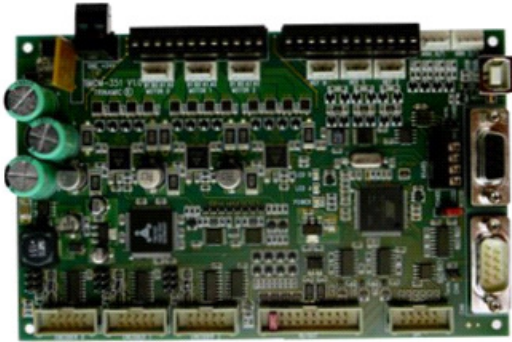


CANopen Firmware Version V3.20

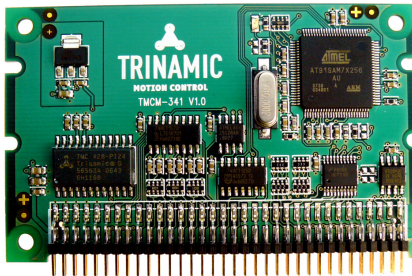
CANopen MANUAL



CANopen

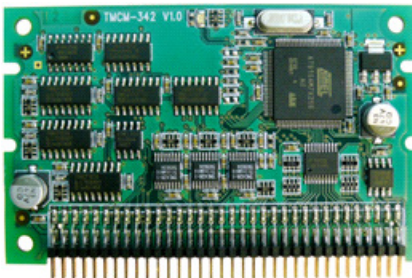
TMCM-351

3-axis stepper
controller/driver board
2.8A RMS / 24V DC
Encoder interface



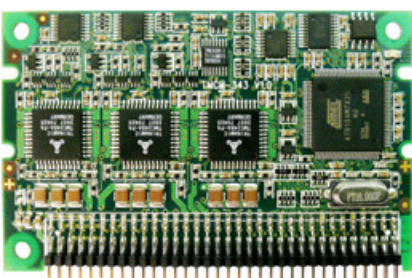
TMCM-341

3-axis
controller board
for SPI drivers
 V_{logic} 5V DC



TMCM-342

3-axis
controller board
for step/dir drivers
 V_{logic} 5V DC



TMCM-343

3-axis stepper
controller/driver board
 V_{nom} 8...34 V DC / V_{logic} 5V DC
1.1A RMS

TRINAMIC Motion Control GmbH & Co. KG
Hamburg, Germany

www.trinamic.com



TRINAMIC

MOTION CONTROL

Table of Contents

| | | |
|--------|--|----|
| 1 | Preface | 6 |
| 1.1 | General Features of the CANopen Implementation | 6 |
| 1.2 | Abbreviations | 7 |
| 1.3 | Firmware Update | 7 |
| 2 | Communication | 8 |
| 2.1 | Reference Model | 8 |
| 2.2 | NMT State Machine | 9 |
| 2.3 | Device Model | 11 |
| 2.4 | Object Dictionary | 12 |
| 3 | Communication Objects | 13 |
| 3.1 | Detailed Object Specifications | 13 |
| 3.1.1 | Object 1000 _h : Device Type | 13 |
| 3.1.2 | Object 1001 _h : Error Register | 13 |
| 3.1.3 | Object 1005 _h : COB-ID SYNC Message | 14 |
| 3.1.4 | Object 1008 _h : Manufacturer Device Name | 14 |
| 3.1.5 | Object 1009 _h : Manufacturer Hardware Version | 14 |
| 3.1.6 | Object 100A _h : Manufacturer Software Version | 14 |
| 3.1.7 | Object 100C _h : Guard Time | 15 |
| 3.1.8 | Object 100D _h : Life Time Factor | 15 |
| 3.1.9 | Object 1010 _h : Store Parameters | 15 |
| 3.1.10 | Object 1011 _h : Restore Default Parameters | 16 |
| 3.1.11 | Object 1014 _h : COB-ID Emergency Object | 18 |
| 3.1.12 | Object 1015 _h : Inhibit Time EMCY | 18 |
| 3.1.13 | Object 1016 _h : Consumer Heartbeat Time | 18 |
| 3.1.14 | Object 1017 _h : Producer Heartbeat Time | 18 |
| 3.1.15 | Object 1018 _h : Identity Object | 19 |
| 3.1.16 | Object 1023 _h : OS Command | 19 |
| 3.1.17 | Object 1029 _h : Error Behavior | 19 |
| 3.1.18 | Objects 1400 _h -1405 _h : Receive PDO Communication Parameter | 20 |
| 3.1.19 | Objects 1440 _h -1445 _h : Receive PDO Communication Parameter | 20 |
| 3.1.20 | Objects 1480 _h -1485 _h : Receive PDO Communication Parameter | 21 |
| 3.1.21 | Objects 1600 _h -1605 _h : Receive PDO Mapping | 22 |
| 3.1.22 | Objects 1640 _h -1645 _h : Receive PDO Mapping | 23 |
| 3.1.23 | Objects 1680 _h -1685 _h : Receive PDO Mapping | 24 |
| 3.1.24 | Objects 1800 _h -1805 _h : Transmit PDO Communication Parameter | 25 |
| 3.1.25 | Objects 1840 _h -1845 _h : Transmit PDO Communication Parameter | 26 |
| 3.1.26 | Objects 1880 _h -1885 _h : Transmit PDO Communication Parameter | 27 |
| 3.1.27 | Objects 1A00 _h -1A05 _h : Transmit PDO Mapping Parameter | 28 |
| 3.1.28 | Objects 1A40 _h -1A45 _h : Transmit PDO Mapping Parameter | 29 |
| 3.1.29 | Objects 1A80 _h -1A85 _h : Transmit PDO Mapping Parameter | 30 |
| 4 | Device Profile Objects (CiA402) and Modes of Operation | 31 |
| 4.1 | Detailed Object Specifications | 31 |
| 4.1.1 | Object 605A _h , 685A _h , and 705A _h : Quick Stop Option Code | 31 |
| 4.1.2 | Object 605B _h , 685B _h , and 705B _h : Shutdown Option Code | 32 |
| 4.1.3 | Object 605C _h , 685C _h , and 705C _h : Disable Operation Option Code | 32 |
| 4.1.4 | Object 605D _h , 685D _h , and 705D _h : Halt Option Code | 32 |
| 4.1.5 | Object 605E _h , 685E _h , and 705E _h : Fault Reaction Option Code | 33 |
| 4.1.6 | Object 6060 _h , 6860 _h , and 7060 _h : Modes of Operation | 33 |
| 4.1.7 | Object 6061 _h , 6861 _h , and 7061 _h : Modes of Operation Display | 34 |
| 4.1.8 | Object 606A _h , 686A _h , and 706A _h : Sensor Selection Code | 34 |
| 4.1.9 | Object 6085 _h , 6885 _h , and 7085 _h : Quick Stop Deceleration | 35 |
| 4.1.10 | Object 608F _h , 688F _h , and 708F _h : Position Encoder Resolution | 35 |
| 4.1.11 | Object 60C5 _h , 68C5 _h , and 70C5 _h : Max Acceleration | 35 |
| 4.1.12 | Object 60C6 _h , 68C6 _h , and 70C6 _h : Max Deceleration | 36 |
| 4.1.13 | Object 60FD _h , 68FD _h , and 70FD _h : Digital Inputs | 36 |
| 4.1.14 | Object 6502 _h , 6D02 _h , and 7502 _h : Supported Drive Modes | 37 |

| | | |
|----------|---|----|
| 4.2 | Profile Position Mode..... | 38 |
| 4.2.1 | Detailed object specifications..... | 38 |
| 4.2.1.1 | Object 6040 _h , 6840 _h , and 7040 _h : Controlword | 39 |
| 4.2.1.2 | Object 6041 _h , 6841 _h , and 7041 _h : Statusword..... | 40 |
| 4.2.1.3 | Object 6062 _h , 6862 _h , and 7062 _h : Position Demand Value | 41 |
| 4.2.1.4 | Object 6063 _h , 6863 _h , and 7063 _h : Position Actual Internal Value..... | 41 |
| 4.2.1.5 | Object 6064 _h , 6864 _h , and 7064 _h : Position Actual Value | 41 |
| 4.2.1.6 | Object 6065 _h , 6865 _h , and 7065 _h : Following Error Window..... | 42 |
| 4.2.1.7 | Object 6067 _h , 6867 _h , and 7067 _h : Position Window..... | 42 |
| 4.2.1.8 | Object 6068 _h , 6868 _h , and 7068 _h : Position Window Time | 43 |
| 4.2.1.9 | Object 606C _h , 686C _h , and 706C _h : Velocity Actual Value..... | 43 |
| 4.2.1.10 | Object 607A _h , 687A _h , and 707A _h : Target Position..... | 43 |
| 4.2.1.11 | Object 607D _h , 687D _h , and 707D _h : Software Position Limit | 44 |
| 4.2.1.12 | Object 607F _h , 687F _h , and 707F _h : Maximum Profile Velocity..... | 44 |
| 4.2.1.13 | Object 6081 _h , 6881 _h , and 7081 _h : Profile Velocity..... | 44 |
| 4.2.1.14 | Object 6083 _h , 6883 _h , and 7083 _h : Profile Acceleration | 45 |
| 4.2.1.15 | Object 6084 _h , 6884 _h , and 7084 _h : Profile Deceleration..... | 45 |
| 4.2.1.16 | Object 6086 _h , 6886 _h , and 7086 _h : Motion Profile Type..... | 45 |
| 4.2.2 | How to move a motor in <i>pp</i> mode..... | 46 |
| 4.3 | Homing Mode..... | 47 |
| 4.3.1 | Connecting home switch, left switch, and right switch..... | 48 |
| 4.3.1.1 | TMC-341/342/343 | 48 |
| 4.3.1.2 | TMC-351..... | 49 |
| 4.3.2 | Connecting a brake..... | 50 |
| 4.3.2.1 | TMC-341/342/343 | 50 |
| 4.3.2.2 | TMC-351..... | 50 |
| 4.3.3 | Homing Methods..... | 51 |
| 4.3.4 | Detailed Object Definitions | 54 |
| 4.3.4.1 | Object 6040 _h , 6840 _h , and 7040 _h : Controlword | 54 |
| 4.3.4.2 | Object 6041 _h , 6841 _h , and 7041 _h : Statusword..... | 55 |
| 4.3.4.3 | Object 606C _h , 686C _h , and 706C _h : Velocity Actual Value..... | 55 |
| 4.3.4.4 | Object 607C _h , 687C _h , and 707C _h : Home Offset..... | 56 |
| 4.3.4.5 | Object 6098 _h , 6898 _h , and 7098 _h : Homing Method | 56 |
| 4.3.4.6 | Object 6099 _h , 6899 _h , and 7099 _h : Homing Speeds | 57 |
| 4.3.4.7 | Object 609A _h , 689A _h , and 709A _h : Homing Acceleration | 57 |
| 4.3.4.8 | Object 2100 _h : Home Offset Display | 57 |
| 4.3.5 | How to start a homing in <i>hm</i> mode | 58 |
| 4.4 | Velocity Mode..... | 59 |
| 4.4.1 | Detailed object definitions | 59 |
| 4.4.1.1 | Object 6040 _h , 6840 _h , and 7040 _h : Controlword | 59 |
| 4.4.1.2 | Object 6041 _h , 6841 _h , and 7041 _h : Statusword..... | 60 |
| 4.4.1.3 | Object 6042 _h , 6842 _h , and 7042 _h : <i>vl</i> Target Velocity..... | 60 |
| 4.4.1.4 | Object 6043 _h , 6843 _h , and 7043 _h : <i>vl</i> Velocity Demand | 61 |
| 4.4.1.5 | Object 6044 _h , 6844 _h , and 7044 _h : <i>vl</i> Velocity Actual Value | 61 |
| 4.4.1.6 | Object 6046 _h , 6846 _h , and 7046 _h : <i>vl</i> Velocity Min Max Amount..... | 61 |
| 4.4.1.7 | Object 6048 _h , 6848 _h , and 7048 _h : <i>vl</i> Velocity Acceleration..... | 62 |
| 4.4.1.8 | Object 6049 _h , 6849 _h , and 7049 _h : <i>vl</i> Velocity Deceleration | 62 |
| 4.4.2 | How to Move a Motor in <i>vl</i> Mode..... | 63 |
| 4.5 | Profile Velocity Mode | 64 |
| 4.5.1.1 | Object 6040 _h , 6840 _h , and 7040 _h : Controlword | 64 |
| 4.5.1.2 | Object 6041 _h , 6841 _h , and 7041 _h : Statusword..... | 65 |
| 4.5.1.3 | Object 6062 _h , 6862 _h , and 7062 _h : Position Demand Value | 65 |
| 4.5.1.4 | Object 6063 _h , 6863 _h , and 7063 _h : Position Actual Internal Value..... | 66 |
| 4.5.1.5 | Object 6064 _h , 6864 _h , and 7064 _h : Position Actual Value | 66 |
| 4.5.1.6 | Object 6065 _h , 6865 _h , and 7065 _h : Following Error Window..... | 66 |
| 4.5.1.7 | Object 606C _h , 686C _h , and 706C _h : Velocity Actual Value..... | 67 |
| 4.5.1.8 | Object 607D _h , 687D _h , and 707D _h : Software Position Limit | 67 |
| 4.5.1.9 | Object 6083 _h , 6883 _h , and 7083 _h : Profile Acceleration | 67 |

| | | |
|----------|---|----|
| 4.5.1.10 | Object 6084 _h , 6884 _h , and 7084 _h : Profile Deceleration..... | 68 |
| 4.5.1.11 | Object 60FF _h , 62FF _h , and 64FF _h : Target Velocity..... | 68 |
| 4.5.2 | How to Move a Motor in <i>pv</i> Mode..... | 68 |
| 5 | Manufacturer Specific Area..... | 69 |
| 5.1 | Detailed Object Specifications..... | 69 |
| 5.1.1 | Object 2000 _h , 2200 _h , and 2400 _h : Microstep Resolution..... | 69 |
| 5.1.2 | Object 2001 _h , 2201 _h , and 2401 _h : Fullstep Resolution..... | 69 |
| 5.1.3 | Object 2002 _h , 2202 _h , and 2402 _h : Brake Delay Times..... | 70 |
| 5.1.4 | Object 2003 _h , 2203 _h , and 2403 _h : Maximum Current..... | 70 |
| 5.1.5 | Object 2004 _h , 2204 _h , and 2404 _h : Standby Current..... | 71 |
| 5.1.6 | Object 2005 _h , 2205 _h , and 2405 _h : Limit Switches..... | 71 |
| 5.1.7 | Object 2006 _h , 2206 _h , and 2406 _h : Mixed Decay Threshold..... | 72 |
| 5.1.8 | Object 2007 _h , 2207 _h , and 2407 _h : Stall Detection Threshold..... | 72 |
| 5.1.9 | Object 2009 _h , 2209 _h , and 2409 _h : Fullstep Threshold..... | 72 |
| 5.1.10 | Object 200A _h , 220A _h , and 240A _h : Enable Drive Delay Time..... | 73 |
| 5.1.11 | Object 200B _h , 220B _h , and 240B _h : Encoder parameters..... | 73 |
| 5.1.12 | Object 200C _h , 220C _h , and 240C _h : Brake Current Feed..... | 74 |
| 5.1.13 | Object 204E _h , 224E _h , and 244E _h : Boost Current..... | 74 |
| 5.1.14 | Object 2085 _h , 2285 _h , and 2485 _h : Ramp Divisor..... | 74 |
| 5.1.15 | Object 2086 _h , 2286 _h , and 2486 _h : Pulse Divisor..... | 75 |
| 5.1.16 | Object 2087 _h , 2287 _h , and 2487 _h : Maximum Velocity..... | 75 |
| 5.1.17 | Object 2088 _h , 2288 _h , and 2488 _h : Maximum Acceleration..... | 76 |
| 5.1.18 | Object 208C _h , 228C _h , and 248C _h : Velocity Dimension Index..... | 76 |
| 5.1.19 | Object 208E _h , 228E _h , and 248E _h : Acceleration Dimension Index..... | 77 |
| 5.1.20 | Object 2100 _h , 2300 _h , and 2500 _h : Home Offset Display..... | 77 |
| 5.1.21 | Object 2101 _h , 2301 _h , and 2501 _h : Actual stallGuard™ Load Value..... | 77 |
| 5.1.22 | Object 2102 _h , 2302 _h , and 2502 _h : Driver Error Flags..... | 78 |
| 5.1.23 | Object 2103 _h , 2303 _h , and 2503 _h : Pulse Divisor Display..... | 78 |
| 5.1.24 | Object 2104 _h , 2304 _h , and 2504 _h : Maximum Velocity Display..... | 78 |
| 5.1.25 | Object 2105 _h , 2305 _h , and 2505 _h : Ramp Divisor Display..... | 79 |
| 5.1.26 | Object 2106 _h , 2306 _h , and 2506 _h : Maximum Acceleration Display..... | 79 |
| 5.1.27 | Object 2107 _h , 2307 _h , and 2507 _h : Microstep Resolution Display..... | 79 |
| 5.1.28 | Object 2700 _h : TMCL™ Direct Communication..... | 79 |
| 5.1.29 | Object 2701 _h : Manufacturer Specific Mode..... | 80 |
| 5.1.30 | Object 2702 _h : Digital Inputs..... | 80 |
| 5.1.31 | Object 2703 _h : Digital Outputs..... | 80 |
| 5.1.32 | Object 2704 _h : CAN Bit Rate..... | 81 |
| 5.1.33 | Object 2705 _h : Node ID..... | 81 |
| 5.1.34 | Object 2706 _h : Store..... | 82 |
| 5.1.35 | Object 2707 _h : CAN bit Rate Load..... | 82 |
| 5.1.36 | Object 2708 _h : Node ID load..... | 82 |
| 5.1.37 | Object 2709 _h : Encoder Interface..... | 82 |
| 5.1.38 | Object 270A _h : Encoder Interface Display..... | 83 |
| 5.1.39 | Object 270E _h : Device Analog Inputs..... | 83 |
| 5.1.40 | Object 2710 _h : Pull-up Resistors (TMCM-351 only)..... | 83 |
| 6 | Emergency Messages..... | 84 |
| 7 | Life Support Policy..... | 86 |
| 8 | Revision History..... | 87 |
| 8.1 | Firmware revision..... | 87 |
| 8.2 | Document Revision..... | 88 |
| 9 | References..... | 89 |

Table of Figures

| | |
|--|----|
| Figure 2.1 Overview CANopen NMT state machine..... | 9 |
| Figure 2.2 Communication architecture | 10 |
| Figure 2.3 Device model..... | 11 |
| Figure 3.1 Restore procedure | 17 |
| Figure 4.1 Finite state machine | 38 |
| Figure 4.2 Homing mode function (motor 0)..... | 47 |
| Figure 4.3: 68-pin connector of the TMCM-34x..... | 48 |
| Figure 4.4: Pin order of the connector..... | 48 |
| Figure 4.5: Connectors of TMCM-351 | 49 |
| Figure 4.6 Homing on negative switch and index pulse..... | 51 |
| Figure 4.7 Homing on positive limit switch and index pulse..... | 51 |
| Figure 4.8 Homing on positive home switch and index pulse..... | 52 |
| Figure 4.9 Homing on negative home switch and index pulse..... | 52 |
| Figure 4.10 Homing without index pulse..... | 53 |
| Figure 4.11 Homing on index pulse..... | 53 |
| Figure 4.12 Home offset definition..... | 56 |
| Figure 4.13 Transfer characteristic of the velocity acceleration | 62 |
| Figure 5.1 Brake output timing..... | 70 |

1 Preface

This document specifies objects and modes of operation of the TRINAMIC modules TCM-341, TCM-342, TCM-343, and TCM-351 with CANopen firmware. As these products are stepper motor controller and driver modules the use of the CiA DSP402 protocol (described in the *CiA CANopen drives and motion control device profile, Part 2*) is fundamental. The CANopen firmware is designed to fulfill the DS301 version 4.02 and DS402 version 3.0 standards. The CiA conformance has also been tested. This CANopen firmware has been developed to turn the ARM-based TRINAMIC motion control modules into CANopen compatible ones. There are versions for each module. This manual assumes that the reader is already familiar with the basics of the CANopen protocol (especially DS301 and DS402). On the following pages you will find a short overview and afterwards the information will be more in detail.

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

1.1 General Features of the CANopen Implementation

MAIN CHARACTERISTICS

Communication according to standard CiA-301 V4.1

CAN bit rate: 20... 1000kBit/s

CAN ID: 11 bit

Node ID: 1... 127 (use vendor specific objects)

NMT services: NMT slave

SDO communication

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

PDO communication

- Producer
- Consumer
- *RPDOs*
 - Axis 0: 1, 2, 3, 4, 6
 - Axis 1: 65, 66, 67, 68, 70
 - Axis 2: 129, 130, 131, 132, 134
 - Transmission modes: asynchronous
 - Dynamic mapping with max. 3 mapping entries
 - Default mappings: according to CiA-402 for PDO 1, 2, 3 and 6, manufacturer specific for PDO4
- *TPDOs*
 - Axis 0: 1, 2, 3, 4, 6
 - Axis 1: 65, 66, 67, 68, 70
 - Axis 2: 129, 130, 131, 132, 134
 - Transmission modes: asynchronous, asynchronous with event timer, synchronous
 - Dynamic mapping with max. 3 mapping entries
 - Default mappings: according to CiA-402 for PDO 1, 2, 3 and 6, manufacturer specific for PDO

Further Characteristics

SYNC: consumer (TPDO3, TPDO6, TPDO67, TPDO70, TPDO131, and TPDO134 are synchronous PDOs)

Emergency: producer

RTR: supported only for node guarding/life guarding

Heartbeat: consumer and producer

1.2 Abbreviations

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

Table 1.1 Abbreviations

1.3 Firmware Update

The software running on the microprocessor 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. New versions can be downloaded free of charge from the TRINAMIC website (<http://www.trinamic.com>). The CANopen firmware can be loaded into the modules easily just by using 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 which synchronizes devices. The functionality the application layer offers to an application is logically divided over different *service data objects* (SDO) in the application layer. A service object offers a specific functionality and all the related services.

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

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

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

Table 2.1 Service primitives

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

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

Table 2.2 Service types

2.2 NMT State Machine

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

Starting and resetting the device is controlled via the state machine. The NMT state machine contains the following states:

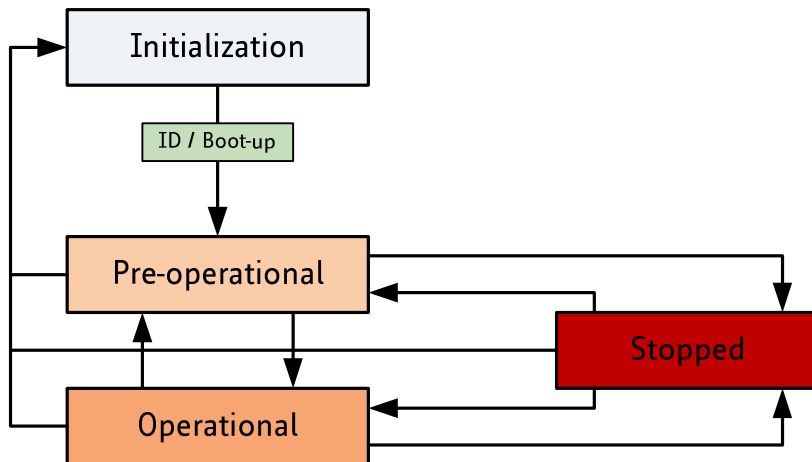


Figure 2.1 Overview CANopen NMT state machine

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

After the device initialization is finished, the device automatically transits to the **Pre-operational state** and indicates this state transition by sending the boot-up message. This way the device indicates that it is ready to work. A device that stays in Pre-operational state may start to transmit SYNC-, time stamp- or heartbeat message. In contrast to the PDO communication which 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 to stop the device), detect remote device boot-ups and error conditions.

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

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

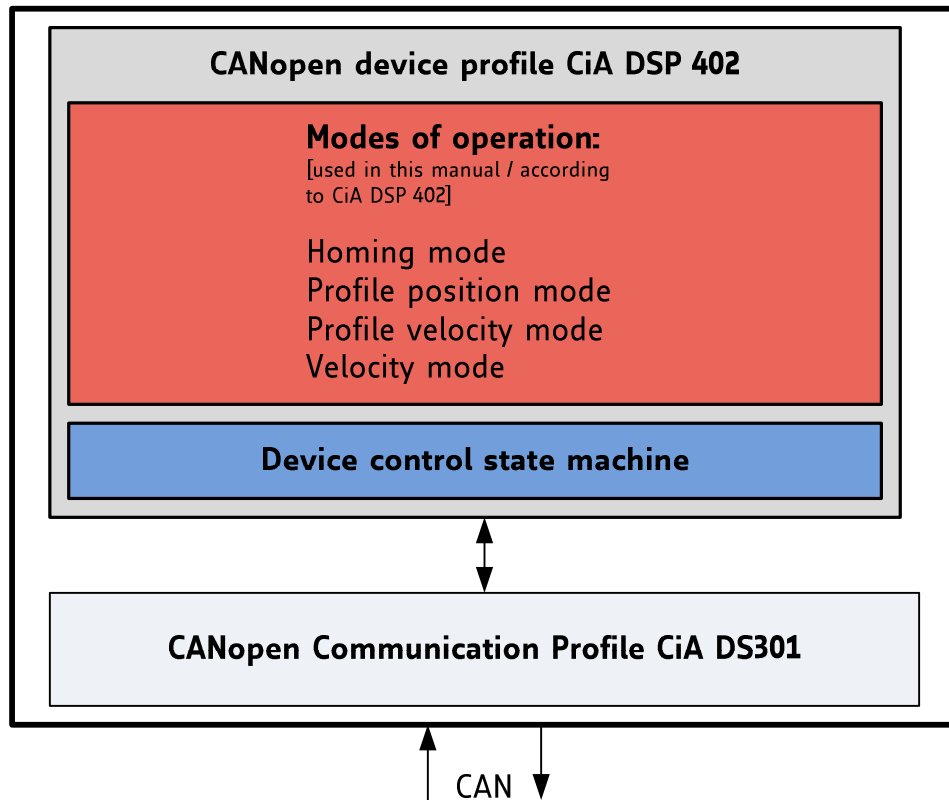


Figure 2.2 Communication architecture

2.3 Device Model

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

Communication This function unit provides the communication objects and the appropriate functionality to transport data items via the underlying network structure.

Object dictionary The object dictionary is a collection of all the data items which have an influence on the behavior of the application objects, the communication objects and the state machine used on this device.

Application The application comprises the functionality of the device with respect to the interaction with the process environment.

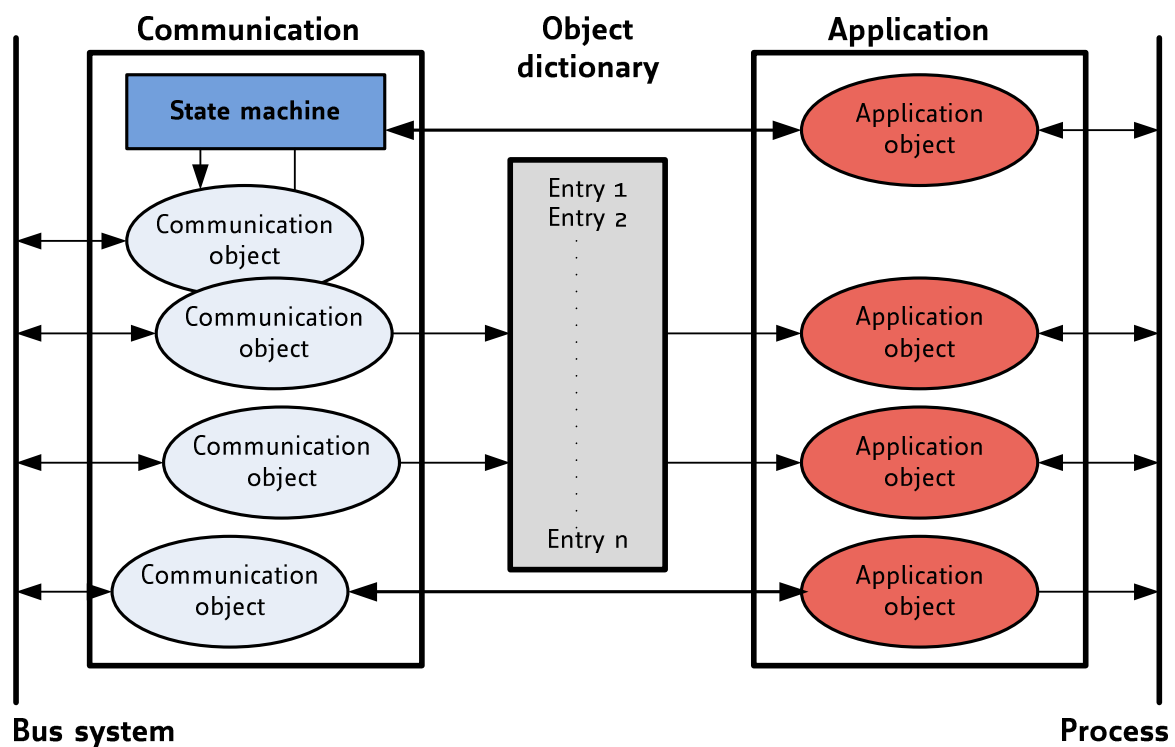


Figure 2.3 Device model

2.4 Object Dictionary

The most important part of a device profile is the object dictionary description. The Object Dictionary is essentially a grouping of objects accessible via the network in an ordered pre-defined fashion. Each object within the dictionary is addressed using a 16-bit index.

THE OVERALL LAYOUT OF THE STANDARD OBJECT DICTIONARY IS AS FOLLOWS:

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

Table 2.3 Object Dictionary

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

*² The manufacturer segment at indices 2000_h through 5FFF_h contains manufacturer specific objects. These objects control special features of the TRINAMIC TMCM-34x resp. TMCM-351 modules.

*³ 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. The device profiles use entries from 6000_h to 9FFF_h to describe the device parameters and the device functionality.

3 Communication Objects

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 gives additional information about optional functionality of the device.

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|-------------|--------------------|
| 00 _h | r0 | no | UNSIGNED32 | 40192 _h |

3.1.2 Object 1001_h: Error Register

This object is an error register. The module can map internal errors and object 1001_h is part of an emergency object.

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|-------------|---------------|
| 00 _h | r0 | yes | UNSIGNED8 | no |

STRUCTURE OF THE ERROR REGISTER

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

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

3.1.3 Object 1005_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.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-------------------|---------------------|-------------|------------|
| 1005 _h | COB-ID SYNC Message | Variable | UNSIGNED32 |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|-------------|-----------------|
| 00 _h | rw | no | UNSIGNED32 | 80 _h |

3.1.4 Object 1008_h: Manufacturer Device Name

This object contains the manufacturer device name.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-------------------|--------------------------|-------------|----------------|
| 1008 _h | Manufacturer device name | Variable | visible string |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|-------------|---|
| 00 _h | Const. | no | no | <i>module specific:</i> TMC-341 TMC-342 TMC-343 TMC-351 |

3.1.5 Object 1009_h: Manufacturer Hardware Version

This object contains the hardware version description.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-------------------|-------------------------------|-------------|----------------|
| 1009 _h | Manufacturer hardware version | Variable | visible string |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|-------------|------------------------------------|
| 00 _h | Const. | no | no | <i>depends on hardware version</i> |

3.1.6 Object 100A_h: Manufacturer Software Version

This object contains the software version description.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-------------------|-------------------------------|-------------|----------------|
| 100A _h | Manufacturer software version | Variable | visible string |

ENTRY DESCRIPTION

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

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 | Object Code | Data Type |
|-------------------|------------|-------------|------------|
| 100C _h | Guard time | Variable | UNSIGNED16 |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--|-------------|-------------|-------------------|
| 00 _h | rw <i>ro, if life guarding is not supported</i> | no | UNSIGNED16 | 0000 _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 | Object Code | Data Type |
|-------------------|------------------|-------------|-----------|
| 100D _h | Life time factor | Variable | UNSIGNED8 |

ENTRY DESCRIPTION

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

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

SEVERAL PARAMETER GROUPS ARE DISTINGUISHED:

- Sub-index 0_h contains the largest sub-index that is supported.
- Sub-index 1_h saves all parameters.
- Sub-index 2_h: saves communication parameters 100C_h, 100D_h, 1015_h, 1017_h, and 1029_h.
- Sub-index 4_h stores parameters for the motor axis 0.
- Sub-index 5_h stores parameters for the motor axis 1.
- Sub-index 6_h stores parameters for the motor axis 2.
- Sub-index 7_h: saves device parameters / TRINAMIC specific parameters (2000_h... 270E_h).

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

Storage write access structure

| Signature | MSB | | LSB | |
|----------------|-----------------|-----------------|-----------------|-----------------|
| ISO 8859 ASCII | e | v | a | s |
| hex | 65 _h | 76 _h | 61 _h | 73 _h |

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

If a wrong signature is written, the device refuses to store and responds with Abort SDO transfer (abort code: 0800 002x_h).

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

Storage read access structure

| UNSIGNED 32 | | | |
|-------------|----------|-----|-----|
| MSB | | LSB | |
| bits | 31-2 | 1 | 0 |
| | reserved | 1/0 | 1/0 |

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

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

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-------------------|------------------|-------------|-------------|
| 1010 _h | Store parameters | ARRAY | UNSIGNED 32 |

ENTRY DESCRIPTION

| Sub-Index | Description | Access | PDO Mapping | Value Range | Default Value |
|-----------------|---|--------|-------------|---------------------------------|---------------|
| 00 _h | Largest sub-index supported | ro | no | 1 _h -7F _h | no |
| 01 _h | Save all parameters | rw | no | UNSIGNED32 | no |
| 02 _h | Save special communication parameters | rw | no | UNSIGNED32 | no |
| 04 _h | Save motor axis 1 parameters | rw | no | UNSIGNED32 | no |
| 05 _h | Save motor axis 2 parameters | rw | no | UNSIGNED32 | no |
| 06 _h | Save motor axis 3 parameters | rw | no | UNSIGNED32 | no |
| 07 _h | Save device parameters (TRINAMIC specific parameters) | rw | no | UNSIGNED32 | no |

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

3.1.10 Object 1011_h: Restore Default Parameters

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

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

SEVERAL PARAMETER GROUPS ARE DISTINGUISHED:

- Sub-index 0_h: contains the largest sub-index that is supported.
- Sub-index 1_h: restores all parameters.
- Sub-index 2_h: restores communication parameters 100C_h, 100D_h, 1015_h, 1017_h, and 1029_h.
- Sub-index 4_h: restores motor axis 0 parameters
- Sub-index 5_h: restores motor axis 1 parameters
- Sub-index 6_h: restores motor axis 2 parameters
- Sub-index 7_h: restores device parameters / TRINAMIC specific parameters (2000_h... 270E_h).

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

| Signature | MSB | | LSB | |
|-----------|-----------------|-----------------|-----------------|-----------------|
| ASCII | d | a | o | l |
| hex | 64 _h | 61 _h | 6F _h | 6C _h |

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

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

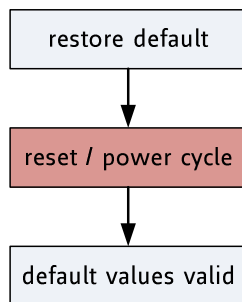


Figure 3.1 Restore procedure

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

Structure of restore read access

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

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-------------------|----------------------------|-------------|-------------|
| 1011 _h | Restore default parameters | ARRAY | UNSIGNED 32 |

ENTRY DESCRIPTION

| Sub-Index | Description | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--|--------|-------------|---------------------------------|---------------|
| 00 _h | Largest sub-index supported | ro | no | 1 _h -7F _h | no |
| 01 _h | Restore all default parameters | rw | no | UNSIGNED32 | no |
| 02 _h | Restore special communication parameters | rw | no | UNSIGNED32 | no |
| 04 _h | Restore motor axis 0 parameters | rw | no | UNSIGNED32 | no |
| 05 _h | Restore motor axis 1 parameters | rw | no | UNSIGNED32 | no |
| 06 _h | Restore motor axis 2 parameters | rw | no | UNSIGNED32 | no |
| 07 _h | Restore device parameters (TRINAMIC specific parameters) | rw | no | UNSIGNED32 | no |

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

3.1.11 Object 1014_h: COB-ID Emergency Object

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

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-------------------|-------------------------|-------------|------------|
| 1014 _h | COB-ID emergency object | Variable | UNSIGNED32 |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|-------------|---------------------------|
| 00 _h | rw | no | UNSIGNED32 | 80 _h + NODE ID |

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 | Object Code | Data Type |
|-------------------|-------------------|-------------|------------|
| 1015 _h | Inhibit time EMCY | Variable | UNSIGNED16 |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|-------------|---------------|
| 00 _h | rw | no | UNSIGNED16 | 0 |

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.

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

Table 3.1 Structure of consumer heartbeat time entry

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-------------------|-------------------------|-------------|------------|
| 1016 _h | Consumer heartbeat time | ARRAY | UNSIGNED32 |

ENTRY DESCRIPTION

| Sub-Index | Description | Access | PDO Mapping | Value Range | Default Value |
|-----------------|-------------------------|--------|-------------|-------------|---------------|
| 00 _h | Number of entries | ro | no | 0... 127 | no |
| 01 _h | Consumer heartbeat time | rw | no | UNSIGNED32 | no |

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 | Object Code | Data Type |
|-------------------|-------------------------|-------------|------------|
| 1017 _h | Producer heartbeat time | Variable | UNSIGNED16 |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|-------------|---------------|
| 00 _h | rw | no | UNSIGNED16 | 0 |

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 0286_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 Code | Data Type | Category |
|-------------------|-----------------|-------------|-----------|----------|
| 1018 _h | Identity object | RECORD | Identity | Optional |

ENTRY DESCRIPTION

| Sub-Index | Description | Access | PDO Mapping | Value Range | Default Value |
|-----------------|----------------------|--------|-------------|-------------|---|
| 00 _h | Number of entries | ro | no | 0.. 3 | number of entries:3 |
| 01 _h | Vendor ID (TRINAMIC) | ro | no | UNSIGNED32 | reads 0286 _h |
| 02 _h | Product code | ro | no | UNSIGNED32 | depends on device e.g. 341, 342, 343, 351 |
| 03 _h | Revision number | ro | no | UNSIGNED32 | firmware revision number; reads e.g. 2003 _h for version 2.3 |
| 04 _h | Serial number | ro | no | UNSIGNED32 | serial number |

3.1.16 Object 1023_h: OS Command

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

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type | Category |
|-------------------|------------|-------------|-------------|----------|
| 1023 _h | OS command | RECORD | Command Par | Optional |

ENTRY DESCRIPTION

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

3.1.17 Object 1029_h: Error Behavior

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

- Communication error
- Application error

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-------------------|----------------|-------------|-----------|
| 1029 _h | Error behavior | ARRAY | UNSIGNED8 |

ENTRY DESCRIPTION

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

3.1.18 Objects 1400_h-1405_h: Receive PDO Communication Parameter

These objects contain the communication parameters for the PDOs of motor 0 the device is able to receive. The sub-index 0_h contains the number of valid entries within the communication record. Its value is at least 2.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|--------------------------------------|-----------------------|-------------|-------------|
| 1400 _h -1405 _h | Receive PDO parameter | RECORD | PDO CommPar |
| 1400 _h | RPDO 1 | RECORD | PDO CommPar |
| 1401 _h | RPDO 2 | RECORD | PDO CommPar |
| 1402 _h | RPDO 3 | RECORD | PDO CommPar |
| 1403 _h | RPDO 4 | RECORD | PDO CommPar |
| 1405 _h | RPDO 6 | RECORD | PDO CommPar |

ENTRY DESCRIPTION

| Sub-Index | Description | Access | PDO Mapping | Value Range | Default Value |
|-----------------|-----------------------------|--------|-------------|----------------------|---|
| 00 _h | Largest sub-index supported | ro | no | Number of entries: 2 | 2 |
| 01 _h | COB-ID used by PDO | rw | no | 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 Index 1405 _h : 0 |
| 02 _h | Transmission type | rw | no | UNSIGNED8 | Index 1400 _h : Ff _h Index 1401 _h : Ff _h Index 1402 _h : Ff _h Index 1403 _h : Ff _h Index 1405 _h : Ff _h |

3.1.19 Objects 1440_h-1445_h: Receive PDO Communication Parameter

This object contains the communication parameters for the PDOs of motor 1 the device is able to receive. The sub-index 0_h contains the number of valid entries within the communication record. Its value is at least 2.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|--------------------------------------|-----------------------|-------------|-------------|
| 1440 _h -1445 _h | Receive PDO parameter | RECORD | PDO CommPar |
| 1440 _h | RPDO 1 | RECORD | PDO CommPar |
| 1441 _h | RPDO 2 | RECORD | PDO CommPar |
| 1442 _h | RPDO 3 | RECORD | PDO CommPar |
| 1443 _h | RPDO 4 | RECORD | PDO CommPar |
| 1445 _h | RPDO 6 | RECORD | PDO CommPar |

ENTRY DESCRIPTION

| Sub-Index | Description | Access | PDO Mapping | Value Range | Default Value |
|-----------------|-----------------------------|--------|-------------|----------------------|---|
| 00 _h | Largest sub-index supported | ro | no | Number of entries: 2 | 2 |
| 01 _h | COB-ID used by PDO | rw | no | UNSIGNED32 | Index 1440 _h : 0 Index 1441 _h : 0 Index 1442 _h : 0 Index 1443 _h : 0 Index 1445 _h : 0 |
| 02 _h | Transmission type | rw | no | UNSIGNED8 | Index 1440 _h : Ff _h Index 1441 _h : Ff _h Index 1442 _h : Ff _h Index 1443 _h : Ff _h Index 1445 _h : Ff _h |

3.1.20 Objects 1480_h-1485_h: Receive PDO Communication Parameter

These objects contain the communication parameters for the PDOs of motor 2 the device is able to receive. The sub-index 0_h contains the number of valid entries within the communication record. Its value is at least 2.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|--------------------------------------|-----------------------|-------------|-------------|
| 1400 _h -1405 _h | Receive PDO parameter | RECORD | PDO CommPar |
| 1480 _h | RPDO 1 | RECORD | PDO CommPar |
| 1481 _h | RPDO 2 | RECORD | PDO CommPar |
| 1482 _h | RPDO 3 | RECORD | PDO CommPar |
| 1483 _h | RPDO 4 | RECORD | PDO CommPar |
| 1485 _h | RPDO 6 | RECORD | PDO CommPar |

ENTRY DESCRIPTION

| Sub-Index | Description | Access | PDO Mapping | Value Range | Default Value |
|-----------------|-----------------------------|--------|-------------|----------------------|---|
| 00 _h | Largest sub-index supported | ro | no | Number of entries: 2 | 2 |
| 01 _h | COB-ID used by PDO | rw | no | UNSIGNED32 | Index 1480 _h : 0 Index 1481 _h : 0 Index 1482 _h : 0 Index 1483 _h : 0 Index 1485 _h : 0 |
| 02 _h | Transmission type | rw | no | UNSIGNED8 | Index 1480 _h : Ff _h Index 1481 _h : Ff _h Index 1482 _h : Ff _h Index 1483 _h : Ff _h Index 1485 _h : Ff _h |

3.1.21 Objects 1600_h-1605_h: Receive PDO Mapping

These objects contain the mapping for the PDOs of motor 0 the device is able to receive. The sub-index 0_h contains the number of valid entries within the mapping record. This number of entries is also the number of the application variables which shall be transmitted/received with the corresponding PDO. The sub-indices from 1_h to the number of entries contain the information about the mapped application variables. These entries describe the PDO contents by their index, sub-index and length. The values are hexadecimally coded.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|--------------------------------------|---------------------|-------------|-------------|
| 1600 _h -1605 _h | Receive PDO mapping | RECORD | PDO Mapping |
| 1600 _h | Mapping for RPDO 1 | RECORD | PDO Mapping |
| 1601 _h | Mapping for RPDO 2 | RECORD | PDO Mapping |
| 1602 _h | Mapping for RPDO 3 | RECORD | PDO Mapping |
| 1603 _h | Mapping for RPDO 4 | RECORD | PDO Mapping |
| 1605 _h | Mapping for RPDO 6 | RECORD | PDO Mapping |

ENTRY DESCRIPTION

| Sub-Index | Description | Access | PDO Mapping | Value Range | Default Value |
|-----------------|---|--------|-------------|-------------------------|---|
| 00 _h | Number of mapped application objects in PDO | rw | no | 1... 3 0 deactivated | Index 1600 _h : 1 Index 1601 _h : 2 Index 1602 _h : 2 Index 1603 _h : 2 Index 1605 _h : 2 |
| 01 _h | Mapping entry 1 | rw | no | UNSIGNED32 | Index 1600 _h : 60400010 _h Index 1601 _h : 60400010 _h Index 1602 _h : 60400010 _h Index 1603 _h : 60400010 _h Index 1605 _h : 60400010 _h |
| 02 _h | Mapping entry 2 | rw | no | UNSIGNED32 | Index 1600 _h : 0 Index 1601 _h : 60600008 _h Index 1602 _h : 607A0020 _h Index 1603 _h : 60FF0020 _h Index 1605 _h : 60420010 _h |
| 03 _h | Mapping entry 3 | rw | no | UNSIGNED32 | Index 1600 _h : 0 Index 1601 _h : 0 Index 1602 _h : 0 Index 1603 _h : 0 Index 1605 _h : 0 |

3.1.22 Objects 1640_h-1645_h: Receive PDO Mapping

These objects contain the mapping for the PDOs of motor 1 the device is able to receive. The sub-index 0_h contains the number of valid entries within the mapping record. This number of entries is also the number of the application variables which shall be transmitted/received with the corresponding PDO. The sub-indices from 1_h to the number of entries contain the information about the mapped application variables. These entries describe the PDO contents by their index, sub-index and length. All three values are hexadecimal coded.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|--------------------------------------|---------------------|-------------|-------------|
| 1640 _h -1645 _h | Receive PDO mapping | RECORD | PDO Mapping |
| 1640 _h | Mapping for RPDO 65 | RECORD | PDO Mapping |
| 1641 _h | Mapping for RPDO 66 | RECORD | PDO Mapping |
| 1642 _h | Mapping for RPDO 67 | RECORD | PDO Mapping |
| 1643 _h | Mapping for RPDO 68 | RECORD | PDO Mapping |
| 1645 _h | Mapping for RPDO 70 | RECORD | PDO Mapping |

ENTRY DESCRIPTION

| Sub-Index | Description | Access | PDO Mapping | Value Range | Default Value |
|-----------------|---|--------|-------------|-------------------------|---|
| 00 _h | Number of mapped application objects in PDO | rw | no | 1... 3 0 deactivated | Index 1640 _h : 1 Index 1641 _h : 2 Index 1642 _h : 2 Index 1643 _h : 2 Index 1645 _h : 2 |
| 01 _h | Mapping entry 1 | rw | no | UNSIGNED32 | Index 1640 _h : 68400010 _h Index 1641 _h : 68400010 _h Index 1642 _h : 68400010 _h Index 1643 _h : 68400010 _h Index 1645 _h : 68400010 _h |
| 02 _h | Mapping entry 2 | rw | no | UNSIGNED32 | Index 1640 _h : 0 Index 1641 _h : 68600008 _h Index 1642 _h : 687A0020 _h Index 1643 _h : 68FF0020 _h Index 1645 _h : 68420010 _h |
| 03 _h | Mapping entry 3 | rw | no | UNSIGNED32 | Index 1640 _h : 0 Index 1641 _h : 0 Index 1642 _h : 0 Index 1643 _h : 0 Index 1645 _h : 0 |

3.1.23 Objects 1680_h-1685_h: Receive PDO Mapping

These objects contain the mapping for the PDOs of motor 2 the device is able to receive. The sub-index 0_h contains the number of valid entries within the mapping record. This number of entries is also the number of the application variables which shall be transmitted/received with the corresponding PDO. The sub-indices from 1_h to the number of entries contain the information about the mapped application variables. These entries describe the PDO contents by their index, sub-index and length. All three values are hexadecimal coded.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|--------------------------------------|----------------------|-------------|-------------|
| 1680 _h -1685 _h | Receive PDO mapping | RECORD | PDO Mapping |
| 1680 _h | Mapping for RPDO 129 | RECORD | PDO Mapping |
| 1681 _h | Mapping for RPDO 130 | RECORD | PDO Mapping |
| 1682 _h | Mapping for RPDO 131 | RECORD | PDO Mapping |
| 1683 _h | Mapping for RPDO 132 | RECORD | PDO Mapping |
| 1685 _h | Mapping for RPDO 134 | RECORD | PDO Mapping |

ENTRY DESCRIPTION

| Sub-Index | Description | Access | PDO Mapping | Value Range | Default Value |
|-----------------|---|--------|-------------|-------------------------|---|
| 00 _h | Number of mapped application objects in PDO | rw | no | 1... 3 0 deactivated | Index 1680 _h : 1 Index 1681 _h : 2 Index 1682 _h : 2 Index 1683 _h : 2 Index 1685 _h : 2 |
| 01 _h | Mapping entry 1 | rw | no | UNSIGNED32 | Index 1680 _h : 70400010 _h Index 1681 _h : 70400010 _h Index 1682 _h : 70400010 _h Index 1683 _h : 70400010 _h Index 1685 _h : 70400010 _h |
| 02 _h | Mapping entry 2 | rw | no | UNSIGNED32 | Index 1680 _h : 0 Index 1681 _h : 70600008 _h Index 1682 _h : 707A0020 _h Index 1683 _h : 70FF0020 _h Index 1685 _h : 70420010 _h |
| 03 _h | Mapping entry 3 | rw | no | UNSIGNED32 | Index 1680 _h : 0 Index 1681 _h : 0 Index 1682 _h : 0 Index 1683 _h : 0 Index 1685 _h : 0 |

3.1.24 Objects 1800_h-1805_h: Transmit PDO Communication Parameter

These objects contain the communication parameters for the PDOs of motor 0 the device is able to transmit.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|--------------------------------------|---|-------------|-------------|
| 1800 _h -1805 _h | Transmit PDO communication parameter | RECORD | PDO CommPar |
| 1800 _h | TPDO 1 transmit communication parameter | RECORD | PDO CommPar |
| 1801 _h | TPDO 2 transmit communication parameter | RECORD | PDO CommPar |
| 1802 _h | TPDO 3 transmit communication parameter | RECORD | PDO CommPar |
| 1803 _h | TPDO 4 transmit communication parameter | RECORD | PDO CommPar |
| 1805 _h | TPDO 6 transmit communication parameter | RECORD | PDO CommPar |

ENTRY DESCRIPTION

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

3.1.25 Objects 1840_h-1845_h: Transmit PDO Communication Parameter

These objects contain the communication parameters for the PDOs of motor 1 the device is able to transmit.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|--------------------------------------|--|-------------|-------------|
| 1840 _h -1845 _h | Transmit PDO communication parameter | RECORD | PDO CommPar |
| 1840 _h | TPDO 65 transmit communication parameter | RECORD | PDO CommPar |
| 1841 _h | TPDO 66 transmit communication parameter | RECORD | PDO CommPar |
| 1842 _h | TPDO 67 transmit communication parameter | RECORD | PDO CommPar |
| 1843 _h | TPDO 68 transmit communication parameter | RECORD | PDO CommPar |
| 1845 _h | TPDO 70 transmit communication parameter | RECORD | PDO CommPar |

ENTRY DESCRIPTION

| Sub-Index | Description | Access | PDO Mapping | Value Range | Default Value |
|-----------------|-----------------------------|--------|-------------|--------------------------------|---|
| 00 _h | Largest sub-index supported | ro | no | 2... 5 Number of entries: 5 | 5 |
| 01 _h | TPDO 1-4, 6: COB-ID | rw | no | UNSIGNED32 | Index 1840 _h : 0 Index 1841 _h : 0 Index 1842 _h : 0 Index 1843 _h : 0 Index 1845 _h : 0 |
| 02 _h | Transmission type | rw | no | UNSIGNED8 | Index 1840 _h : ff _h Index 1841 _h : ff _h Index 1842 _h : 1 Index 1843 _h : 1 Index 1845 _h : 1 |
| 03 _h | Inhibit time | rw | no | UNSIGNED16 | 0 |
| 04 _h | Compatibility entry | ro | no | UNSIGNED8 | 0 |
| 05 _h | Event timer | rw | no | 0 not used UNSIGNED16 | 0 |

3.1.26 Objects 1880_h-1885_h: Transmit PDO Communication Parameter

These objects contain the communication parameters for the PDOs of motor 2 the device is able to transmit.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|--------------------------------------|---|-------------|-------------|
| 1880 _h -1885 _h | Transmit PDO communication parameter | RECORD | PDO CommPar |
| 1880 _h | TPDO 129 transmit communication parameter | RECORD | PDO CommPar |
| 1881 _h | TPDO 130 transmit communication parameter | RECORD | PDO CommPar |
| 1882 _h | TPDO 131 transmit communication parameter | RECORD | PDO CommPar |
| 1883 _h | TPDO 132 transmit communication parameter | RECORD | PDO CommPar |
| 1885 _h | TPDO 134 transmit communication parameter | RECORD | PDO CommPar |

ENTRY DESCRIPTION

| Sub-Index | Description | Access | PDO Mapping | Value Range | Default Value |
|-----------------|-----------------------------|--------|-------------|--------------------------------|---|
| 00 _h | Largest sub-index supported | ro | no | 2... 5 Number of entries: 5 | 5 |
| 01 _h | TPDO 1-4, 6: COB-ID | rw | no | UNSIGNED32 | Index 1880 _h : 0 Index 1881 _h : 0 Index 1882 _h : 0 Index 1883 _h : 0 Index 1885 _h : 0 |
| 02 _h | Transmission type | rw | no | UNSIGNED8 | Index 1880 _h : ff _h Index 1881 _h : ff _h Index 1882 _h : 1 Index 1883 _h : 1 Index 1885 _h : 1 |
| 03 _h | Inhibit time | rw | no | UNSIGNED16 | 0 |
| 04 _h | Compatibility entry | ro | no | UNSIGNED8 | 0 |
| 05 _h | Event timer | rw | no | 0 not used UNSIGNED16 | 0 |

3.1.27 Objects 1A00_h-1A05_h: Transmit PDO Mapping Parameter

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

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|--------------------------------------|--|-------------|-------------|
| 1A00 _h -1A05 _h | Transmit PDO mapping parameter | RECORD | PDO mapping |
| 1A00 _h | TPDO 1: transmit PDO mapping parameter | RECORD | PDO mapping |
| 1A01 _h | TPDO 2: transmit PDO mapping parameter | RECORD | PDO mapping |
| 1A02 _h | TPDO 3: transmit PDO mapping parameter | RECORD | PDO mapping |
| 1A03 _h | TPDO 4: transmit PDO mapping parameter | RECORD | PDO mapping |
| 1A05 _h | TPDO 6: transmit PDO mapping parameter | RECORD | PDO mapping |

ENTRY DESCRIPTION

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

3.1.28 Objects 1A40_h-1A45_h: Transmit PDO Mapping Parameter

These objects contain the mapping for the PDOs of motor 1 the device is able to transmit.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|--------------------------------------|---|-------------|-------------|
| 1A40 _h -1A45 _h | Transmit PDO mapping parameter | RECORD | PDO mapping |
| 1A40 _h | TPDO 65: transmit PDO mapping parameter | RECORD | PDO mapping |
| 1A41 _h | TPDO 66: transmit PDO mapping parameter | RECORD | PDO mapping |
| 1A42 _h | TPDO 67: transmit PDO mapping parameter | RECORD | PDO mapping |
| 1A43 _h | TPDO 68: transmit PDO mapping parameter | RECORD | PDO mapping |
| 1A45 _h | TPDO 70: transmit PDO mapping parameter | RECORD | PDO mapping |

ENTRY DESCRIPTION

| Sub-Index | Description | Access | PDO Mapping | Value Range | Default Value |
|-----------------|---|--------|-------------|---|---|
| 00 _h | Number of mapped application objects in PDO | rw | no | 0 deactivated Number of entries: 1... 3 | Index 1A40 _h : 1 Index 1A41 _h : 2 Index 1A42 _h : 2 Index 1A43 _h : 2 Index 1A45 _h : 2 |
| 01 _h | Mapping entry 1 | rw | no | UNSIGNED32 | Index 1A40 _h : 68410010 _h Index 1A41 _h : 68410010 _h Index 1A42 _h : 68410010 _h Index 1A43 _h : 68410010 _h Index 1A45 _h : 68410010 _h |
| 02 _h | Mapping entry 2 | rw | no | UNSIGNED32 | Index 1A40 _h : 0 Index 1A41 _h : 68610008 _h Index 1A42 _h : 68640020 _h Index 1A43 _h : 686c0020 _h Index 1A45 _h : 68440010 _h |
| 03 _h | Mapping entry 3 | rw | no | UNSIGNED32 | Index 1A40 _h : 0 Index 1A41 _h : 0 Index 1A42 _h : 0 Index 1A43 _h : 0 Index 1A45 _h : 0 |

3.1.29 Objects 1A80_h-1A85_h: Transmit PDO Mapping Parameter

These objects contain the mapping for the PDOs of motor 2 the device is able to transmit.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|--------------------------------------|--|-------------|-------------|
| 1A80 _h -1A85 _h | Transmit PDO mapping parameter | RECORD | PDO mapping |
| 1A80 _h | TPDO 129: transmit PDO mapping parameter | RECORD | PDO mapping |
| 1A81 _h | TPDO 130: transmit PDO mapping parameter | RECORD | PDO mapping |
| 1A82 _h | TPDO 131: transmit PDO mapping parameter | RECORD | PDO mapping |
| 1A83 _h | TPDO 132: transmit PDO mapping parameter | RECORD | PDO mapping |
| 1A85 _h | TPDO 134: transmit PDO mapping parameter | RECORD | PDO mapping |

ENTRY DESCRIPTION

| Sub-Index | Description | Access | PDO Mapping | Value Range | Default Value |
|-----------------|---|--------|-------------|---|---|
| 00 _h | Number of mapped application objects in PDO | rw | no | 0 deactivated Number of entries: 1... 3 | Index 1A80 _h : 1 Index 1A81 _h : 2 Index 1A82 _h : 2 Index 1A83 _h : 2 Index 1A85 _h : 2 |
| 01 _h | Mapping entry 1 | rw | no | UNSIGNED32 | Index 1A80 _h : 70410010 _h Index 1A81 _h : 70410010 _h Index 1A82 _h : 70410010 _h Index 1A83 _h : 70410010 _h Index 1A85 _h : 70410010 _h |
| 02 _h | Mapping entry 2 | rw | no | UNSIGNED32 | Index 1A80 _h : 0 Index 1A81 _h : 70610008 _h Index 1A82 _h : 70640020 _h Index 1A83 _h : 706c0020 _h Index 1A85 _h : 70440010 _h |
| 03 _h | Mapping entry 3 | rw | no | UNSIGNED32 | Index 1A80 _h : 0 Index 1A81 _h : 0 Index 1A82 _h : 0 Index 1A83 _h : 0 Index 1A85 _h : 0 |

4 Device Profile Objects (CiA402) and Modes of Operation

The PDS (power drive system) behavior depends on the activated mode of operation. The PDS implements several modes of operation. Since it is not possible to operate the modes in parallel, the user is able to activate the required function by selecting a mode of operation.

The control device writes to the *modes of operation* object in order to select the operation mode. The drive device provides the *modes of operation display* object to indicate the actual activated operation mode. Controlword, statusword, and set-points are used mode-specific. This implies the responsibility of the control device to avoid inconsistencies and erroneous behavior.

THE FOLLOWING OPERATION MODES CAN BE CHOSEN:

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

Please refer to object 6060_h (section 4.1.6) for information about how to choose an operation mode.

4.1 Detailed Object Specifications

4.1.1 Object 605A_h, 685A_h, and 705A_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 operations. The following quick stop option codes are supported in the current version of the CANopen firmware:

VALUE DEFINITION

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

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|------------------------|-------------|-----------|
| 605A _h (motor 0) | Quick stop option code | Variable | SIGNED16 |
| 685A _h (motor 1) | | | |
| 705A _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|----------------------------|---------------|
| 00 _h | rw | Refer to CiA402-3 | See value definition above | 2 |

4.1.2 Object 605B_h, 685B_h, and 705B_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 slow down ramp is the deceleration value of the used mode of operations. The shutdown option code always has the value 0 as only this is supported.

VALUE DEFINITION

| Value | Definition |
|-------|--|
| 0 | Disable <i>drive function</i> (switch-off the drive power stage) |

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|----------------------|-------------|------------|
| 605B _h (motor 0) | Shutdown option code | Variable | UNSIGNED16 |
| 685B _h (motor 1) | | | |
| 705B _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|----------------------------|---------------|
| 00 _h | r0 | Refer to CiA402-3 | See value definition above | 0 |

4.1.3 Object 605C_h, 685C_h, and 705C_h: Disable Operation Option Code

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

VALUE DEFINITION

| Value | Definition |
|-------|---|
| 1 | Slow down with <i>slow down ramp</i> ; disable of the <i>drive function</i> |

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|-------------------------------|-------------|------------|
| 605C _h (motor 0) | Disable operation option code | Variable | UNSIGNED16 |
| 685C _h (motor 1) | | | |
| 705C _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|----------------------------|---------------|
| 00 _h | ro | Refer to CiA402-3 | See value definition above | 1 |

4.1.4 Object 605D_h, 685D_h, and 705D_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. The halt option code always has the value 1 as only this is supported.

VALUE DEFINITION

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

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|------------------|-------------|------------|
| 605D _h (motor 0) | Halt option code | Variable | UNSIGNED16 |
| 685D _h (motor 1) | | | |
| 705D _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|-----------------------------|---------------|
| 00 _h | r0 | Refer to CiA402-3 | See value definition above. | 1 |

4.1.5 Object 605E_h, 685E_h, and 705E_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 <i>quick stop ramp</i> |

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|---|----------------------------|-------------|------------|
| 605E _h (motor 0) 685E _h (motor 1) 705E _h (motor 2) | Fault reaction option code | Variable | UNSIGNED16 |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|----------------------------|---------------|
| 00 _h | r0 | Refer to CiA402-3 | See value definition above | 2 |

4.1.6 Object 6060_h, 6860_h, and 7060_h: Modes of Operation

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

VALUE DEFINITION

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

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

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|---|--------------------|-------------|-----------|
| 6060 _h (motor 0) 6860 _h (motor 1) 7060 _h (motor 2) | Modes of operation | Variable | SIGNED8 |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|----------------------------|---------------|
| 00 _h | rw | Refer to CiA402-3 | See value definition above | 0 |

4.1.7 Object 6061_h, 6861_h, and 7061_h: Modes of Operation Display

This object shows the operating mode that is set.

VALUE DEFINITION

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

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

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|---|----------------------------|-------------|-----------|
| 6061 _h (motor 0) 6861 _h (motor 1) 7061 _h (motor 2) | Modes of operation display | Variable | SIGNED8 |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|----------------------------|---------------|
| 00 _h | ro | Refer to CiA402-3 | See value definition above | 0 |

4.1.8 Object 606A_h, 686A_h, and 706A_h: Sensor Selection Code

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

VALUE DEFINITION

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

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|---|-----------------------|-------------|-----------|
| 606A _h (motor 0) 686A _h (motor 1) 706A _h (motor 2) | Sensor selection code | Variable | SIGNED16 |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|-----------------------------|---------------|
| 00 _h | rw | Refer to CiA402-3 | See value definition above. | -1 |

4.1.9 Object 6085_h, 6885_h, and 7085_h: Quick Stop Deceleration

These objects indicate the configured deceleration used to stop a motor when the *quick stop function* is activated and the *quick stop code object* (605A_h/685A_h/705A_h) is set to 2 [or 6]. The value is given in the same unit as *profile acceleration object* 6083_h. The units can be chosen with objects 208E_h/228E_h/248E_h, described in paragraph 0. The objects 208E_h/228E_h/248E_h read 0 when internal units are selected or 179 when user units (PPS/s) are selected.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|---|-------------------------|-------------|------------|
| 6085 _h (motor 0) 6885 _h (motor 1) 7085 _h (motor 2) | Quick stop deceleration | Variable | UNSIGNED32 |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|-------------|---------------|
| 00 _h | rw | Refer to CiA402-3 | UNSIGNED32 | 1000 |

4.1.10 Object 608F_h, 688F_h, and 708F_h: Position Encoder Resolution

These objects define the resolution of the encoder. The position encoder resolution is calculated by the following formula:

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

All values are dimensionless.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|---|--------------------|-------------|------------|
| 608F _h (motor 0) 688F _h (motor 1) 708F _h (motor 2) | Encoder resolution | Array | UNSIGNED32 |

ENTRY DESCRIPTION

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

4.1.11 Object 60C5_h, 68C5_h, and 70C5_h: Max Acceleration

This object indicates the configured maximal acceleration. It is used to limit the acceleration to an acceptable value in order to prevent the motor and the moved mechanics from being destroyed. It is given in internal or user specific units (depending on object 208C_h, paragraph 5.1.18, object 2087_h and object 2088_h, paragraph 5.1.16 and paragraph 5.1.17).

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|---|------------------|-------------|------------|
| 60C5 _h (motor 0) 68C5 _h (motor 1) 70C5 _h (motor 2) | Max acceleration | Variable | UNSIGNED32 |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|-------------|----------------------|
| 00 _h | ro | Refer to CiA402-3 | UNSIGNED32 | Depends on the units |

4.1.12 Object 60C6_h, 68C6_h, and 70C6_h: Max Deceleration

This object indicates the configured maximal deceleration. It is used to limit the acceleration to an acceptable value in order to prevent the motor and the moved mechanics from being destroyed. The max deceleration is given in internal or user specific units (depending on object 208C_h, paragraph 5.1.18, object 2087_h and object 2088_h, paragraph 5.1.16 and paragraph 5.1.17). Object 60C6_h has the same unit as the max acceleration object 60C5_h.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|------------------|-------------|------------|
| 60C6 _h (motor 0) | Max deceleration | Variable | UNSIGNED32 |
| 68C6 _h (motor 1) | | | |
| 70C6 _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|-------------|----------------------|
| 00 _h | ro | Refer to CiA402-3 | UNSIGNED32 | Depends on the units |

4.1.13 Object 60FD_h, 68FD_h, and 70FD_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.

VALUE DEFINITION

| Value | Definition |
|----------------|--------------|
| 0 _b | Switched off |
| 1 _b | Switched on |

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|----------------|-------------|------------|
| 60FD _h (motor 0) | Digital inputs | Variable | UNSIGNED32 |
| 68FD _h (motor 1) | | | |
| 70FD _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|-------------|------------------------|
| 00 _h | r0 | Refer to CiA402-3 | UNSIGNED32 | 0000 0000 _h |

4.1.14 Object 6502_h, 6D02_h, and 7502_h: Supported Drive Modes

00000025_h, which means that the following modes are provided by the drive:

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

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|-----------------------|-------------|------------|
| 6502 _h (motor 0) | Supported drive modes | Variable | UNSIGNED32 |
| 6D02 _h (motor 1) | | | |
| 7502 _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|-------------|-----------------------|
| 00 _h | r0 | Refer to CiA402-3 | See above | 00000025 _h |

4.2 Profile Position Mode

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

Please refer to objects 6060_h/6860_h/7060_h (section 4.1.6) for information about how to choose an operation mode. Object 6061_h/6861_h/7061_h (section 4.1.7) show the operation mode for each motor that is set.

4.2.1 Detailed object specifications

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

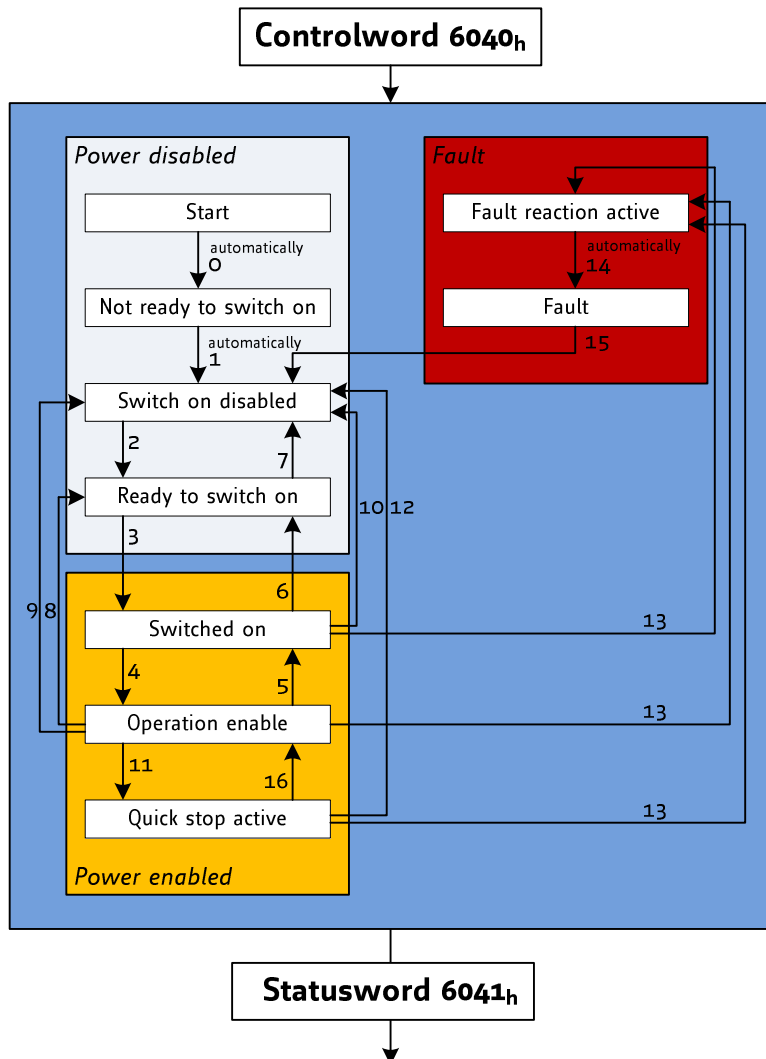


Figure 4.1 Finite state machine

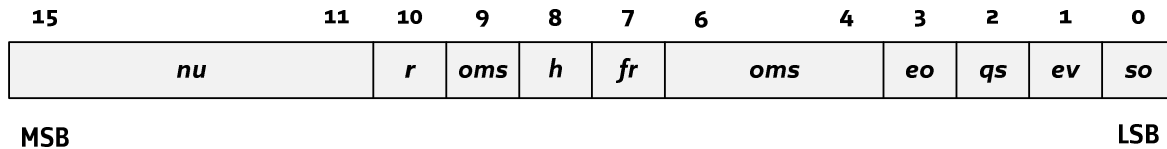
NOTES ON STATE TRANSITIONS

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

4.2.1.1 Object 6040_n, 6840_n, and 7040_n: Controlword

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

STRUCTURE OF CONTROLWORD / VALUE DEFINITION



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

OPERATION MODE SPECIFIC BITS IN PP MODE

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

COMMAND CODING

| Command | Bits of controlword | | | | | Transitions |
|------------------------------|---------------------|-------|-------|-------|-------|--------------|
| | Bit 7 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
| Shutdown | 0 | x | 1 | 1 | 0 | 2, 6, 8 |
| Switch on | 0 | 0 | 1 | 1 | 1 | 3 |
| Switch on + enable operation | 0 | 1 | 1 | 1 | 1 | 3 + 4 (NOTE) |
| Disable voltage | 0 | x | x | 0 | x | 7, 9, 10, 12 |
| Quick stop | 0 | x | 0 | 1 | x | 7, 10, 11 |
| Disable operation | 0 | 0 | 1 | 1 | 1 | 5 |
| Enable operation | 0 | 1 | 1 | 1 | 1 | 4, 16 |
| Fault reset | ↑ | x | x | x | x | 15 |

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|---|-------------|-------------|------------|
| 6040 _n (motor 0) 6840 _n (motor 1) 7040 _n (motor 2) | Controlword | Variable | UNSIGNED16 |

ENTRY DESCRIPTION

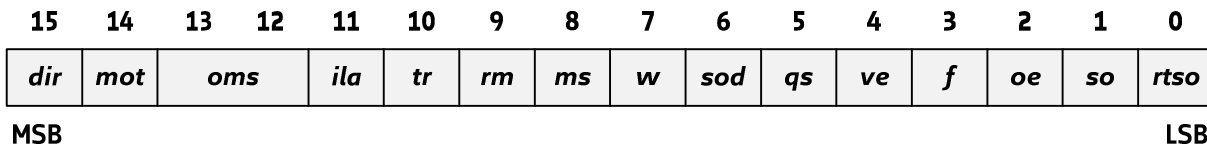
| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|--------------------------|---|
| 00 _n | rw | Refer to CiA402-3 | See command coding above | <i>Device and operation mode specific</i> |

4.2.1.2 Object 6041_h, 6841_h, and 7041_h: Statusword

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

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

STRUCTURE OF STATUSWORD / VALUE DEFINITION



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

TRINAMIC SPECIFIC BITS

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

OPERATION MODE SPECIFIC BITS IN PP MODE

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

STATE CODING

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

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|------------|-------------|------------|
| 6041 _h (motor 0) | Statusword | Variable | UNSIGNED16 |
| 6841 _h (motor 1) | | | |
| 7041 _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|------------------------|---------------|
| 00 _h | ro | Refer to CiA402-3 | See state coding above | no |

4.2.1.3 Object 6062_h, 6862_h, and 7062_h: Position Demand Value

These objects provide the demanded position value. The value is given in microsteps. The *position demand value* indicates the actual position that the motor should have. It is not to be confused with the *position actual internal value* (objects 6063_h/6863_h/7063_h) and *position actual value* (objects 6064_h/6864_h/7064_h).

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|---|-----------------------|-------------|-----------|
| 6062 _h (motor 0) 6862 _h (motor 1) 7062 _h (motor 2) | Position demand value | Variable | SIGNED32 |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|-------------|---------------|
| 00 _h | ro | Refer to CiA402-3 | SIGNED32 | no |

4.2.1.4 Object 6063_h, 6863_h, and 7063_h: Position Actual Internal Value

These objects provide the actual value of an encoder or a motor. Please use the *sensor selection objects* (see paragraph 4.1.8) for selecting the specific motor or the encoder first. The *position actual internal value* indicates the actual position of an encoder or a motor re-scaled to the microstep resolution. The value is given in microsteps.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|---|--------------------------------|-------------|-----------|
| 6063 _h (motor 0) 6863 _h (motor 1) 7063 _h (motor 2) | Position actual internal value | Variable | SIGNED32 |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|-------------|---------------|
| 00 _h | ro | Refer to CiA402-3 | SIGNED32 | no |

4.2.1.5 Object 6064_h, 6864_h, and 7064_h: Position Actual Value

These objects provide the actual value of a position measurement device. They always contain the same value as the related *position actual internal value* objects (6063_h/6863_h/7063_h).

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|---|-----------------------|-------------|-----------|
| 6064 _h (motor 0) 6864 _h (motor 1) 7064 _h (motor 2) | Position actual value | Variable | SIGNED32 |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|-------------|---------------|
| 00 _h | ro | Refer to CiA402-3 | SIGNED32 | no |

4.2.1.6 Object 6065_h, 6865_h, and 7065_h: Following Error Window

This object indicates the configured range of tolerated position values symmetrically to the position demand value. If the position actual value is out of the following error window, a following error occurs. A following error may occur when a drive is blocked, e.g., unreachable profile velocity occurs. The value is given in microsteps. If the value of the following error window is 0 or FFFF FFFF_h, the following control will be switched off.

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

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

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|------------------------|-------------|------------|
| 6065 _h (motor 0) | Following error window | Variable | UNSIGNED32 |
| 6865 _h (motor 1) | | | |
| 7065 _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|-------------|---------------|
| 00 _h | rw | Refer to CiA402-3 | UNSIGNED32 | 0... 8388607 |

4.2.1.7 Object 6067_h, 6867_h, and 7067_h: Position Window

This object indicates the configured symmetrical range of accepted positions relative to the target position. If the value of the actual position is within the position window, this target position is regarded as having been reached. The value is given in microsteps. If the value of the position window is FFFF FFFF_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 (607A_h).

When the position window is set to a value greater than zero, the target reached event will be signaled when the actual position value (6064_h) is within the (target position – position window) and (target position + position window).

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|-----------------|-------------|------------|
| 6067 _h (motor 0) | Position window | Variable | UNSIGNED32 |
| 6867 _h (motor 1) | | | |
| 7067 _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|-------------|---------------|
| 00 _h | rw | Refer to CiA402-3 | UNSIGNED32 | 0... 8388607 |

4.2.1.8 Object 6068_n, 6868_n, and 7068_n: 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_n) is set to a value greater than zero the target reached event will not be signaled until the actual position (6064_n) is at least as many milliseconds within the position window as defined by this object.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|----------------------|-------------|------------|
| 6068 _n (motor 0) | Position window time | Variable | UNSIGNED16 |
| 6868 _n (motor 1) | | | |
| 7068 _n (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|-------------|---------------|
| 00 _h | rw | Refer to CiA402-3 | UNSIGNED16 | 0 |

4.2.1.9 Object 606C_h, 686C_h, and 706C_h: Velocity Actual Value

These objects give the actual velocity value derived either from a velocity sensor or from a position sensor of the related motor.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-------------------|-----------------------|-------------|-----------|
| 606C _h | Velocity actual value | Variable | INTEGER32 |
| 686C _h | | | |
| 706C _h | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|-------------|---------------|
| 00 _h | ro | Refer to CiA402-3 | INTEGER32 | no |

4.2.1.10 Object 607A_h, 687A_h, and 707A_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 these objects is interpreted as absolute or relative depending on the *abs/rel* flag in the controlword. It is given in microsteps.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|-----------------|-------------|-----------|
| 607A _h (motor 0) | Target position | Variable | SIGNED32 |
| 687A _h (motor 1) | | | |
| 707A _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|-------------|----------------------------|
| 00 _h | rw | Refer to CiA402-3 | SIGNED32 | -2147483647... +2147483647 |

4.2.1.11 Object 607D_h, 687D_h, and 707D_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* (6062_h) and the *position actual value* (6064_h). 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* (607A_h), they are corrected internally by the *home offset* as follows:

Corrected min position limit = min position limit – home offset
 Corrected max position limit = max position limit – home offset

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

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|-------------------------|-------------|-----------|
| 607D _h (motor 0) | Software position limit | Array | SIGNED32 |
| 687D _h (motor 1) | | | |
| 707D _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Description | Entry category | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------------------|----------------|--------|-------------------|-------------|---------------|
| 01 _h | Min position limit | Mandatory | rw | Refer to CiA402-3 | SIGNED32 | -2147483647 |
| 02 _h | Max position limit | Mandatory | rw | Refer to CiA402-3 | SIGNED32 | +2147483647 |

4.2.1.12 Object 607F_h, 687F_h, and 707F_h: Maximum Profile Velocity

This object indicates the configured maximal allowed velocity in either direction during a profiled motion. It is the maximum velocity that can be used for positioning. The value is given in internal or user specific units (depending on object 208C_h, paragraph 5.1.18).

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|--------------------------|-------------|------------|
| 607F _h (motor 0) | Maximum profile velocity | Variable | UNSIGNED32 |
| 687F _h (motor 1) | | | |
| 707F _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|-------------|---------------|
| 00 _h | ro | Refer to CiA402-3 | UNSIGNED32 | 2047 |

4.2.1.13 Object 6081_h, 6881_h, and 7081_h: Profile Velocity

This object indicates the configured velocity normally attained at the end of the acceleration ramp during a profiled motion and is valid for both directions of motion. The profile velocity is the maximum velocity used when driving to a new position. It is given in internal or user specific units (depending on object 208C_h, paragraph 5.1.18)

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|------------------|-------------|------------|
| 6081 _h (motor 0) | Profile velocity | Variable | UNSIGNED32 |
| 6881 _h (motor 1) | | | |
| 7081 _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|-------------|---------------|
| 00 _h | rw | Refer to CiA402-3 | UNSIGNED32 | 1000 |

4.2.1.14 Object 6083_h, 6883_h, and 7083_h: Profile Acceleration

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

Please choose the units for object 6083_h with object 208E_h, described in paragraph 5.1.19. Object 208E_h reads 0 when internal units are selected or 179 when user units (PPS/s) are selected.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|----------------------|-------------|------------|
| 6083 _h (motor 0) | Profile acceleration | Variable | UNSIGNED32 |
| 6883 _h (motor 1) | | | |
| 7083 _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|-------------|---------------|
| 00 _h | rw | Refer to CiA402-3 | UNSIGNED32 | 1000 |

4.2.1.15 Object 6084_h, 6884_h, and 7084_h: Profile Deceleration

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

Please choose the units for object 6084_h with object 208E_h, described in paragraph 5.1.19. Object 208E_h reads 0 when internal units are selected or 179 when user units (PPS/s) are selected. The deceleration corresponds to the acceleration.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|----------------------|-------------|------------|
| 6084 _h (motor 0) | Profile deceleration | Variable | UNSIGNED32 |
| 6884 _h (motor 1) | | | |
| 7084 _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|-------------|---------------|
| 00 _h | rw | Refer to CiA402-3 | UNSIGNED32 | 1000 |

4.2.1.16 Object 6086_h, 6886_h, and 7086_h: Motion Profile Type

These objects indicate the configured type of motion profile used to perform a profiled motion. Set these objects to 0 for trapezoid ramps. Only sub-index 00_h is supported.

VALUE DEFINITION

| Value | Definition |
|-------|----------------------------------|
| 0 | Linear ramp, trapezoidal profile |

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|---------------------|-------------|------------|
| 6086 _h (motor 0) | Motion profile type | Variable | UNSIGNED16 |
| 6886 _h (motor 1) | | | |
| 7086 _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|-------------|---------------|
| 00 _h | rw | Refer to CiA402-3 | 0 | 0 |

4.2.2 How to move a motor in *pp* mode

Here is a little example that shows how to get a motor running in *pp* mode. In this little example we assume that the module has been reset (and then switched to 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).

4.3 Homing Mode

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

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

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

Figure 4.2 shows the defined input objects as well as the output objects. The user can specify the speeds, acceleration and method of homing. The home offset object 607C_h allows displacing zero in the coordinate system for the home position.

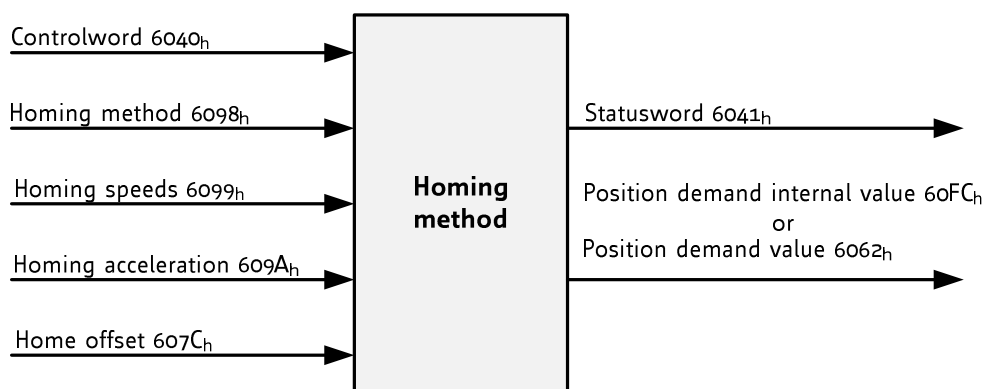


Figure 4.2 Homing mode function (motor 0)

BY CHOOSING A HOMING METHOD, THE FOLLOWING IS DETERMINED:

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

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

THERE ARE FOUR SOURCES OF HOMING SIGNAL AVAILABLE:

- negative and positive 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.

Please refer to objects 6060_h/6860_h/7060_h (section 4.1.6) for information about how to choose an operation mode. Objects 6061_h/6861_h/7061_h (section 4.1.7) show the operation modes that are set for each motor.

4.3.1 Connecting home switch, left switch, and right switch

First of all it is necessary to connect the switches. The tables and figures below will show you the right way.

4.3.1.1 TMCM-341/342/343

The 68-pin connector provides communication to a host, configuration of the EEPROM and connection of SPI drivers as well as connection of reference switches. Pin 1 of this connector is located in the lower left corner on the top site, while the connector is pointing towards the user.

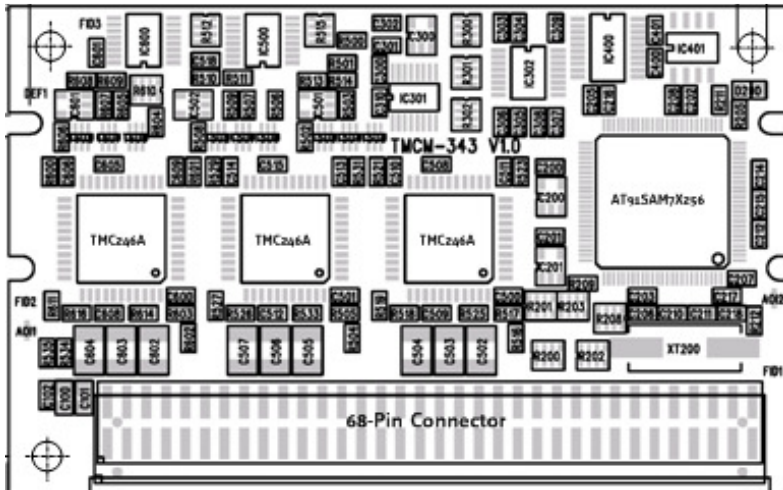


Figure 4.3: 68-pin connector of the TMCM-34x

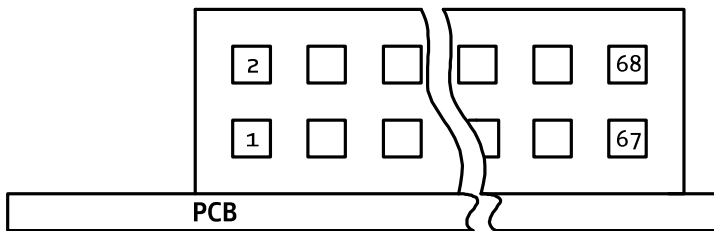


Figure 4.4: Pin order of the connector

Please connect home switch, left switch, and right switch with the power and I/O connector as follows:

CONNECTING SWITCHES

| Pin | Name | Function |
|-----|-------|-------------------------------------|
| 19 | REF_R | Right stop switch input for motor 0 |
| 21 | REF_L | Left stop switch input for motor 0 |
| 45 | HOME | Input 0 for motor 0 |
| 23 | REF_R | Right stop switch input for motor 1 |
| 25 | REF_L | Left stop switch input for motor 1 |
| 47 | HOME | Input 1 for motor 1 |
| 27 | REF_R | Right stop switch input for motor 2 |
| 29 | REF_L | Left stop switch input for motor 2 |
| 49 | HOME | Input 2 for motor 2 |

4.3.1.2 TMCM-351

For the reference switch inputs (two reference switch inputs are supported for each motor) there are two connector options: either one detachable screw connector (for prototyping, smaller series) or three separate crimp connectors (for higher volume series).

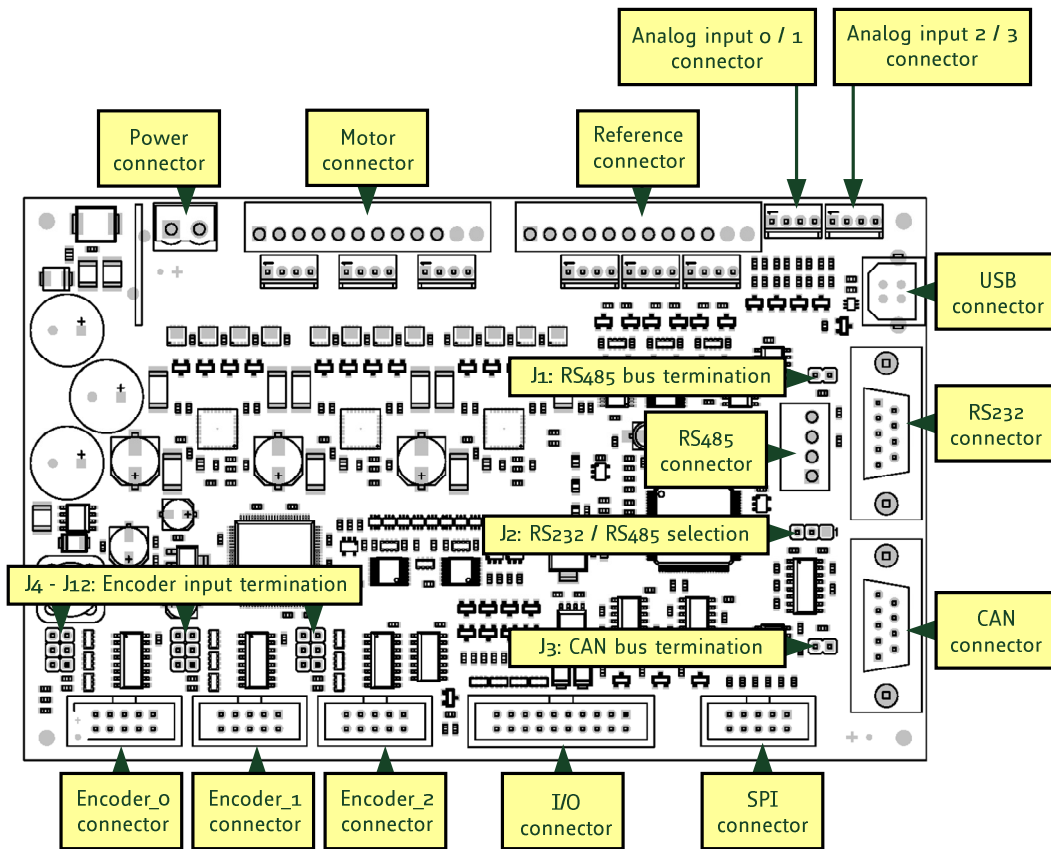


Figure 4.5: Connectors of TMCM-351

Please connect home switch, left switch, and right switch with the reference connector and the I/O connector as follows:

CONNECTING SWITCHES

| Pin | Name | Function |
|-----|---------|-------------------------|
| 1 | REF_0_R | Right stop switch input |
| 2 | REF_0_L | Left stop switch input |
| 11 | HOME_0 | Input 0, 24V compatible |
| 5 | REF_1_R | Right stop switch input |
| 6 | REF_1_L | Left stop switch input |
| 12 | HOME_1 | Input 1, 24V compatible |
| 9 | REF_2_R | Right stop switch input |
| 10 | REF_2_L | Left stop switch input |
| 13 | HOME_2 | Input 2, 24V compatible |

4.3.2 Connecting a brake

The modules provide the opportunity to connect a brake for each motor.

4.3.2.1 TMCM-341/342/343

Please connect the brakes with the power and I/O connector as follows:

CONNECTING BRAKES

| Pin | Name | Function |
|-----|-------|--------------------------------|
| 46 | OUT_0 | Connector for brake of motor 0 |
| 48 | OUT_1 | Connector for brake of motor 1 |
| 50 | OUT_2 | Connector for brake of motor 2 |

4.3.2.2 TMCM-351

Please connect the brakes with the I/O connector of the module as follows:

CONNECTING BRAKES

| Pin | Name | Function |
|-----|-------|--------------------------------|
| 1 | OUT_0 | Connector for brake of motor 0 |
| 2 | OUT_1 | Connector for brake of motor 1 |
| 3 | OUT_2 | Connector for brake of motor 2 |

4.3.3 Homing Methods

There are several different methods of homing. For choosing your homing method, refer to objects 6098_h/6898_h/7098_h (4.3.4.5).

OVERVIEW: HOMING METHODS

| Method no. | Description |
|------------|--|
| 0 | No homing. This is the default setting. |
| 1 | Search the left end switch, than search the next encoder index pulse. |
| 2 | Search the right end switch, than search the next encoder index pulse. |
| 3 | Search the positive edge of the home switch, than search the next encoder index pulse. |
| 5 | Search the negative edge of the home switch, than search the next encoder index pulse. |
| 17 | Search the left end switch. |
| 18 | Search the right end switch. |
| 19 | Search the positive edge of the home switch. |
| 21 | Search the negative edge of the home switch. |
| 33 | Search next index pulse in negative direction. |
| 34 | Search next index pulse in positive direction. |
| 35 | The actual position is used as home position. All position values (objects 6062 _h /6862 _h /7062 _h , 6063 _h /6863 _h /7063 _h , and 6064 _h /6864 _h /7064 _h) are set to zero, but the motor will not move. |

METHOD 1: HOMING ON NEGATIVE LIMIT SWITCH AND INDEX PULSE

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

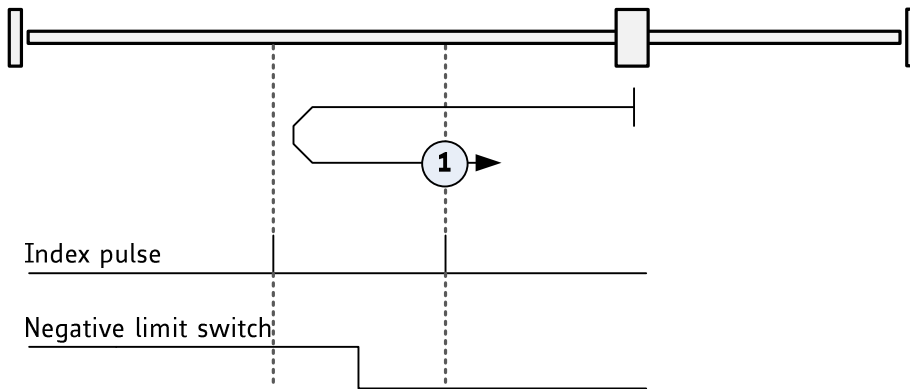


Figure 4.6 Homing on negative switch and index pulse

METHOD 2: HOMING ON POSITIVE LIMIT SWITCH AND INDEX PULSE

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

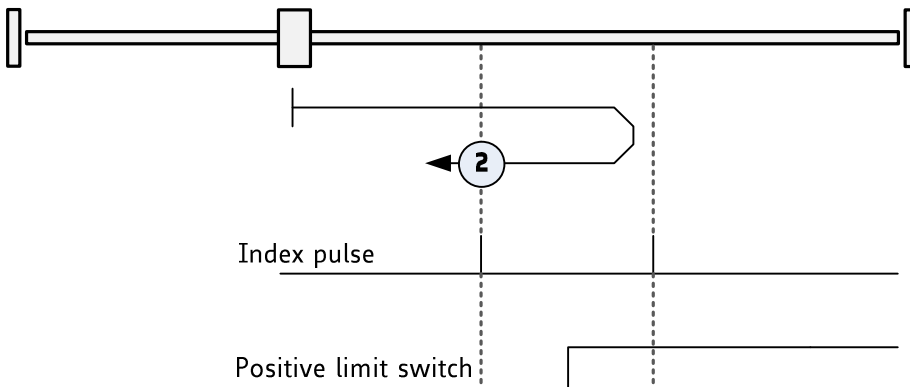


Figure 4.7 Homing on positive limit switch and index pulse

METHOD 3: HOMING ON POSITIVE HOME SWITCH AND INDEX PULSE

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

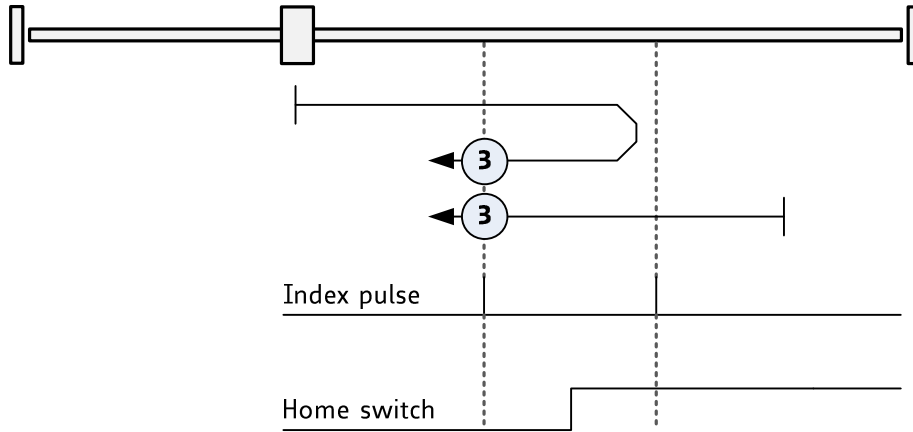


Figure 4.8 Homing on positive home switch and index pulse

METHOD 5: HOMING ON NEGATIVE HOME SWITCH AND INDEX PULSE

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

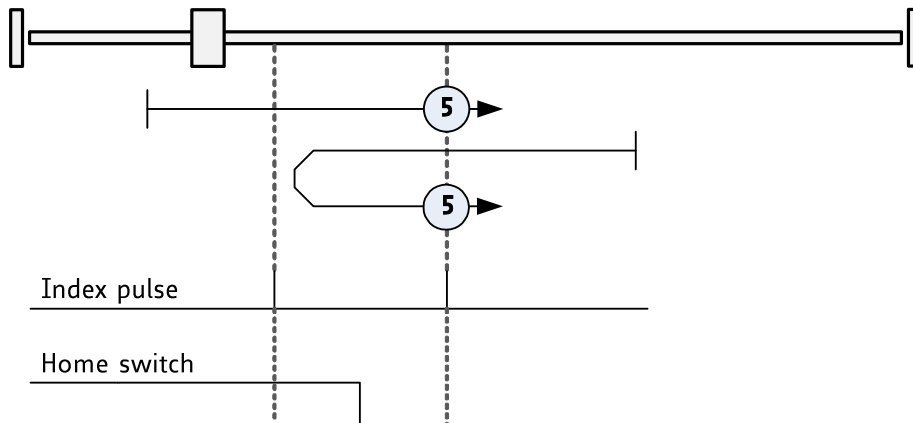


Figure 4.9 Homing on negative home switch and index pulse

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

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

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

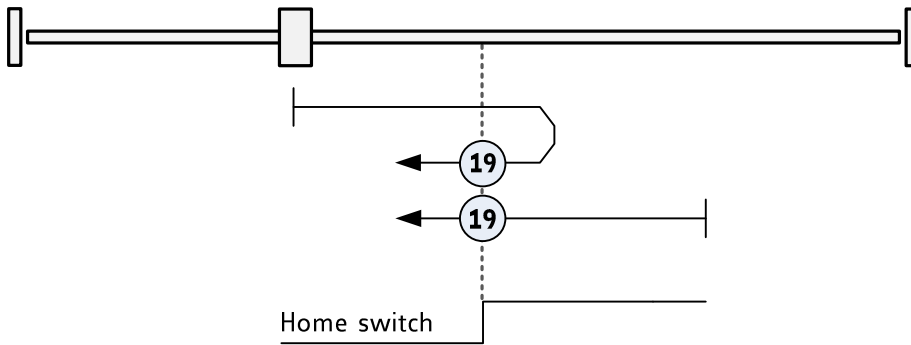


Figure 4.10 Homing without index pulse

METHOD 33 AND 34: HOMING ON NEXT INDEX PULSE

Using these methods, the direction of homing is negative or positive respectively. The home position shall be at the index pulse found in the selected direction as shown in Figure 4.11.

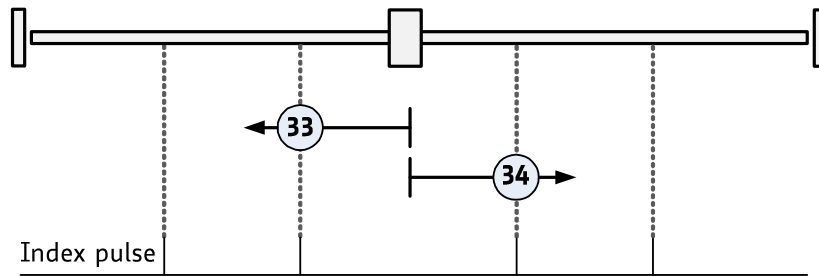


Figure 4.11 Homing on index pulse

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

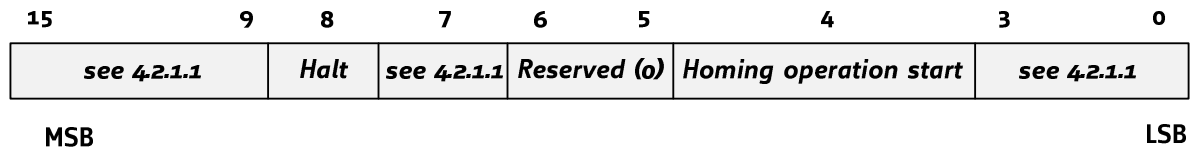
In this method, the current position shall be taken to be the home position. This method does not require the drive device to be in *operation enabled* state.

4.3.4 Detailed Object Definitions

4.3.4.1 Object 6040_h, 6840_h, and 7040_h: Controlword

These objects indicate the received commands controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using the controlword. Please refer to *Figure 4.1 Finite state machine* for detailed information.

STRUCTURE OF CONTROLWORD / VALUE DEFINITION



OPERATION MODE SPECIFIC BITS IN HM MODE

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

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|-------------|-------------|------------|
| 6040 _h (motor 0) | Controlword | Variable | UNSIGNED16 |
| 6840 _h (motor 1) | | | |
| 7040 _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|--------------------------|------------------------------------|
| 00 _h | rw | Refer to CiA402-3 | See command coding above | Device and operation mode specific |

4.3.4.2 Object 6041_h, 6841_h, and 7041_h: Statusword

These objects provide the status of the PDS FSA. They reflect the status of the CiA402 state machine for each motor axis. Please refer to *Figure 4.1 Finite state machine* for detailed information. The objects are structured as defined below.

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

STRUCTURE OF STATUSWORD / VALUE DEFINITION

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

OPERATION MODE SPECIFIC BITS IN HM MODE

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

DEFINITION OF BIT 10, BIT 12, AND BIT 13

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

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|------------|-------------|------------|
| 6041 _h (motor 0) | Statusword | Variable | UNSIGNED16 |
| 6841 _h (motor 1) | | | |
| 7041 _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|------------------------|---------------|
| 00 _h | ro | Refer to CiA402-3 | See state coding above | no |

4.3.4.3 Object 606C_h, 686C_h, and 706C_h: Velocity Actual Value

These objects give the actual velocity value for the related motor derived either from a velocity sensor or from a position sensor.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-------------------|-----------------------|-------------|-----------|
| 606C _h | Velocity actual value | Variable | INTEGER32 |
| 686C _h | | | |
| 706C _h | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|-------------|---------------|
| 00 _h | ro | Refer to CiA402-3 | INTEGER32 | no |

4.3.4.4 Object 607C_h, 687C_h, and 707C_h: Home Offset

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

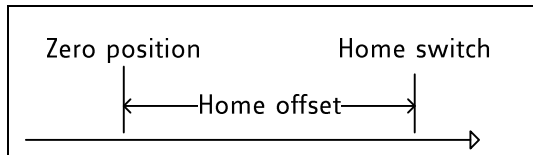


Figure 4.12 Home offset definition

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|-------------|-------------|-----------|
| 607C _h (motor 0) | Home offset | Variable | SIGNED32 |
| 687C _h (motor 1) | | | |
| 707C _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|-------------|----------------------------|
| 00 _h | rw | Refer to CiA402-3 | SIGNED32 | -2147483647... +2147483647 |

4.3.4.5 Object 6098_h, 6898_h, and 7098_h: Homing Method

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

VALUE DEFINITION

| Value | Definition |
|-------|--|
| 0 | No homing. This is the default value, but there is actually no homing method 0. |
| 1 | Search the left end switch; then search the next encoder index pulse. |
| 2 | Search the right end switch; then search the next encoder index pulse. |
| 3 | Search the positive edge of the home switch; then search the next encoder index pulse. |
| 5 | Search the negative edge of the home switch; then search the next encoder index pulse. |
| 17 | Search the left end switch. |
| 18 | Search the right end switch. |
| 19 | Search the positive edge of the home switch. |
| 21 | Search the negative edge of the home switch. |
| 33 | Search next index pulse in negative direction. |
| 34 | Search next index pulse in positive direction. |
| 35 | The actual position is used as home position. All position values (objects 6062 _h /6862 _h /7062 _h , 6063 _h /6863 _h /7063 _h , and 6064 _h /6864 _h /7064 _h) are set to zero, but the motor will not move. |

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|---------------|-------------|-----------|
| 6098 _h (motor 0) | Homing method | Variable | SIGNED8 |
| 6898 _h (motor 1) | | | |
| 7098 _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|---|---------------|
| 00 _h | rw | Refer to CiA402-3 | 0, 1, 2, 3, 5, 17, 18, 19, 21, 33, 34, 35 | 0 |

4.3.4.6 Object 6099_h, 6899_h, and 7099_h: Homing Speeds

These objects indicate the configured speeds used during homing procedure. The values are given in units, which can be selected with objects 208C_h/228C_h/248C_h, described in paragraph 5.1.18.

Using the *homing speeds* objects, 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*, described in paragraph 4.3.4.7) and then stopped at the exact switch point.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|---------------|-------------|------------|
| 6099 _h (motor 0) | Homing speeds | ARRAY | UNSIGNED32 |
| 6899 _h (motor 1) | | | |
| 7099 _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Description | Access | PDO Mapping | Value Range | Default Value |
|-----------------|-------------------|--------|-------------------|-------------|-----------------------------|
| 01 _h | Fast homing speed | rw | Refer to CiA402-3 | UNSIGNED32 | Depends on the unit setting |
| 02 _h | Slow homing speed | rw | Refer to CiA402-3 | UNSIGNED32 | Depends on the unit setting |

4.3.4.7 Object 609A_h, 689A_h, and 709A_h: Homing Acceleration

This object indicates the configured acceleration and deceleration to be used during homing operation. The values for accelerating to the fast homing speed and for decelerating to the slow homing speed can be set here. The values are given in units, which can be selected with object 208E_h/228E_h/248E_h, described in paragraph 0.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|---------------------|-------------|------------|
| 609A _h (motor 0) | Homing acceleration | Variable | UNSIGNED32 |
| 689A _h (motor 1) | | | |
| 709A _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range |
|-----------------|--------|-------------------|-------------|
| 00 _h | rw | Refer to CiA402-3 | UNSIGNED32 |

4.3.4.8 Object 2100_h: Home Offset Display

These objects show the configured difference between the zero position for the application and the machine home position/home switch (found during homing). The value is given in microsteps.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-------------------|---------------------|-------------|-----------|
| 2100 _h | Home offset display | Variable | SIGNED32 |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|----------------------|---------------|
| 00 _h | ro | no | Refer to home offset | 0 |

4.3.5 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. 1500 to object 6099_h sub index 1 and e.g. 1000 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.

4.4 Velocity Mode

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

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

- Velocity
- Acceleration
- Deceleration
- Emergency stop

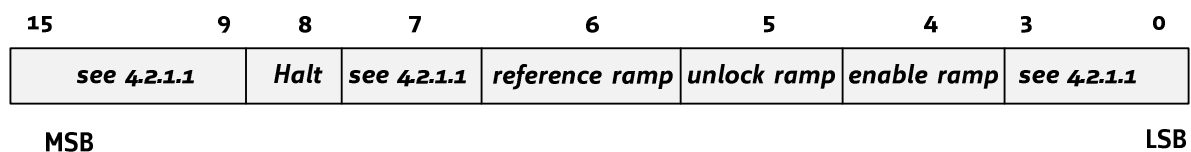
Please refer to object 6060_h (section 4.1.6) for information about how to choose an operation mode. Object 6061_h (section 4.1.7) shows the operation mode that is set.

4.4.1 Detailed object definitions

4.4.1.1 Object 6040_h, 6840_h, and 7040_h: Controlword

These objects indicate the received commands 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 4.1* Finite state machine for detailed information.

STRUCTURE OF CONTROLWORD / VALUE DEFINITION



There are no mode specific bits supported in vl mode.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|-------------|-------------|------------|
| 6040 _h (motor 0) | Controlword | Variable | UNSIGNED16 |
| 6840 _h (motor 1) | | | |
| 7040 _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|--------------------------|-------------------------|
| 00 _h | rw | Refer to CiA402-3 | See command coding above | Operation mode specific |

4.4.1.2 Object 6041_h, 6841_h, and 7041_h: Statusword

These objects provide the status of the PDS FSA. They reflect the status of the CiA402 state machine for each motor. Please refer to *Figure 4.1 Finite state machine* for detailed information. The objects are structured as defined below.

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

STRUCTURE OF STATUSWORD / VALUE DEFINITION

| | | | | | | | |
|--------------------|----|---------------------|----|--------------------|----|--------------------|------------|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 0 |
| <i>see 4.2.1.2</i> | | <i>reserved (0)</i> | | <i>see 4.2.1.2</i> | | <i>See 4.2.1.2</i> | |
| MSB | | | | | | | LSB |

There are no mode specific bits supported in vl mode.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|-------------|-------------|------------|
| 6041 _h (motor 0) | Controlword | Variable | UNSIGNED16 |
| 6841 _h (motor 1) | | | |
| 7041 _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|------------------------|---------------|
| 00 _h | ro | Refer to CiA402-3 | See state coding above | No |

4.4.1.3 Object 6042_h, 6842_h, and 7042_h: vl Target Velocity

This object indicates the required velocity of the system. It is the target velocity when the velocity mode is active. Use this object to control the velocity of the motor in velocity mode. When this object is changed the motor will be accelerated or decelerated to the new velocity.

The range depends on the chosen units, which can be internal or user-specific. Please refer to object 208C_h/228C_h/248C_h (paragraph 5.1.18) for more information about that.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|--------------------|-------------|-----------|
| 6042 _h (motor 0) | vl target velocity | Variable | SIGNED16 |
| 6842 _h (motor 1) | | | |
| 7042 _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|-------------|-----------------------|
| 00 _h | rw | Refer to CiA402-3 | SIGNED32 | Depends on the units. |

4.4.1.4 Object 6043_h, 6843_h, and 7043_h: vI Velocity Demand

The object provides the instantaneous velocity generated by the ramp function. It shows the last target velocity that has been set using object 6042_h, 6842_h, and 7042_h.

The value is given in the same unit as the target velocity. Please refer to object 208C_h/228C_h/248C_h (paragraph 5.1.18) for more information about that.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|--------------------|-------------|-----------|
| 6043 _h (motor 0) | vI velocity demand | Variable | SIGNED16 |
| 6843 _h (motor 1) | | | |
| 7043 _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|-------------|---------------|
| 00 _h | ro | Refer to CiA402-3 | SIGNED32 | 0 |

4.4.1.5 Object 6044_h, 6844_h, and 7044_h: vI Velocity Actual Value

This object shows the actual velocity of the motor when the motor is in velocity mode.

The value is given in the same unit as vI target velocity and vI velocity demand. Please refer to object 208C_h/228C_h/248C_h (paragraph 5.1.18) for more information about that.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|--------------------------|-------------|-----------|
| 6044 _h (motor 0) | vI velocity actual value | Variable | SIGNED16 |
| 6844 _h (motor 1) | | | |
| 7044 _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|-------------|---------------|
| 00 _h | ro | Refer to CiA402-3 | SIGNED32 | 0 |

4.4.1.6 Object 6046_h, 6846_h, and 7046_h: vI Velocity Min Max Amount

These objects indicate the configured minimum and maximum amounts of velocity in velocity mode for all three motors.

OBJECT DESCRIPTION

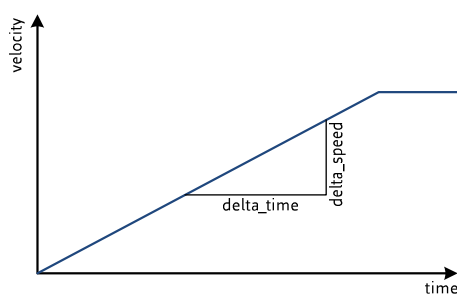
| Index | Name | Object Code | Data Type |
|-----------------------------|----------------------------|-------------|------------|
| 6046 _h (motor 0) | vI velocity min max amount | ARRAY | UNSIGNED32 |
| 6846 _h (motor 1) | | | |
| 7046 _h (motor 2) | | | |

ENTRY DESCRIPTION

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

4.4.1.7 Object 6048_n, 6848_n, and 7048_n: vI Velocity Acceleration

These objects indicate the configured delta speed and delta time of the slope of the acceleration ramp for each motor. The objects provide the accelerations used in velocity mode. These values are also used for the deceleration. The values of the delta speed are given in internal or in user units (rpm/s).



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

Figure 4.13 Transfer characteristic of the velocity acceleration

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|---|--------------------------|-------------|------------------------------|
| 6048 _n (motor 0) 6848 _n (motor 1) 7048 _n (motor 2) | vI velocity acceleration | Record | vI acceleration deceleration |

ENTRY DESCRIPTION

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

4.4.1.8 Object 6049_n, 6849_n, and 7049_n: vI Velocity Deceleration

These objects contain the same values as set in the objects 6048_n, 6848_n, and 7048_n. A separate deceleration value is not supported. The deceleration is according to the acceleration.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|---|--------------------------|-------------|------------------------------|
| 6049 _n (motor 0) 6849 _n (motor 1) 7049 _n (motor 2) | vI velocity deceleration | Record | vI acceleration deceleration |

ENTRY DESCRIPTION

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

4.4.2 How to Move a Motor in vI Mode

Here is a little example that shows how to get a motor running in vI mode. In this little example we assume that the module has been reset (and then switched to pre-operational or operationla) 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 vI mode by writing 2 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. 500) to object 6042_h. The motor now accelerates to that speed.
- Stop the motor by writing 0 to object 6042_h.

4.5 Profile Velocity Mode

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

THE PROFILE VELOCITY MODE COVERS THE FOLLOWING SUB-FUNCTIONS:

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

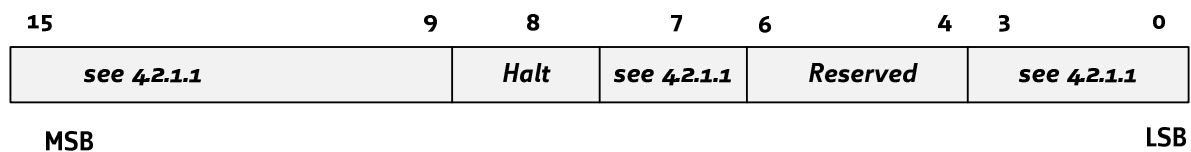
THE OPERATION OF THE REFERENCE VALUE GENERATOR AND ITS INPUT PARAMETERS INCLUDE:

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

4.5.1.1 Object 6040_n, 6840_n, and 7040_n: Controlword

These objects indicate the received commands controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using these objects. Please refer to Figure 4.1 for detailed information.

STRUCTURE OF CONTROLWORD / VALUE DEFINITION



There are no mode specific bits supported in this mode.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|-------------|-------------|------------|
| 6040 _n (motor 0) | Controlword | Variable | UNSIGNED16 |
| 6840 _n (motor 1) | | | |
| 7040 _n (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|--------------------------|------------------------------------|
| 00 _h | rw | Refer to CiA402-3 | See command coding above | Device and operation mode specific |

4.5.1.2 Object 6041_h, 6841_h, and 7041_h: Statusword

These objects provide the status of the PDS FSA. They reflect the status of the CiA402 state machine for each motor. Please refer to *Figure 4.1 Finite state machine* for detailed information. The objects are structured as defined below.

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

STRUCTURE OF STATUSWORD / VALUE DEFINITION

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

OPERATION MODE SPECIFIC BITS IN PV MODE

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

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|-------------|-------------|------------|
| 6041 _h (motor 0) | Controlword | Variable | UNSIGNED16 |
| 6841 _h (motor 1) | | | |
| 7041 _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|------------------------|---------------|
| 00 _h | ro | Refer to CiA402-3 | See state coding above | no |

4.5.1.3 Object 6062_h, 6862_h, and 7062_h: Position Demand Value

These objects provide the demanded position values. The values are given in microsteps. The *position demand value* objects indicate the actual positions that the motors should have. These objects are not to be confused with the *position actual internal value* (paragraph 4.5.1.4) and the *position actual value* (paragraph 4.5.1.5).

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|-----------------------|-------------|-----------|
| 6062 _h (motor 0) | Position demand value | Variable | SIGNED32 |
| 6862 _h (motor 1) | | | |
| 7062 _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|-------------|---------------|
| 00 _h | ro | Refer to CiA402-3 | SIGNED32 | no |

4.5.1.4 Object 6063_h, 6863_h, and 7063_h: Position Actual Internal Value

These objects indicate the actual position of the encoder of each motor, re-scaled to the microstep resolution. If necessary, the data unit may be transformed from user-defined units to increments. The value is given in microsteps or encoder steps, depending on objects 606A_h, 686A_h, and 706A_h (4.1.8).

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|---|--------------------------------|-------------|-----------|
| 6063 _h (motor 0) 6863 _h (motor 1) 7063 _h (motor 2) | Position actual internal value | Variable | SIGNED32 |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|-------------|---------------|
| 00 _h | ro | Refer to CiA402-3 | SIGNED32 | no |

4.5.1.5 Object 6064_h, 6864_h, and 7064_h: Position Actual Value

These objects provide the actual values of the position measurement devices. They always contain the same values as objects 6063_h, 6863_h, and 7063_h. The values are given in microsteps.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|---|-----------------------|-------------|-----------|
| 6064 _h (motor 0) 6864 _h (motor 1) 7064 _h (motor 2) | Position actual value | Variable | SIGNED32 |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|-------------|---------------|
| 00 _h | ro | Refer to CiA402-3 | SIGNED32 | no |

4.5.1.6 Object 6065_h, 6865_h, and 7065_h: Following Error Window

These objects indicate the configured ranges of tolerated position values symmetrically to the position demand values. If a position actual value is out of the following error window, a following error occurs. A following error may occur when a drive is blocked, unreachable profile velocity occurs, or at wrong closed-loop coefficients. The values are given in microsteps. If the value of the following error window is 0 or FFFF FFFF_h, the following control will be switched off.

If the difference between motor position (objects 6062_h, 6862_h, and 7062_h) and encoder position (objects 6063_h, 6863_h, and 7063_h or 6064_h, 6864_h, and 7064_h) is greater than the value set here, the related motors will be stopped and emergency messages will be sent.

Setting these objects to zero will turn off this feature completely. Setting these objects to a too low value will lead to false alarms.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|---|------------------------|-------------|------------|
| 6065 _h (motor 0) 6865 _h (motor 1) 7065 _h (motor 2) | Following error window | Variable | UNSIGNED32 |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|-------------|---------------|
| 00 _h | rw | Refer to CiA402-3 | UNSIGNED32 | 0... 8388607 |

4.5.1.7 Object 606C_h, 686C_h, and 706C_h: Velocity Actual Value

These objects give the actual velocity values derived either from the velocity sensors or from the position sensors of the related motors.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|-----------------------|-------------|-----------|
| 606C _h (motor 0) | Velocity actual value | Variable | INTEGER32 |
| 686C _h (motor 1) | | | |
| 706C _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|-------------|---------------|
| 00 _h | ro | Refer to CiA402-3 | INTEGER32 | no |

4.5.1.8 Object 607D_h, 687D_h, and 707D_h: Software Position Limit

These objects indicate the configured maximum and minimum software position limits for each axis. These parameters define the absolute position limits for the *position demand value* (4.5.1.3) and the *position actual value* (4.5.1.5). The *position actual value* is always checked against these limits. The limit position is always relative to the machine home position. Before being compared with the specific target position, they have to be corrected internally by the home offset as follows:

$$\begin{aligned} \text{Corrected min position limit} &= \text{min position limit} - \text{home offset} \\ \text{Corrected max position limit} &= \text{max position limit} - \text{home offset} \end{aligned}$$

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

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

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|-------------------------|-------------|-----------|
| 607D _h (motor 0) | Software position limit | Array | SIGNED32 |
| 687D _h (motor 1) | | | |
| 707D _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Description | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------------------|--------|-------------------|-------------|----------------------------|
| 01 _h | Min position limit | rw | Refer to CiA402-3 | SIGNED32 | -2147483647... +2147483647 |
| 02 _h | Max position limit | rw | Refer to CiA402-3 | SIGNED32 | -2147483647... +2147483647 |

4.5.1.9 Object 6083_h, 6883_h, and 7083_h: Profile Acceleration

These objects indicate the configured acceleration. The three *profile acceleration* objects set the maximum acceleration for each axis to be used in profile velocity mode.

Please choose the units for these objects with objects 208E_h/228E_h/248E_h, described in paragraph 0. The objects 208E_h/228E_h/248E_h read 0 when internal units are selected or 179 when user units (PPS/s) are selected.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|----------------------|-------------|------------|
| 6083 _h (motor 0) | Profile acceleration | Variable | UNSIGNED32 |
| 6883 _h (motor 1) | | | |
| 7083 _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|-------------|----------------------|
| 00 _h | rw | Refer to CiA402-3 | UNSIGNED32 | Depends on the units |

4.5.1.10 Object 6084_h, 6884_h, and 7084_h: Profile Deceleration

These objects indicate the configured deceleration. They set the maximum deceleration for each axis used in profile velocity mode.

Please choose the units for these objects with objects 208E_h/228E_h/248E_h, described in paragraph 5.1.19. The objects 208E_h/228E_h/248E_h read 0 when internal units are selected or 179 when user units (PPS/s) are selected. The deceleration is according to the acceleration.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|----------------------|-------------|------------|
| 6084 _h (motor 0) | Profile deceleration | Variable | UNSIGNED32 |
| 6884 _h (motor 1) | | | |
| 7084 _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------------|-------------|----------------------|
| 00 _h | rw | Refer to CiA402-3 | UNSIGNED32 | Depends on the units |

4.5.1.11 Object 60FF_h, 62FF_h, and 64FF_h: Target Velocity

The target velocity objects set the target velocity for the related motors when using profile velocity mode. They are used as input for the trajectory generator. The drive then accelerates or decelerates to that velocity using the acceleration and deceleration set by objects 6083_h/6883_h/7083_h (4.2.1.14) and 6084_h/6884_h/7084_h (4.2.1.15). The values are given in units, which can be selected with objects 208C_h/228C_h/248C_h, described in paragraph 5.1.18.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-------------------|-----------------|-------------|-----------|
| 60FF _h | Target velocity | Variable | SIGNED32 |
| 68FF _h | | | |
| 70FF _h | | | |

ENTRY DESCRIPTION

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

4.5.2 How to Move a Motor in *pv* Mode

Here is a little example that shows how to get a motor running in *pv* mode. In this little example we assume that the module has been reset (and then switched to 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. 1500) to object 60FF_h. The motor now accelerates to that speed.
- Stop the motor by writing 0 to object 60FF_h.

5 Manufacturer Specific Area

The manufacturer segment contains manufacturer specific objects. These objects control the special features of the TRINAMIC motion control devices TMC320-341/342/343 and TMC320-351.

5.1 Detailed Object Specifications

5.1.1 Object 2000_h, 2200_h, and 2400_h: Microstep Resolution

These objects set the microstep resolution for each axis. A value of 6 means 64 microsteps (2⁶). They are only writeable in the SWITCHED_ON_DISABLED state, but always readable.

VALUE DESCRIPTION

| Value | Description |
|-------|---------------|
| 0 | 1 step |
| 1 | 2 microsteps |
| 2 | 4 microsteps |
| 3 | 8 microsteps |
| 4 | 16 microsteps |
| 5 | 32 microsteps |
| 6 | 64 microsteps |

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|----------------------|-------------|-----------|
| 2000 _h (motor 0) | Microstep resolution | Variable | UNSIGNED8 |
| 2200 _h (motor 1) | | | |
| 2400 _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|-------------|---------------|
| 00 _h | rw | no | 0.. 8 | 6 |

5.1.2 Object 2001_h, 2201_h, and 2401_h: Fullstep Resolution

These objects show the fullstep resolutions of each axis.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|---------------------|-------------|------------|
| 2001 _h (motor 0) | Fullstep resolution | Variable | UNSIGNED16 |
| 2201 _h (motor 1) | | | |
| 2401 _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|-------------|---------------|
| 00 _h | ro | no | 200 | 200 |

5.1.3 Object 2002_h, 2202_h, and 2402_h: Brake Delay Times

With this object the delay times for applying and releasing an (optional) brake can be defined. Please see also object 200A_h, 220A_h, and 240A_h for an additional delay between enabling the power stage and releasing the brake. Both times are given in ms.

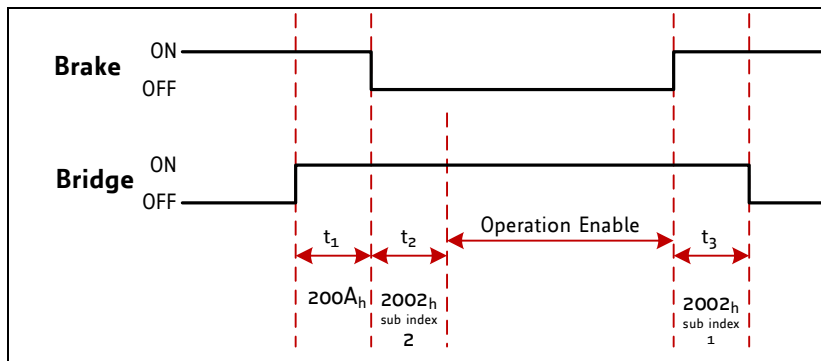


Figure 5.1 Brake output timing

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|-------------------|-------------|------------|
| 2002 _h (motor 0) | Brake delay times | ARRAY | UNSIGNED16 |
| 2202 _h (motor 1) | | | |
| 2402 _h (motor 2) | | | |

ENTRY DESCRIPTION

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

5.1.4 Object 2003_h, 2203_h, and 2403_h: Maximum Current

These objects define the currents used when the motors are moving. A value of 255 means 100% of the maximum current of a drive.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|-----------------|-------------|-----------|
| 2003 _h (motor 0) | Maximum current | Variable | UNSIGNED8 |
| 2203 _h (motor 1) | | | |
| 2403 _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|-------------|---------------|
| 00 _h | rw | no | 0... 255 | 228 |

5.1.5 Object 2004_h, 2204_h, and 2404_h: Standby Current

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

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|-----------------|-------------|-----------|
| 2004 _h (motor 0) | Standby current | Variable | UNSIGNED8 |
| 2204 _h (motor 1) | | | |
| 2404 _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|-------------|---------------|
| 00 _h | rw | no | 0... 255 | 20 |

5.1.6 Object 2005_h, 2205_h, and 2405_h: Limit Switches

These objects define which limit switches are to be used for related specific motor axes. 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 objects can only be written when the drive is in the SWITCHED_ON_DISABLED state (but is always readable).

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

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

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|----------------|-------------|------------|
| 2005 _h (motor 0) | Limit switches | Variable | UNSIGNED32 |
| 2205 _h (motor 1) | | | |
| 2405 _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|-------------|---------------|
| 00 _h | rw | no | 0... 7 | 0 |

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

5.1.7 Object 2006_h, 2206_h, and 2406_h: Mixed Decay Threshold

This object defines the behavior of the mixed decay feature of the drive. When it is set to the value -1, fast decay is always used. When it is set to a value between 0... 2047, fast decay will only be used when the internal velocity is higher or equal to that value. When set to 2048, fast decay will never be used.

It is recommended to use -1. Set this value to 2048 if stallGuard is used.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|-----------------------|-------------|------------|
| 2006 _h (motor 0) | Mixed decay threshold | Variable | UNSIGNED32 |
| 2206 _h (motor 1) | | | |
| 2406 _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|-------------|---------------|
| 00 _h | rw | no | -1... 2048 | -1 |

5.1.8 Object 2007_h, 2207_h, and 2407_h: Stall Detection Threshold

This object defines if stallGuard is to be used. When set to 0, stallGuard is not used. If set to 1... 7 the motor will be stopped and an emergency message will be generated if the stall guard value is the same or higher than the value set here. Switch off mixed decay (2006_h) before using stallGuard.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|---------------------------|-------------|-----------|
| 2007 _h (motor 0) | Stall detection threshold | Variable | UNSIGNED8 |
| 2207 _h (motor 1) | | | |
| 2407 _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|-------------|---------------|
| 00 _h | rw | no | 0... 7 | 0 |

5.1.9 Object 2009_h, 2209_h, and 2409_h: Fullstep Threshold

If the internal velocity gets higher than the value defined here the motor is switched to fullstep operation. A value of 0 or 2048 turns off this feature. Normally only needed for getting more force in higher speeds.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|--------------------|-------------|-----------|
| 2009 _h (motor 0) | Fullstep threshold | Variable | UNSIGNED8 |
| 2209 _h (motor 1) | | | |
| 2409 _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|-------------|---------------|
| 00 _h | ro | no | 0... 2048 | 0 |

5.1.10 Object 200A_h, 220A_h, and 240A_h: Enable Drive Delay Time

This is an additional delay time (in milliseconds) between enabling the power stage and releasing the brake. It can be used to prevent the brake from being released too early (before the hold current in the motor has been reached). Please see also object 2002_h.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|-------------------------|-------------|------------|
| 200A _h (motor 0) | Enable drive delay time | Variable | UNSIGNED16 |
| 220A _h (motor 1) | | | |
| 240A _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|-------------|---------------|
| 00 _h | rw | no | 0... 65535 | 0 |

5.1.11 Object 200B_h, 220B_h, and 240B_h: Encoder parameters

These objects define the polarity of the encoder null channel. They are only writable in SWITCHED_ON_DISABLED state. With the TMCM-351, TMCM-341, TMCM-342, and TMCM-343 only sub-index 03_h can be used.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|--------------------|-------------|-----------|
| 200B _h (motor 0) | Encoder parameters | ARRAY | UNSIGNED8 |
| 220B _h (motor 1) | | | |
| 240B _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Description | Access | PDO Mapping | Value Range | Default Value |
|-----------------|-----------------------|--------|-------------|-------------|---------------|
| 01 _h | Null channel polarity | rw | no | UNSIGNED8 | 7 |

This object defines the polarity of the encoder null channel. The following options can be selected:

Bit 0: polarity of channel A on index pulse (0=low, 1=high).

Bit 1: polarity of channel B on index pulse (0=low, 1=high).

Bit 2: polarity of channel N on index pulse (0=low, 1=high).

Bit 3: ignore channel A/B polarity on index pulse.

This object is only writable in SWITCHED_ON_DISABLED state.

5.1.12 Object 200Ch, 220Ch, and 240Ch: Brake Current Feed

This object configures how much current has to be fed into the brake to apply and to release it. 0 means 0%, 255 means 100% of the maximum current (this depends on the module). In most cases it is needed to feed current into the brake to release it. The default configuration is made for this case.

Setting both values to 0 disables the automatic brake control. This object is only writable in SWITCHED_ON_DISABLED state.

Please note that on the three axis modules only 0 (brake output switched off) and 255 (brake output switched on) can be used. Set both values to zero if no brake is used. In such a case the brake output can be used as a normal digital output.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------|--------------------|-------------|-----------|
| 200Ch (motor 0) | Brake current feed | ARRAY | UNSIGNED8 |
| 220Ch (motor 1) | | | |
| 240Ch (motor 2) | | | |

ENTRY DESCRIPTION

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

5.1.13 Object 204Eh, 224Eh, and 244Eh: Boost Current

These objects are used to set the current used for acceleration and deceleration phases for each motor. If set to 0 the current set with objects 2003h, 2203h, and 2403h (5.1.4) will be used. The value 1 can be chosen for a minimum boost current setting. A value of 255 means 100% of the maximum possible current of the module.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------|---------------|-------------|-----------|
| 204Eh (motor 0) | Boost current | Variable | UNSIGNED8 |
| 224Eh (motor 1) | | | |
| 244Eh (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------|--------|-------------|-------------|---------------|
| 00h | rw | no | 0... 255 | 0 |

5.1.14 Object 2085h, 2285h, and 2485h: Ramp Divisor

Use these objects to define the ramp divisor for each axis when internal units are selected for velocity and acceleration. These objects can only be written in the SWITCHED_ON_DISABLE state (but is always readable). The exponent of the scaling factor for a ramp generator- should be de/incremented carefully (in steps of one).

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------|--------------|-------------|-----------|
| 2085h (motor 0) | Ramp divisor | Variable | UNSIGNED8 |
| 2285h (motor 1) | | | |
| 2485h (motor 2) | | | |

ENTRY DESCRIPTION

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

5.1.15 Object 2086_h, 2286_h, and 2486_h: Pulse Divisor

Use these objects to define the pulse divisor for each axis when internal units are to be used for velocity and acceleration. These objects can only be written in the SWITCHED_ON_DISABLED state (but is always readable). The exponent of the scaling factor for a pulse (step) generator – should be de/incremented carefully (in steps of one).

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|---|---------------|-------------|-----------|
| 2086 _h (motor 0) 2286 _h (motor 1) 2486 _h (motor 2) | Pulse divisor | Variable | UNSIGNED8 |

ENTRY DESCRIPTION

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

5.1.16 Object 2087_h, 2287_h, and 2487_h: Maximum Velocity

The functionality of these objects depends on the unit selection for each axis (refer to objects 208C_h/228C_h/248C_h in chapter 5.1.18):

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

This value can only be changed when the drive is in the SWITCHED_ON_DISABLED state.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|---|------------------|-------------|------------|
| 2087 _h (motor 0) 2287 _h (motor 1) 2487 _h (motor 2) | Maximum velocity | Variable | UNSIGNED32 |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|---|---------------|
| 00 _h | rw | no | 143... 293 (rpm) 499755 (pps) resp. 2047 (internal) | 2047 |

5.1.17 Object 2088_h, 2288_h, and 2488_h: Maximum Acceleration

The functionality of these objects also depends on the unit selection (refer to objects 208E_h/228E_h/248E_h in chapter 5.1.19):

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

This value can only be change when the drive is in the SWITCHED_ON_DISABLED state.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|----------------------|-------------|------------|
| 2088 _h (motor 0) | Maximum acceleration | Variable | UNSIGNED32 |
| 2288 _h (motor 1) | | | |
| 2488 _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|---|---------------|
| 00 _h | rw | no | 2183.. 4000 (rpm/s) 976085662 (pps/s) resp. 2047 (internal) | 2047 |

5.1.18 Object 208C_h, 228C_h, and 248C_h: Velocity Dimension Index

With these objects different units can be chosen:

- Writing 0 selects internal units for velocity and acceleration.
- Writing 164 sets RPM for velocity and RPM/s for acceleration.
- Writing 181 sets PPS for velocity and PPS/s for acceleration.

This can only be changed in SWITCHED_ON_DISABLED mode.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|--------------------------|-------------|-----------|
| 208C _h (motor 0) | Velocity dimension index | Variable | UNSIGNED8 |
| 228C _h (motor 1) | | | |
| 248C _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|-------------|---------------|
| 00 _h | rw | no | 0, 164, 181 | 0 |

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

5.1.19 Object 208E_h, 228E_h, and 248E_h: Acceleration Dimension Index

With these objects, the unit for acceleration can be read out for each axis. The units can be set using objects 208C_h/228C_h/248C_h (5.1.18).

The *acceleration dimension index* objects read 0 when internal units are selected. They read 177 when RPM/s is selected, and the value for PPS/s is 179 then.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|---|------------------------------|-------------|-----------|
| 208E _h (motor 0) 228E _h (motor 1) 248E _h (motor 2) | Acceleration dimension index | Variable | UNSIGNED8 |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|-------------|---------------|
| 00 _h | ro | no | 0, 177, 179 | 0 |

5.1.20 Object 2100_h, 2300_h, and 2500_h: Home Offset Display

These objects show the home offset (4.3.4.4). The value is given in microsteps.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|---|---------------------|-------------|-----------|
| 2100 _h (motor 0) 2300 _h (motor 1) 2500 _h (motor 2) | Home offset display | Variable | SIGNED32 |

ENTRY DESCRIPTION

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

5.1.21 Object 2101_h, 2301_h, and 2501_h: Actual stallGuard™ Load Value

This object shows the actual stallGuard load value. It is needed for configuring the stallGuard feature, to find the optimal velocity (please see the TMCL manual for more about using stallGuard.)

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|---|-------------------------------|-------------|-----------|
| 2101 _h (motor 0) 2301 _h (motor 1) 2501 _h (motor 2) | Actual stallGuard™ load value | Variable | UNSIGNED8 |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|-------------|---------------|
| 00 _h | ro | no | 0... 7 | 0 |

5.1.22 Object 2102_h, 2302_h, and 2502_h: Driver Error Flags

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

THERE ARE THE FOLLOWING ERROR BITS:

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

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|---|--------------------|-------------|-----------|
| 2102 _h (motor 0) 2302 _h (motor 1) 2502 _h (motor 2) | Driver error flags | Variable | UNSIGNED8 |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|-------------|---------------|
| 00 _h | ro | no | 0.. 255 | 0 |

5.1.23 Object 2103_h, 2303_h, and 2503_h: Pulse Divisor Display

The pulse divisor display shows the pulse divisor that has been calculated by the drive when user units are selected or that can be set using object 2086_h/2286_h/2486_h when internal units are selected.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|---|-----------------------|-------------|-----------|
| 2103 _h (motor 0) 2303 _h (motor 1) 2503 _h (motor 2) | Pulse divisor display | Variable | UNSIGNED8 |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|-------------|---------------|
| 00 _h | ro | no | 0.. 13 | 3 |

5.1.24 Object 2104_h, 2304_h, and 2504_h: Maximum Velocity Display

These objects show the values of the objects 2087_h/2287_h/2487_h (5.1.16).

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|---|--------------------------|-------------|------------|
| 2104 _h (motor 0) 2304 _h (motor 1) 2504 _h (motor 2) | Maximum velocity display | Variable | UNSIGNED32 |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|-------------|---------------|
| 00 _h | ro | no | 0.. 2047 | 2047 |

5.1.25 Object 2105_h, 2305_h, and 2505_h: Ramp Divisor Display

The ramp divisor display shows the ramp divisor that has been calculated by the drive when user units are selected or that can be set using objects 2085_h/2285_h/2485_h (5.1.14) when internal units are selected.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|----------------------|-------------|-----------|
| 2105 _h (motor 0) | Ramp divisor display | Variable | UNSIGNED8 |
| 2305 _h (motor 1) | | | |
| 2505 _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|-------------|---------------|
| 00 _h | ro | no | 0... 13 | 7 |

5.1.26 Object 2106_h, 2306_h, and 2506_h: Maximum Acceleration Display

These objects show the values of objects 2088_h/2288_h/2488_h (5.1.17).

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|------------------------------|-------------|------------|
| 2106 _h (motor 0) | Maximum acceleration display | Variable | UNSIGNED32 |
| 2306 _h (motor 1) | | | |
| 2506 _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|-------------|---------------|
| 00 _h | ro | no | 0... 2047 | 2047 |

5.1.27 Object 2107_h, 2307_h, and 2507_h: Microstep Resolution Display

These objects show the microstep resolution set by objects 2000_h/2200_h/2400_h (5.1.1) when internal units are selected or calculated when user units are selected.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-----------------------------|------------------------------|-------------|-----------|
| 2107 _h (motor 0) | Microstep resolution display | Variable | UNSIGNED8 |
| 2307 _h (motor 1) | | | |
| 2507 _h (motor 2) | | | |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|-------------|---------------|
| 00 _h | ro | no | 0... 8 | 6 |

5.1.28 Object 2700_h: TMCL™ Direct Communication

After writing the *make signature* 656b616d_h to this object the drive switches to TMCL mode. The drive can then only be controlled via TMCL commands written to the *OS command* object 1023/01_h

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-------------------|------------------------------|-------------|------------|
| 2700 _h | Microstep resolution display | Variable | UNSIGNED32 |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|---|---------------|
| 00 _h | ro | no | Make signature 656b616d _h | 0 |

The drive has to be reset (using the *DS-301 reset application command*) to return to normal CANopen functionality. Then, the parameters of the manufacturer specific profile area and of the standardized device profile area are set to their power-on values.

5.1.29 Object 2701_h: Manufacturer Specific Mode

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

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

THE MANUFACTURER SPECIFIC MODE HAS THE FOLLOWING FEATURES:

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

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-------------------|----------------------------|-------------|------------|
| 2701 _h | Manufacturer specific mode | Variable | UNSIGNED32 |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|-------------|---------------|
| 00 _h | rw | no | See textbox | 0 |

5.1.30 Object 2702_h: Digital Inputs

Bits 16... 23 of this object reflect the states of the digital inputs (general purpose inputs 0... 8, the number of available inputs depend on the module type).

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-------------------|----------------|-------------|------------|
| 2702 _h | Digital inputs | Variable | UNSIGNED32 |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|-------------|---------------|
| 00 _h | ro | no | - | 0 |

5.1.31 Object 2703_h: Digital Outputs

With this object the digital outputs (general purpose outputs) can be set. Bits 16... 23 of sub index 1 switch outputs 0... 8 of the module. Bits 16... 23 of sub index 2 determine which outputs can be switched. The number of available digital outputs depends on the module type.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-------------------|-----------------|-------------|-----------|
| 2703 _h | Digital outputs | Array | |

ENTRY DESCRIPTION

| Sub-Index | Description | Access | PDO Mapping | Value Range | Default Value |
|-----------------|------------------|--------|-------------|-------------|---------------|
| 01 _h | Physical outputs | rw | no | UNSIGNED32 | 0 |
| 02 _h | Output mask | rw | no | UNSIGNED32 | 0 |

5.1.32 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 2706_h. After that, reset the module. The new setting becomes active then.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-------------------|-------------------|-------------|------------|
| 2704 _h | CAN bit rate load | Variable | UNSIGNED16 |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|-----------------------------|---------------|
| 00 _h | rw | no | 20, 50, 125, 250, 500, 1000 | 250 |

HOW TO CHANGE THE CAN BIT RATE:

- Write new bit rate in object 2704_h.
- Write save signature 65766173_h in sub-index 01_h of object 2706_h.
- Reset module.

5.1.33 Object 2705_h: Node ID

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

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

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-------------------|--------------|-------------|-----------|
| 2705 _h | Node ID load | Variable | UNSIGNED8 |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|-------------|---------------|
| 00 _h | rw | no | 1... 127 | 1 |

HOW TO CHANGE THE NODE ID:

- Write new node ID in object 2705_h.
- Write save signature 65766173_h in sub-index 01_h of object 2706_h.
- Reset module.

5.1.34 Object 2706_h: Store

Writing the save signature to this object permanently saves changes made on objects 2707_h and 2708_h.

Save signature: 65766173_h

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-------------------|-------|-------------|------------|
| 2706 _h | Store | Variable | UNSIGNED32 |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|-----------------------------|---------------|
| 00 _h | ro | no | Highest sub-index supported | 1 |
| 01 _h | rw | no | Save signature | 0 |

5.1.35 Object 2707_h: CAN bit Rate Load

This object shows the selected CAN bit rate.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-------------------|--------------|-------------|------------|
| 2707 _h | CAN bit rate | Variable | UNSIGNED16 |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|----------------------------------|---------------|
| 00 _h | ro | no | 20, 50, 125, 250, 500, 800, 1000 | 1000 |

5.1.36 Object 2708_h: Node ID load

This object shows the selected node ID.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-------------------|---------|-------------|-----------|
| 2708 _h | Node ID | Variable | UNSIGNED8 |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|-------------|------------------------|
| 00 _h | ro | no | 1... 127 | Depends on ID switches |

5.1.37 Object 2709_h: Encoder Interface

This object defines whether the optional encoder interface (TMC343 module) is used with the module. Only the values 0 (no encoder interface) or 1 (use encoder interface) are possible. The value is automatically stored in the non volatile memory. After changing this object, a reset (*power cycle, reset pulse or reset application*) command is needed.

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-------------------|-------------------|-------------|-----------|
| 2709 _h | Encoder interface | Variable | UNSIGNED8 |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|-------------|---------------|
| 00 _h | rw | no | 0/1 | 0 |

5.1.38 Object 270A_h: Encoder Interface Display

This object provides the encoder interface setting that is used at the moment (0 or 1).

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-------------------|---------------------------|-------------|-----------|
| 270A _h | Encoder interface display | Variable | UNSIGNED8 |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|-------------|---------------|
| 00 _h | ro | no | 0/1 | 0 |

5.1.39 Object 270E_h: Device Analog Inputs

This object provides the values of the analog inputs of the device (assumed that the device is equipped with analogue inputs).

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

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-------------------|----------------------|-------------|------------|
| 270E _h | Device Analog Inputs | Variable | UNSIGNED32 |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|-------------------|---------------|
| 00 _h | ro | no | Highest sub-index | 8 |
| 01 _h | ro | no | 0..1023 | 0 |
| 02 _h | ro | no | 0..1023 | 0 |
| 03 _h | ro | no | 0..1023 | 0 |
| 04 _h | ro | no | 0..1023 | 0 |
| 05 _h | ro | no | 0..1023 | 0 |
| 06 _h | ro | no | 0..1023 | 0 |
| 07 _h | ro | no | 0..1023 | 0 |
| 08 _h | ro | no | 0..1023 | 0 |

5.1.40 Object 2710_h: Pull-up Resistors (TMCM-351 only)

This object switches the pull-up resistors of the inputs on or off (each bit switches the pull-up resistor of one general purpose input of the TMCM--351).

OBJECT DESCRIPTION

| Index | Name | Object Code | Data Type |
|-------------------|-------------------|-------------|-----------|
| 2710 _h | Pull-up resistors | Variable | UNSIGNED8 |

ENTRY DESCRIPTION

| Sub-Index | Access | PDO Mapping | Value Range | Default Value |
|-----------------|--------|-------------|-------------|---------------|
| 00 _h | rw | no | 0..255 | 0 |

6 Emergency Messages

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

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

ERROR CODES FOR TMC341/342/343 AND TMC351:

| Error code | Add. Byte | | | | | Description |
|------------|-----------|-------|---|---|---|--|
| | 1 | 2 | 3 | 4 | 5 | |
| 0x0000 | 0 | 0/1/2 | 0 | 0 | 0 | Fault reset The fault reset command has been executed. |
| 0x1000 | 1 | 0/1/2 | 0 | 0 | 0 | Generic error: open load bridge A The motor driver indicates open load on bridge A. It is possible that the motor cable is broken or that there is an error in the power amplifier itself. |
| 0x1000 | 2 | 0/1/2 | 0 | 0 | 0 | Generic error: open load bridge B The motor driver indicates open load on bridge B. It is possible that the motor cable is broken or that there is an error in the power amplifier itself. |
| 0x2310 | 0 | 0/1/2 | 0 | 0 | 0 | Overcurrent high side The motor driver indicates an overcurrent on the high side. This can be caused by a short circuit in the driver stage. |
| 0x2311 | 0 | 0/1/2 | 0 | 0 | 0 | Overcurrent bridge B The motor driver indicates that there is overcurrent on bridge B. This can be caused by a short circuit in the motor itself or in the motor driver stage. |
| 0x2312 | 0 | 0/1/2 | 0 | 0 | 0 | Overcurrent bridge A The motor driver indicates that there is overcurrent on bridge A. This can be caused by a short circuit in the motor itself or in the motor driver stage. |
| 0x3230 | 0 | 0/1/2 | 0 | 0 | 0 | stallGuard™ error The actual load value exceeds the stallGuard2 limit. |
| 0x4310 | 1 | 0/1/2 | 0 | 0 | 0 | Overtemperature pre-warning The temperature in the motor driver exceeds the pre-warning limit. |
| 0x4310 | 2 | 0/1/2 | 0 | 0 | 0 | Overtemperature error The motor driver has been switched off because the temperature limit has been exceeded. |
| 0x5441 | 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. |
| 0x6320 | 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. |
| 0x8110 | 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. |
| 0x81110 | 2 | 255 | 0 | 0 | 0 | CAN Tx buffer overflow The software CAN transmit buffer is full and thus some CAN messages are lost. |
| 0x81110 | 3 | 255 | 0 | 0 | 0 | CAN Rx buffer overflow The software CAN receive buffer is full and so some CAN messages are lost. |
| 0x8120 | 0 | 255 | 0 | 0 | 0 | CAN error passive The CAN controller has detected communication errors and has entered the CAN Error passive state. |
| 0x8140 | 0 | 255 | 0 | 0 | 0 | CAN controller recovered from bus-off state The CAN controller had detected too many errors and had changed into the bus-off state. The drive has been stopped and disabled. This message is sent after the CAN controller has recovered from bus-off state and is bus-on again. |
| 0x8611 | 0 | 0/1/2 | 0 | 0 | 0 | Following error The deviation between motor position counter and encoder position counter has exceeded the following error window. |
| 0xff00 | 0 | 0/1/2 | 0 | 0 | 0 | Undervoltage The supply voltage is too low to drive a motor. |
| 0xff01 | 1 | 0/1/2 | 0 | 0 | 0 | Positive software limit The actual position is outside the range defined by object 0x607d. |

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

7 Life Support Policy

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

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

© TRINAMIC Motion Control GmbH & Co. KG 2008-2014

Information given in this data sheet is believed to be accurate and reliable. However neither responsibility is assumed for the consequences of its use nor for any infringement of patents or other rights of third parties, which may result from its use.

Specifications are subject to change without notice.

All trademarks used are property of their respective owners.



8 Revision History

8.1 Firmware revision

| Version | Date | Author | Description |
|---------|-------------|--------|--|
| 2.50 | 2008-OCT-31 | OK | First release |
| 3.01 | 2009-OCT-12 | OK | Node ID and bitrate can be configured via manufacturer specific objects Limit switch polarity can be configured via object 2005 _h |
| 3.02 | 2010-JAN-07 | OK | Home switch polarity switchable (via object 2005 _h) |
| 3.05 | 2010-DEC-15 | OK | Heartbeat consumer functionality added Non volatile storage of some motor parameters added Analog inputs can be read via object 270E _h |
| 3.16 | 2013-APR-18 | OK | Not deployed. |
| 3.17 | | OK | Not deployed. |
| 3.18 | 25.04.2014 | OK | <i>Attention!</i> Per default, the encoder is not active. Set the sensor selection code (object 606A _h) to meet your needs. Profile velocity mode added. New or modified objects: <ul style="list-style-type: none"> - 100C_h guard time - 100D_h life time factor - 1010_h store parameters - 204E_h boost current - 2089_h standby delay - 2108_h absolute resolver value - 6098_h homing methods - 6098_h position encoder resolution - 6098_h encoder prescaler - 2005_h, 2205_h, 2405_h updated |
| 3.20 | 9.6.2017 | OK | Fixed save/restore function. New object 0x2710 (TMCM-351 only). |

Table 8.1 Firmware revisions

8.2 Document Revision

| Version | Date | Author | Description |
|---------|-------------|--------|---|
| 0.94 | 2008-OCT-24 | OK | Preliminary version |
| 0.95 | 2009-FEB-03 | OK | Digital Inputs and Outputs added |
| 0.97 | 2009-OCT-15 | OK | Homing methods corrected, objects 1014 _h / 1015 _h / 1017 _h corrected |
| 1.00 | 2010-MAR-31 | SD | Initial version |
| 1.01 | 2010-May-24 | SD | Objects 2704 _h , 2705 _h , 2706 _h , 2707 _h , 2708 _h corrected |
| 1.02 | 2010-JUN-15 | SD | Object 2706 _h and object 6098 _h corrected, minor changes |
| 1.03 | 2010-JUN-21 | SD | Value range of object 2005 _h corrected |
| 1.04 | 2011-AUG-01 | SD | New front, new objects: 1010 _h , 1011 _h , and 1016 _h |
| 2.00 | 2014-MAY-30 | SD | <p><i>Attention!</i> Per default, the encoder is not active. Set the sensor selection code (object 606A_h) to meet your needs.</p> <ul style="list-style-type: none"> - Profile velocity mode added. - Some object descriptions shifted because of new profile velocity mode. - New front design. - Features of modules delighted. Refer to hardware manual please. - 100C_h guard time new. - 100D_h life time factor new. - 605A_h quick stop option code updated. - 6060_h, 6860_h, 7060_h modes of operation updated. - 6061_h, 6861_h, 7061_h modes of operation display updated. - 1010_h (store parameters) and 1011_h (restore parameters) corrected. - 6502_h, 6D02_h, 7502_h supported drive modes updated. - New homing methods added. - 6098_h, 6898_h, 7098_h homing method updated. - 2100_h home offset display new. - 2000_h, 2200_h, 2400_h microstep resolution updated. - 2005_h, 2205_h, 2405_h limit switches updated. - 204E_h boost current new. - 2701_h manufacturer specific mode updated. - Information about features and order codes of the modules erased. Please refer to the hardware manuals. |
| 2.01 | 2017-JUL-26 | OK | Firmware revision 3.20 included. |

Table 8.2 Document revisions

9 References

Please refer to www.trinamic.com for the following manuals.

| | |
|-------------|--|
| [TMCM-341] | TMCM-341 Hardware Manual and TMCM-341 TMCL Firmware Manual |
| [TMCM-342] | TMCM-342 Hardware Manual and TMCM-342 TMCL Firmware Manual |
| [TMCM-343] | TMCM-343 Hardware Manual and TMCM-343 TMCL Firmware Manual |
| [TMCM-351] | TMCM-351 Hardware Manual and TMCM-351 TMCL Firmware Manual |
| [BB-301] | BB-301 Manual |
| [BB-301-S] | BB-301-S Manual |
| [BB-302] | BB-302 Manual |
| [BB-303] | BB-303 Manual |
| [BB-323] | BB-323 Manual |
| [BB-323-02] | BB-323-02 Manual |
| [BB-323-03] | BB-323-03 Manual |
| [TMCM-035] | TMCM-035 Manual |
| [TMCM-323] | TMCM-323 Manual |
| [TMCM-EVAL] | TMCM-EVAL Manual |
| [TMCM-023] | TMCM-023 Manual |
| [TMCM-078] | TMCM-078 Manual |
| [PD-013-42] | PD-013-42 Manual |
| [QSH5718] | QSH5718 Manual |
| [QSH6018] | QSH6018 Manual |