: Position Scaler). It is the same as object 6062$_h$.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6063_h	Position Actual Internal Value	ro	yes	SIGNED32	$-2^{31}...2^{31}-1$	—	—

Table 204: Object Description (6063_h)

11.1.5 Object 6064_h: Position Actual Value

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

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6064_h	Position Actual Value	ro	yes	SIGNED32	$-2^{31}...2^{31}-1$	—	—

Table 205: Object Description (6064_h)

11.1.6 Object 6071_h: Target Torque

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

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6071_h	Target Torque	rw	yes	SIGNED32	-60000...60000	0	—

Table 206: Object Description (6071_h)

11.1.7 Object 6077_h: Torque Actual Value

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

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6077_h	Torque Actual Value	rw	yes	SIGNED32	-60000...60000	—	mA

Table 207: Object Description (6077_h)

11.1.8 Object $607D_h$: Software Position Limit

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

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

$$Corrected\_min\_position\_limit = min\_position\_limit - home\_offset$$
$$Corrected\_max\_position\_limit = max\_position\_limit - home\_offset$$

Object Description			
Index	Name	Object Type	Data Type
$607D_h$	Software Position Limit	Array	SIGNED32

Table 208: Object Description ($607D_h$)

Entry Description						
Sub-index	Name	Access	PDO Mapping	Value Range	Default Value	Unit
1	Minimum Position Limit	rw	no	$-2^{31}\ldots2^{31}-1$	-2^{31}	—
2	Maximum Position Limit	rw	no	$-2^{31}\ldots2^{31}-1$	$2^{31}-1$	—

Table 209: Entry Description ($607D_h$)

11.1.9 Object $60B2_h$: Torque Offset

This object provides an offset to the torque value. It will be added to the target torque (object 6071_h, see section 11.1.6).

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$60B2_h$	Torque Offset	rw	yes	SIGNED32	-60000…60000	—	mA

Table 210: Object Description ($60B2_h$)

11.1.10 Object $60C2_h$: Interpolation Time Period

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

Object Description			
Index	Name	Object Type	Data Type
$60C2_h$	Interpolation Time Period	Record	Interpolation Time Period Record (0080_h)

Table 211: Object Description ($60C2_h$)

Entry Description							
Sub-index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
1	Interpolation time period value	rw	no	UNSIGNED8	0…2^8-1	1	—
2	Interpolation time index	rw	no	SIGNED8	-3…3	-1	—

Table 212: Entry Description (2043_h)

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_h (error register) is part of every emergency object.

Emergency Messages (EMCY) of the TMCM-1636						
Error code	Additional byte					Description
	1	2	3	4	5	
0000_h	0	0	0	0	0	**Fault reset** The fault reset command has been executed.
4310_h	2	0	0	0	0	**Overtemperature error** The motor driver has been switched off because the temperature limit has been exceeded.
5441_h	0	255	0	0	0	**Shutdown switch active** The enable signal is missing (due to the shutdown switch) and the motor driver has been switched off.
6320_h	0	255	0	0	0	**Parameter error** The data in the received PDO is either wrong or cannot be accepted due to the internal state of the drive.
8100_h	0	255	0	0	0	**Communication error** General CAN bus communication error.
8110_h	1	255	0	0	0	**CAN controller overflow** The receive message buffer of the CAN controller hardware is full and some CAN messages are lost.
8110_h	2	255	0	0	0	**CAN Tx buffer overflow** The software CAN transmit buffer is full and thus some CAN messages are lost.
8110_h	3	255	0	0	0	**CAN Rx buffer overflow** The software CAN receive buffer is full and so some CAN messages are lost.
8120_h	0	255	0	0	0	**CAN error passive** The CAN controller has detected communication errors and has entered the CAN Error passive state.
8130_h	0	255	0	0	0	**Heartbeat or lifeguard error** The module did not receive a heartbeat or lifeguard message in time.
8140_h	0	255	0	0	0	**CAN controller recovered from bus-off state** 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_h	0	255	0	0	0	**PDO not processed due to length error** 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_h	0	255	0	0	0	**PDO length exceeded** A PDO sent to the module could not be processed because too many bytes were supplied.
8611_h	0	0	0	0	0	**Following error** The deviation between motor position counter and encoder position counter has exceeded the following error window.
$ff00_h$	0	0	0	0	0	**Undervoltage** The supply voltage is too low to drive a motor.
$ff01_h$	1	0	0	0	0	**Positive software limit** The actual position is outside the range defined by object $607d_h$.
$ff01_h$	2	0	0	0	0	**Negative software limit** The actual position is outside the range defined by object $607d_h$.
$ff01_h$	3	0	0	0	0	**Positive limit switch** The positive limit switch has been touched outside of the homing function.
$ff01_h$	4	0	0	0	0	**Negative limit switch** The negative limit switch has been touched outside of the homing function.

Table 213: 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-1636 module.

SDO Abort Codes	
Abort code	Description
05030000_h	Toggle bit not alternated.
05040000_h	SDO protocol timed out.
05040001_h	Client/server command specifier not valid or unknown.
05040002_h	Invalid block size.
05040003_h	Invalid sequence number.
05040004_h	CRC error.
05040005_h	Out of memory.
06010000_h	Unsupported access to an object.
06010001_h	Attempt to read a write only object.
06010002_h	Attempt to write a read only object.
06020000_h	Object does not exist in object dictionary.
06040041_h	Object cannot be mapped to the PDO.
06040042_h	The number and length of the objects to be mapped would exceed the PDO length.
06040043_h	General parameter incompatibility reason.
06040047_h	General internal incompatibility in the device.
06060000_h	Access failed due to a hardware error.
06070010_h	Data type does not match, length of service parameter does not match.
06070012_h	Data type does not match, length of service parameter too high.
06070013_h	Data type does not match, length of service parameter too low.
06090011_h	Sub-index does not exist.
06090030_h	Invalid value for parameter.
06090031_h	Value of parameter too high.
06090032_h	Value of parameter too low.
06090036_h	Maximum value is less than minimum value.
$060A0023_h$	Resource not available.
08000000_h	General error.
08000020_h	Data cannot be transferred or stored to the application.
08000021_h	Data cannot be transferred or stored to the application because of local control.

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

Table 214: SDO Abort Codes

14 Figures Index

1 NMT State Machine 7
2 Communication Architecture 8
3 Device Model 9
4 DS402 Finite State Machine 44
5 Homing Mode Function 60
6 Homing Method 17 61
7 Homing Method 18 62
8 Homing Method 19 62
9 Homing Method 21 62
10 Home Offset 67

15 Tables Index

1 Abbreviations used in this Manual . . 4
2 Service Primitives 5
3 Service Types 6
4 Object Dictionary 9
5 Object Description (1000_h) 11
6 Object Description (1001_h) 11
7 Error Register Bits 11
8 Value Definition (1005_h) 12
9 Object Description (1005_h) 12
10 Object Description (1008_h) 12
11 Object Description (1009_h) 12
12 Object Description ($100A_h$) 13
13 Object Description ($100C_h$) 13
14 Object Description ($100D_h$) 13
15 Save Signature 14
16 Object Description (1010_h) 14
17 Entry Description (1010_h) 14
18 Load Signature 15
19 Object Description (1011_h) 15
20 Entry Description (1011_h) 15
21 Object Description (1014_h) 16
22 Object Description (1015_h) 16
23 Value Definition (1016_h) 16
24 Object Description (1016_h) 16
25 Entry Description (1016_h) 17
26 Object Description (1017_h) 17
27 Object Description (1018_h) 17
28 Entry Description (1018_h) 18
29 Object Description (1029_h) 18
30 Entry Description (1029_h) 18
31 Object Description (1400_h) 19
32 Entry Description (1400_h) 19
33 Object Description (1600_h) 20
34 Entry Description (1600_h) 20
35 Object Description (1800_h) 21
36 Entry Description (1800_h) 21
37 Object Description ($1A00_h$) 22
38 Entry Description ($1A00_h$) 22
39 Object Description (2003_h) 23
40 Object Description (2004_h) 23
41 Object Description (2005_h) 24
42 Bit Definitions (2005_h) 24
43 Object Description (2006_h) 24
44 Entry Description (2006_h) 24
45 Object Description (2041_h) 25
46 Entry Description (2041_h) 25
47 Object Description (2042_h) 25
48 Entry Description (2042_h) 26
49 Object Description (2043_h) 26
50 Entry Description (2043_h) 27
51 Object Description (2050_h) 27
52 Commutation Modes 28
53 Object Description (2055_h) 28
54 Object Description (2056_h) 28
55 Object Description (2057_h) 28
56 Object Description (2058_h) 29
57 Object Description (2060_h) 29
58 Entry Description (2060_h) 29
59 Object Description (2070_h) 29
60 Entry Description (2070_h) 30
61 Encoder Initialization Modes 30
62 Object Description (2080_h) 30
63 Entry Description (2080_h) 30
64 Encoder Types 31
65 Encoder Initialization Modes 31
66 Object Description (2085_h) 31
67 Entry Description (2085_h) 31
68 Motor Status Flags (2101_h) 32
69 Object Description (2101_h) 32
70 Object Description (2102_h) 33
71 Object Description (2103_h) 33
72 Object Description (2104_h) 33
73 Object Description (2140_h) 33
74 Bit Definitions (2702_h) 34
75 Object Description (2702_h) 34
76 Bit Definitions (2703_h) 34
77 Object Description (2703_h) 34
78 Entry Description (2703_h) 35
79 Object Description (2704_h) 35
80 Object Description (2705_h) 35
81 Object Description (2707_h) 36
82 Object Description (2708_h) 36
83 Object Description ($270E_h$) 36
84 Entry Description (2703_h) 36
85 Object Description ($5FFF_h$) 37
86 Value Description ($605A_h$) 38
87 Object Description ($605A_h$) 38
88 Value Description ($605B_h$) 39
89 Object Description ($605B_h$) 39
90 Value Description ($605C_h$) 39
91 Object Description ($605C_h$) 39
92 Value Description ($605D_h$) 40
93 Object Description ($605D_h$) 40
94 Value Description ($605E_h$) 40
95 Object Description ($605E_h$) 40
96 Value Description (6060_h) 41
97 Object Description (6060_h) 41
98 Value Description (6061_h) 41
99 Object Description (6061_h) 42
100 Object Description ($60FD_h$) 42
101 Value Definition (6502_h) 42
102 Object Description (6502_h) 43
103 Value Definition ($67FF_h$) 43
104 Object Description ($67ff_h$) 43

105 Structure of the Controlword in pp Mode . . . 45
106 Operation Mode specific Bits in pp Mode . . . 45
107 Command Coding . . . 45
108 Object Description (6040_h) . . . 46
109 Structure of the Staus Word in pp Mode 46
110 Trinamic Specific Bits . . . 46
111 Operation Mode specific Bits in pp Mode . . . 46
112 State Coding . . . 47
113 Object Description (6041_h) . . . 47
114 Object Description (6062_h) . . . 47
115 Object Description (6063_h) . . . 48
116 Object Description (6064_h) . . . 48
117 Object Description (6067_h) . . . 48
118 Object Description (6068_h) . . . 49
119 Object Description ($606B_h$) . . . 49
120 Object Description ($606C_h$) . . . 49
121 Object Description ($607A_h$) . . . 49
122 Object Description ($607D_h$) . . . 50
123 Entry Description ($607D_h$) . . . 50
124 Object Description (6081_h) . . . 50
125 Object Description (6083_h) . . . 51
126 Object Description (6084_h) . . . 51
127 Object Description (6085_h) . . . 51
128 Structure of the Controlword in pv Mode . . . 53
129 Command Coding . . . 54
130 Object Description (6040_h) . . . 54
131 Structure of the Statusword in pv Mode 54
132 Trinamic Specific Bits . . . 55
133 Operation Mode specific Bits in pv Mode 55
134 State Coding . . . 55
135 Object Description (6041_h) . . . 55
136 Object Description (6062_h) . . . 56
137 Object Description (6063_h) . . . 56
138 Object Description (6064_h) . . . 56
139 Object Description ($606C_h$) . . . 57
140 Object Description ($607D_h$) . . . 57
141 Entry Description ($607D_h$) . . . 57
142 Object Description (6083_h) . . . 57
143 Object Description (6084_h) . . . 58
144 Object Description (6085_h) . . . 58
145 Object Description ($60FF_h$) . . . 58
146 Supported CANopen Homing Methods 61
147 Structure of the Controlword in hm Mode . . . 64
148 Operation Mode specific Bits in hm Mode . . . 64
149 Command Coding . . . 64
150 Object Description (6040_h) . . . 65
151 Structure of the Statusword in hm Mode 65
152 Trinamic Specific Bits . . . 65
153 Operation Mode specific Bits in hm Mode . . . 65
154 State Coding . . . 66
155 Object Description (6041_h) . . . 66
156 Object Description ($606C_h$) . . . 66
157 Object Description ($607C_h$) . . . 67
158 Object Description (6098_h) . . . 67
159 Object Description (6099_h) . . . 67
160 Entry Description (6099_h) . . . 68
161 Object Description ($609A_h$) . . . 68
162 Structure of the Controlword in csp Mode . . . 69
163 Command Coding . . . 70
164 Object Description (6040_h) . . . 70
165 Structure of the Statusword in csp Mode 70
166 Trinamic Specific Bits . . . 71
167 Operation Mode specific Bits in csp Mode . . . 71
168 State Coding . . . 71
169 Object Description (6041_h) . . . 71
170 Object Description (6062_h) . . . 72
171 Object Description (6063_h) . . . 72
172 Object Description (6064_h) . . . 72
173 Object Description ($606C_h$) . . . 73
174 Object Description ($607A_h$) . . . 73
175 Object Description ($607D_h$) . . . 73
176 Entry Description ($607D_h$) . . . 73
177 Object Description ($60B0_h$) . . . 74
178 Object Description ($60C2_h$) . . . 74
179 Entry Description (2043_h) . . . 74
180 Structure of the Controlword in csv Mode . . . 75
181 Command Coding . . . 76
182 Object Description (6040_h) . . . 76
183 Structure of the Statusword in csv Mode 76
184 Trinamic Specific Bits . . . 77
185 Operation Mode specific Bits in csv Mode . . . 77
186 State Coding . . . 77
187 Object Description (6041_h) . . . 77
188 Object Description ($606C_h$) . . . 78
189 Object Description ($60FF_h$) . . . 78
190 Object Description ($607D_h$) . . . 78
191 Entry Description ($607D_h$) . . . 79
192 Object Description ($60B1_h$) . . . 79
193 Object Description ($60C2_h$) . . . 79
194 Entry Description (2043_h) . . . 79
195 Structure of the Controlword in cst Mode . . . 80
196 Command Coding . . . 80
197 Object Description (6040_h) . . . 81
198 Structure of the Statusword in cst Mode 81
199 Trinamic Specific Bits . . . 81

200 Operation Mode specific Bits in cst Mode 81
201 State Coding 82
202 Object Description (6041_h) 82
203 Object Description (6062_h) 82
204 Object Description (6063_h) 83
205 Object Description (6064_h) 83
206 Object Description (6071_h) 83
207 Object Description (6077_h) 83
208 Object Description ($607D_h$) 84
209 Entry Description ($607D_h$) 84
210 Object Description ($60B2_h$) 84
211 Object Description ($60C2_h$) 84
212 Entry Description (2043_h) 85
213 Emergency Messages (EMCY) 87
214 SDO Abort Codes 89
215 Firmware Revision 94
216 Document Revision 95

16 Supplemental Directives

16.1 Producer Information

16.2 Copyright

16.3 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.

16.4 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.

16.5 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.

16.6 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.

16.7 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.

17 Revision History

17.1 Firmware Revision

Version	Date	Author	Description
1.05	2020-FEB-06	SW/OK	First release.
1.09	2020-MAY-18	ED/OK	Launch release.
1.12	2020-SEP-24	ED	Bugfix for UART input. Added object 0x2085, updated object 0x2703, removed objects 0x2707 and 0x2708.
1.13	2022-MAY-30	BP	Bugfix and update release. • Add missing limit checks for some objects. • The data type of the following objects changed from SIGNED16 to SIGNED32: – Object 6071_h: Target Torque – Object 6077_h: Torque Actual Value – Object $60B2_h$: Torque Offset • Rework of the implementation of Object 1010_h: Store Parameters and Object 1011_h: Restore Parameters. PDO configuration can now be stored as part of the “Communication Parameters”. • The Overcurrent flag and Driver Error flag of the Object 2101_h: Motor Status Flags got implemented. • Fix: Bad ramp at first position move after power-on reset. • Fix: In PDO the “motor rotates” status flag does not go active if the motor turns. • Fix: twist of REF_R and REF_L polarity inversion in Object 2005_h: Switch Parameters. • Activates space vector PWM. • Removed objects: – 2000_h Device Info, – 6072_h Max Torque, – 6074_h Torque Demand Value.

Table 215: Firmware Revision

17.2 Document Revision

Version	Date	Author	Description
1.00	2020-FEB-06	TMC	First release.

Version	Date	Author	Description
1.01	2020-JUN-07	TMC	Features of firmware V1.09 added.
1.02	2020-SEP-28	TMC	Added object 0x2085, updated object 0x2703, removed objects 0x2707 and 0x2708.
1.03	2021-MAR-08	TMC	Removed analog encoder option.
1.04	2022-MAY-30	BP	New firmware release (1.13). The changes to the firmware and EDS file were integrated into this document, see the Firmware Revision History. Apart from that the following things were revised in the manual. • Removed object 6065_h and $60F2_h$ as they are not available in the firmware. • Object Object 6084_h: Profile Deceleration corrected - access is “ro” instead of “rw”. • Added object Object $67FF_h$: Single Device Type. • Updated Object 2041_h: Torque Mode Settings, Object 2042_h: Velocity Mode Settings and Object 2043_h: Position Mode Settings. • Fixed microstep, pps and pps^2 unit relics. • Added section SDO Abort Codes. • Reworked the object tables. • Fixed the name of object 6040_h from “Control Word” to “Controlword”, to align with the EDS file. • Fixed the name of object 6041_h from “Status Word” to “Statusword”, to align with the EDS file. Removed 2x incremental encoder support - only one is supported. Changed the front page product text - the UART interface should only be used as secondary configuration interface.

Table 216: Document Revision