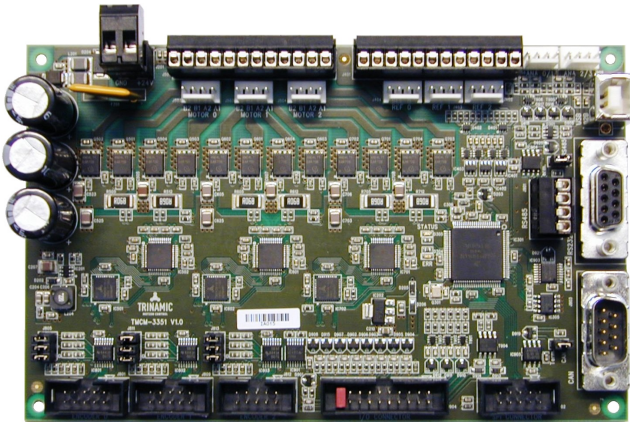


# TMCM-3351 CANopen® Firmware Manual

Firmware Version V3.22 | Document Revision V1.01 • 2019-FEB-14

The TMCM-3351 is a three axes controller/driver modules for 2-phase bipolar stepper motors with separate differential encoder and separate home and stop switch inputs for each axis. Dynamic current control, and quiet, smooth and efficient operation are combined with StealthChop™, Dc-Step™, StallGuard™ and CoolStep™ features. The module also offers eight digital inputs as well as four analogue inputs and eight digital outputs.



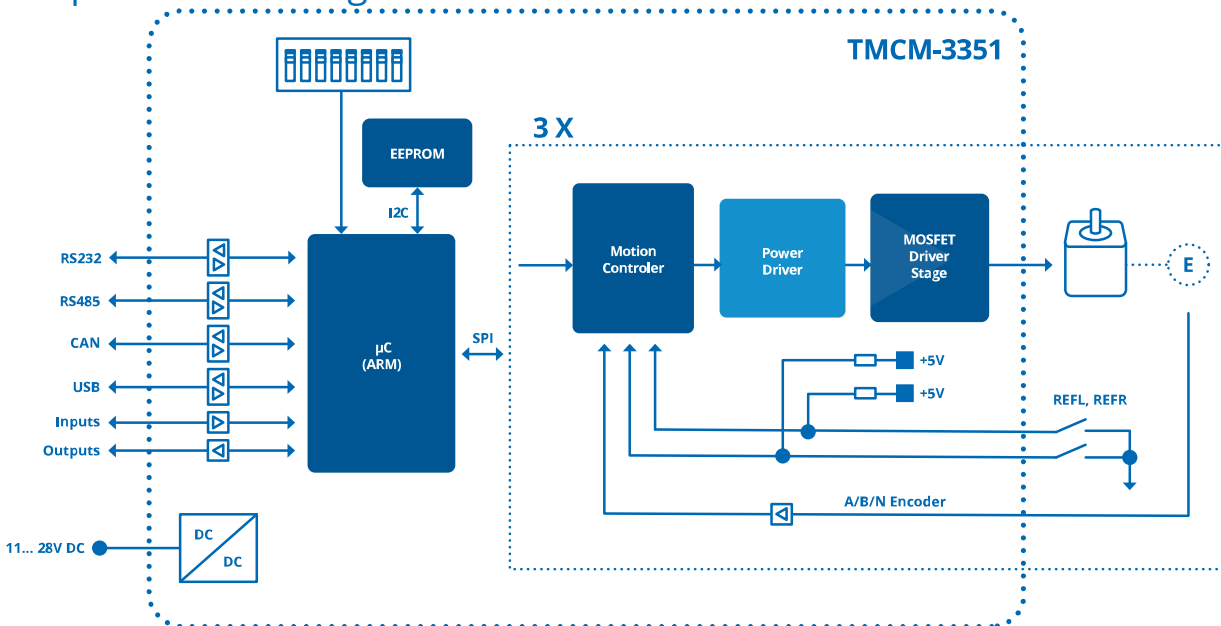
## Features

- 3-Axes stepper motor control
- CANopen® CiA-402 drive profile
- Linear ramps, SixPoint™ ramps and s-shaped ramps selectable
- Closed loop encoder support
- CoolStep™
- DcStep™
- StallGuard2™
- StealthChop™

## Applications

- Laboratory Automation
- Manufacturing
- Semiconductor Handling
- Robotics
- Factory Automation
- Test & Measurement
- Life Science
- Biotechnology
- Liquid Handling

## Simplified Block Diagram



©2021 TRINAMIC Motion Control GmbH & Co. KG, Hamburg, Germany  
 Terms of delivery and rights to technical change reserved.  
 Download newest version at: [www.trinamic.com](http://www.trinamic.com)



Read entire documentation.

# Contents

<b>1</b>	<b>Preface</b>	<b>7</b>
1.1	General Features of this CANopen Implementation	7
1.2	Abbreviations used in this Manual	8
1.3	Firmware Update	8
1.4	Trinamic's unique Features — easy to use with CANopen®	9
1.4.1	StallGuard2™	9
1.4.2	CoolStep™	9
1.5	Closed-Loop Operation	10
1.5.1	Closed-Loop Parameters	11
1.5.2	Load Angle Control	12
1.5.3	Current Level Control	13
1.5.4	Field Weakening	14
1.5.5	Position Catch up	14
1.6	PID Operation	15
1.6.1	PID Parameters	15
1.6.2	PID Modes	15
1.6.3	PID Example	16
<b>2</b>	<b>Communication</b>	<b>17</b>
2.1	Reference Model	17
2.2	NMT State Machine	19
2.3	Device Model	20
2.4	Object Dictionary	21
2.4.1	Object Indices on Multi-Axis Modules	22
<b>3</b>	<b>Communication Area</b>	<b>23</b>
3.1	Detailed Object Specifications	23
3.1.1	Object 1000 <sub>h</sub> ; Device Type	23
3.1.2	Object 1001 <sub>h</sub> ; Error Register	23
3.1.3	Object 1005 <sub>h</sub> ; COB-ID SYNC Message	24
3.1.4	Object 1008 <sub>h</sub> ; Manufacturer Device Name	25
3.1.5	Object 1009 <sub>h</sub> ; Manufacturer Hardware Version	25
3.1.6	Object 100A <sub>h</sub> ; Manufacturer Software Version	25
3.1.7	Object 100C <sub>h</sub> ; Guard Time	26
3.1.8	Object 100D <sub>h</sub> ; Life Time Factor	26
3.1.9	Object 1010 <sub>h</sub> ; Store Parameters	26
3.1.10	Object 1011 <sub>h</sub> ; Restore Parameters	28
3.1.11	Object 1014 <sub>h</sub> ; COB-ID Emergency Object	29
3.1.12	Object 1015 <sub>h</sub> ; Inhibit Time EMCY	29
3.1.13	Object 1016 <sub>h</sub> ; Consumer Heartbeat Time	30
3.1.14	Object 1017 <sub>h</sub> ; Producer Heartbeat Time	31
3.1.15	Object 1018 <sub>h</sub> ; Identity Object	31
3.1.16	Object 1023 <sub>h</sub> ; OS Command	32
3.1.17	Object 1029 <sub>h</sub> ; Error Behaviour	32
3.1.18	Objects 1400 <sub>h</sub> – 1403 <sub>h</sub> ; Receive PDO Communication Parameter	33
3.1.19	Objects 1600 <sub>h</sub> – 1603 <sub>h</sub> ; Receive PDO Mapping Parameter	34
3.1.20	Objects 1800 <sub>h</sub> – 1803 <sub>h</sub> ; Transmit PDO Communication Parameter	35
3.1.21	Objects 1A00 <sub>h</sub> – 1A03 <sub>h</sub> ; Transmit PDO Mapping Parameter	36
<b>4</b>	<b>Manufacturer specific Area</b>	<b>38</b>
4.1	Objects related to CoolStep™	38
4.2	Detailed Object Specifications	41



4.2.1	Object 2000 <sub>h</sub> : Microstep Resolution	41
4.2.2	Object 2001 <sub>h</sub> : Fullstep Resolution	41
4.2.3	Object 2002 <sub>h</sub> : Brake Delay Times	41
4.2.4	Object 2003 <sub>h</sub> : Maximum Current	42
4.2.5	Object 2004 <sub>h</sub> : Standby Current	43
4.2.6	Object 2005 <sub>h</sub> : Limit Switches	43
4.2.7	Object 200A <sub>h</sub> : Enable Drive Delay Time	44
4.2.8	Object 200B <sub>h</sub> : Encoder Parameters	44
4.2.9	Object 200C <sub>h</sub> : Brake Current Feed	45
4.2.10	Object 200F <sub>h</sub> : Encoder N Channel Latch	45
4.2.11	Object 2010 <sub>h</sub> : Profile Start Velocity	46
4.2.12	Object 2011 <sub>h</sub> : Profile Start Acceleration	46
4.2.13	Object 2012 <sub>h</sub> : Profile Break Velocity	47
4.2.14	Object 2013 <sub>h</sub> : Profile Final Deceleration	47
4.2.15	Object 2014 <sub>h</sub> : Profile Stop Deceleration	47
4.2.16	Object 2015 <sub>h</sub> : Bow Scaling Factor	48
4.2.17	Object 2020 <sub>h</sub> : Closed Loop Mode	48
4.2.18	Object 2021 <sub>h</sub> : Correction Position P	49
4.2.19	Object 2022 <sub>h</sub> : Maximum Correction Tolerance	49
4.2.20	Object 2027 <sub>h</sub> : Closed Loop Beta	49
4.2.21	Object 2028 <sub>h</sub> : Closed Loop Offset	50
4.2.22	Object 2029 <sub>h</sub> : Current Scaler Minimum	50
4.2.23	Object 202A <sub>h</sub> : Current Scaler Maximum	50
4.2.24	Object 202B <sub>h</sub> : Correction Velocity P	51
4.2.25	Object 202C <sub>h</sub> : Correction Velocity I	51
4.2.26	Object 202D <sub>h</sub> : Correction Velocity I Clipping	52
4.2.27	Object 202E <sub>h</sub> : Correction Velocity DV Clock	52
4.2.28	Object 202F <sub>h</sub> : Correction Velocity DV Clipping	53
4.2.29	Object 2030 <sub>h</sub> : Upscale Delay	53
4.2.30	Object 2031 <sub>h</sub> : Downscale Delay	53
4.2.31	Object 2033 <sub>h</sub> : Actual Scaling Factor	54
4.2.32	Object 2034 <sub>h</sub> : Field Weakening Minimum Velocity	54
4.2.33	Object 2035 <sub>h</sub> : Field Weakening Maximum Velocity	55
4.2.34	Object 2036 <sub>h</sub> : Field Weakening	55
4.2.35	Object 204E <sub>h</sub> : Boost Current	55
4.2.36	Object 2089 <sub>h</sub> : Setting Delay	56
4.2.37	Object 208C <sub>h</sub> : Velocity Dimension Index	56
4.2.38	Object 208E <sub>h</sub> : Acceleration Dimension Index	57
4.2.39	Object 2092 <sub>h</sub> : Chopper Blank Time	57
4.2.40	Object 2093 <sub>h</sub> : Chopper Mode	58
4.2.41	Object 2094 <sub>h</sub> : Chopper Hysteresis Decrement	58
4.2.42	Object 2095 <sub>h</sub> : Chopper Hysteresis End	59
4.2.43	Object 2096 <sub>h</sub> : Chopper Hysteresis Start	59
4.2.44	Object 2097 <sub>h</sub> : Chopper Off Time	59
4.2.45	Object 2098 <sub>h</sub> : Smart Energy Current Minimum	60
4.2.46	Object 2099 <sub>h</sub> : Smart Energy Current Down Step	60
4.2.47	Object 209A <sub>h</sub> : Smart Energy Hysteresis	61
4.2.48	Object 209B <sub>h</sub> : Smart Energy Current Up Step	61
4.2.49	Object 209C <sub>h</sub> : Smart Energy Hysteresis Start	62
4.2.50	Object 209D <sub>h</sub> : Smart Energy Filter Enable	62
4.2.51	Object 209E <sub>h</sub> : StallGuard2 Threshold	63
4.2.52	Object 20A1 <sub>h</sub> : Short Protection Disable	63
4.2.53	Object 20A3 <sub>h</sub> : Vsense	64
4.2.54	Object 20A4 <sub>h</sub> : Stop on Stall	64



4.2.55	Object 20A5 <sub>h</sub> : Smart Energy Threshold Speed	65
4.2.56	Object 2100 <sub>h</sub> : Home Offset Display	65
4.2.57	Object 2101 <sub>h</sub> : Actual Load Value	65
4.2.58	Object 2102 <sub>h</sub> : Driver Error Flags	66
4.2.59	Object 2107 <sub>h</sub> : Microstep Resolution Display	66
4.2.60	Object 210B <sub>h</sub> : Step Counter	67
4.2.61	Object 2120 <sub>h</sub> : Closed Loop Initialization Flag	67
4.2.62	Object 2700 <sub>h</sub> : TMCL Direct Communication	68
4.2.63	Object 2701 <sub>h</sub> : Manufacturer Specific Mode	68
4.2.64	Object 2702 <sub>h</sub> : Device Digital Inputs	69
4.2.65	Object 2703 <sub>h</sub> : Device Digital Outputs	69
4.2.66	Object 2704 <sub>h</sub> : CAN Bit Rate	70
4.2.67	Object 2705 <sub>h</sub> : Node ID	71
4.2.68	Object 2706 <sub>h</sub> : Store	72
4.2.69	Object 2707 <sub>h</sub> : CAN Bit Rate Load	72
4.2.70	Object 2708 <sub>h</sub> : Node ID Load	72
4.2.71	Object 270E <sub>h</sub> : Device Analog Inputs	73
<b>5</b>	<b>Profile specific Area</b>	<b>74</b>
5.1	Detailed Object Specifications	74
5.1.1	Object 605A <sub>h</sub> : Quick Stop Option Code	74
5.1.2	Object 605B <sub>h</sub> : Shutdown Option Code	75
5.1.3	Object 605C <sub>h</sub> : Disable Operation Option Code	76
5.1.4	Object 605D <sub>h</sub> : Halt Option Code	76
5.1.5	Object 605E <sub>h</sub> : Fault Reaction Option Code	77
5.1.6	Object 6060 <sub>h</sub> : Modes of Operation	77
5.1.7	Object 6061 <sub>h</sub> : Modes of Operation Display	78
5.1.8	Object 606A <sub>h</sub> : Sensor Selection Code	79
5.1.9	Object 608F <sub>h</sub> : Position Encoder Resolution	80
5.1.10	Object 60FD <sub>h</sub> : Digital Inputs	80
5.1.11	Object 6502 <sub>h</sub> : Supported Drive Modes	81
<b>6</b>	<b>Profile Position Mode</b>	<b>83</b>
6.1	Detailed Object Specifications	83
6.1.1	Object 6040 <sub>h</sub> : Control Word	84
6.1.2	Object 6041 <sub>h</sub> : Status Word	85
6.1.3	Object 6062 <sub>h</sub> : Position Demand Value	86
6.1.4	Object 6063 <sub>h</sub> : Position Actual Internal Value	87
6.1.5	Object 6064 <sub>h</sub> : Position Actual Value	87
6.1.6	Object 6065 <sub>h</sub> : Following Error Window	88
6.1.7	Object 6067 <sub>h</sub> : Position Window	88
6.1.8	Object 6068 <sub>h</sub> : Position Window Time	89
6.1.9	Object 606C <sub>h</sub> : Velocity Actual Value	89
6.1.10	Object 607A <sub>h</sub> : Target Position	90
6.1.11	Object 607D <sub>h</sub> : Software Position Limit	90
6.1.12	Object 6081 <sub>h</sub> : Profile Velocity	91
6.1.13	Object 6082 <sub>h</sub> : End Velocity	91
6.1.14	Object 6083 <sub>h</sub> : Profile Acceleration	92
6.1.15	Object 6084 <sub>h</sub> : Profile Deceleration	92
6.1.16	Object 6085 <sub>h</sub> : Quick Stop Deceleration	92
6.1.17	Object 6086 <sub>h</sub> : Motion Profile Type	93
6.1.18	Object 60A4 <sub>h</sub> : Profile Jerk	93
6.1.19	Object 60F2 <sub>h</sub> : Positioning Option Code	94
6.2	How to move a Motor in pp Mode	95



<b>7</b>	<b>Profile Velocity Mode</b>	<b>96</b>
7.1	Detailed Object Specifications . . . . .	96
7.1.1	Object 6040 <sub>h</sub> : Control Word . . . . .	96
7.1.2	Object 6041 <sub>h</sub> : Status Word . . . . .	97
7.1.3	Object 6062 <sub>h</sub> : Position Demand Value . . . . .	99
7.1.4	Object 6063 <sub>h</sub> : Position Actual Internal Value . . . . .	99
7.1.5	Object 6064 <sub>h</sub> : Position Actual Value . . . . .	100
7.1.6	Object 6065 <sub>h</sub> : Following Error Window . . . . .	100
7.1.7	Object 606C <sub>h</sub> : Velocity Actual Value . . . . .	101
7.1.8	Object 607D <sub>h</sub> : Software Position Limit . . . . .	101
7.1.9	Object 6083 <sub>h</sub> : Profile Acceleration . . . . .	102
7.1.10	Object 6084 <sub>h</sub> : Profile Deceleration . . . . .	102
7.1.11	Object 6085 <sub>h</sub> : Quick Stop Deceleration . . . . .	102
7.1.12	Object 6086 <sub>h</sub> : Motion Profile Type . . . . .	103
7.1.13	Object 60A4 <sub>h</sub> : Profile Jerk . . . . .	103
7.1.14	Object 60FF <sub>h</sub> : Target Velocity . . . . .	104
7.2	How to move a Motor in pv Mode . . . . .	105
<b>8</b>	<b>Homing Mode</b>	<b>106</b>
8.1	Homing Methods . . . . .	107
8.1.1	Homing Method 1: Homing on negative Limit Switch and Index Pulse . . . . .	107
8.1.2	Homing Method 2: Homing on positive Limit Switch and Index Pulse . . . . .	108
8.1.3	Homing Method 3: Homing on positive Home Switch and Index Pulse . . . . .	108
8.1.4	Homing Method 5: Homing on negative Home Switch and Index Pulse . . . . .	108
8.1.5	Homing Method 17: Homing on negative Limit Switch . . . . .	109
8.1.6	Homing Method 18: Homing on positive Limit Switch . . . . .	109
8.1.7	Homing Method 19: Homing on positive Home Switch . . . . .	110
8.1.8	Homing Method 21: Homing on negative Home Switch . . . . .	110
8.1.9	Homing Method 33 and 34: Homing on next Index Pulse . . . . .	110
8.1.10	Homing Method 35: Current Position as Home Position . . . . .	111
8.2	Detailed Object Specifications . . . . .	112
8.2.1	Object 6040 <sub>h</sub> : Control Word . . . . .	112
8.2.2	Object 6041 <sub>h</sub> : Status Word . . . . .	113
8.2.3	Object 606C <sub>h</sub> : Velocity Actual Value . . . . .	114
8.2.4	Object 607C <sub>h</sub> : Home Offset . . . . .	115
8.2.5	Object 6098 <sub>h</sub> : Homing Method . . . . .	116
8.2.6	Object 6099 <sub>h</sub> : Homing Speeds . . . . .	116
8.2.7	Object 609A <sub>h</sub> : Homing Acceleration . . . . .	116
8.3	How to start a Homing in hm Mode . . . . .	117
<b>9</b>	<b>Cyclic synchronous Position Mode</b>	<b>118</b>
9.1	Detailed Object Specifications . . . . .	118
9.1.1	Object 6040 <sub>h</sub> : Control Word . . . . .	118
9.1.2	Object 6041 <sub>h</sub> : Status Word . . . . .	119
9.1.3	Object 6062 <sub>h</sub> : Position Demand Value . . . . .	121
9.1.4	Object 6063 <sub>h</sub> : Position Actual Internal Value . . . . .	121
9.1.5	Object 6064 <sub>h</sub> : Position Actual Value . . . . .	122
9.1.6	Object 606C <sub>h</sub> : Velocity Actual Value . . . . .	122
9.1.7	Object 607A <sub>h</sub> : Target Position . . . . .	122
9.1.8	Object 607D <sub>h</sub> : Software Position Limit . . . . .	123
9.1.9	Object 60B0 <sub>h</sub> : Position Offset . . . . .	123
9.1.10	Object 60C2 <sub>h</sub> : Interpolation Time Period . . . . .	124
<b>10</b>	<b>Cyclic synchronous Velocity Mode</b>	<b>125</b>



10.1 Detailed Object Specifications . . . . .	125
10.1.1 Object 6040 <sub>h</sub> : Control Word . . . . .	125
10.1.2 Object 6041 <sub>h</sub> : Status Word . . . . .	126
10.1.3 Object 606C <sub>h</sub> : Velocity Actual Value . . . . .	128
10.1.4 Object 60FF <sub>h</sub> : Target Velocity . . . . .	128
10.1.5 Object 607D <sub>h</sub> : Software Position Limit . . . . .	128
10.1.6 Object 60B1 <sub>h</sub> : Velocity Offset . . . . .	129
10.1.7 Object 60C2 <sub>h</sub> : Interpolation Time Period . . . . .	129
<b>11 Cyclic synchronous Torque Mode</b>	<b>131</b>
11.1 Detailed Object Specifications . . . . .	131
11.1.1 Object 6040 <sub>h</sub> : Control Word . . . . .	131
11.1.2 Object 6041 <sub>h</sub> : Status Word . . . . .	132
11.1.3 Object 6062 <sub>h</sub> : Position Demand Value . . . . .	133
11.1.4 Object 6063 <sub>h</sub> : Position Actual Internal Value . . . . .	134
11.1.5 Object 6064 <sub>h</sub> : Position Actual Value . . . . .	134
11.1.6 Object 6071 <sub>h</sub> : Target Torque . . . . .	135
11.1.7 Object 6077 <sub>h</sub> : Torque actual Value . . . . .	135
11.1.8 Object 607D <sub>h</sub> : Software Position Limit . . . . .	135
11.1.9 Object 60B2 <sub>h</sub> : Torque Offset . . . . .	136
11.1.10 Object 60C2 <sub>h</sub> : Interpolation Time Period . . . . .	136
<b>12 Emergency Messages (EMCY)</b>	<b>138</b>
<b>13 Figures Index</b>	<b>141</b>
<b>14 Tables Index</b>	<b>142</b>
<b>15 Supplemental Directives</b>	<b>147</b>
15.1 Producer Information . . . . .	147
15.2 Copyright . . . . .	147
15.3 Trademark Designations and Symbols . . . . .	147
15.4 Target User . . . . .	147
15.5 Disclaimer: Life Support Systems . . . . .	147
15.6 Disclaimer: Intended Use . . . . .	147
15.7 Collateral Documents & Tools . . . . .	148
<b>16 Revision History</b>	<b>149</b>
16.1 Firmware Revision . . . . .	149
16.2 Document Revision . . . . .	149



# 1 Preface

This document specifies objects and modes of operation of the Trinamic TMCM-3351 stepper motor control module with CANopen firmware. The CANopen firmware is designed to fulfill the CANopen DS402 and DS301 standards. This manual assumes that the reader is already familiar with the basics of the CANopen protocol, defined by the DS301 and DS402 standards of the CAN-CiA. If necessary it is always possible to turn the module into a TMCL module by loading the TMCM-3351 TMCL firmware again through the USB interface, with the help of the firmware update function of the TMCL-IDE 3.0.

## 1.1 General Features of this CANopen Implementation

### Main Characteristics

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

### SDO Communication

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

### PDO Communication

- Producer
- Consumer
- RPDOs
  - Axis 0: 1, 2, 3, 4
  - Axis 1: 65, 66, 67, 68
  - Axis 2: 129, 130, 131, 132
  - Transmission modes: asynchronous.
  - Dynamic mapping with max. 3 mapping entries.
  - Default mappings: according to CiA-402 for first three PDOs of each axis, manufacturer specific for other PDOs of each axis.
- TPDOs
  - Axis 0: 1, 2, 3, 4
  - Axis 1: 65, 66, 67, 68
  - Axis 2: 129, 130, 131, 132
  - Transmission modes: asynchronous, asynchronous with event timer, synchronous.
  - Dynamic mapping with max. 3 mapping entries.
  - Default mappings: according to CiA-402 for first three PDOs of each axis, manufacturer specific for other PDOs of each axis.





## Further Characteristics

- SYNC: consumer (TPDOs 3, 67, 131 are synchronous PDOs)
- Emergency: producer
- RTR: supported only for node guarding/life guarding
- Heartbeat: consumer and producer

## 1.2 Abbreviations used in this Manual

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

Table 1: Abbreviations used in this Manual

## 1.3 Firmware Update

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





## 1.4 Trinamic’s unique Features — easy to use with CANopen®

### 1.4.1 StallGuard2™

StallGuard2™ is a high-precision sensorless load measurement using the back EMF of the coils. It can be used for stall detection as well as other uses at loads below those which stall the motor. The StallGuard2™ measurement value changes linearly over a wide range of load, velocity, and current settings. At maximum motor load, the value reaches zero or is near zero. This is the most energy-efficient point of operation for the motor.

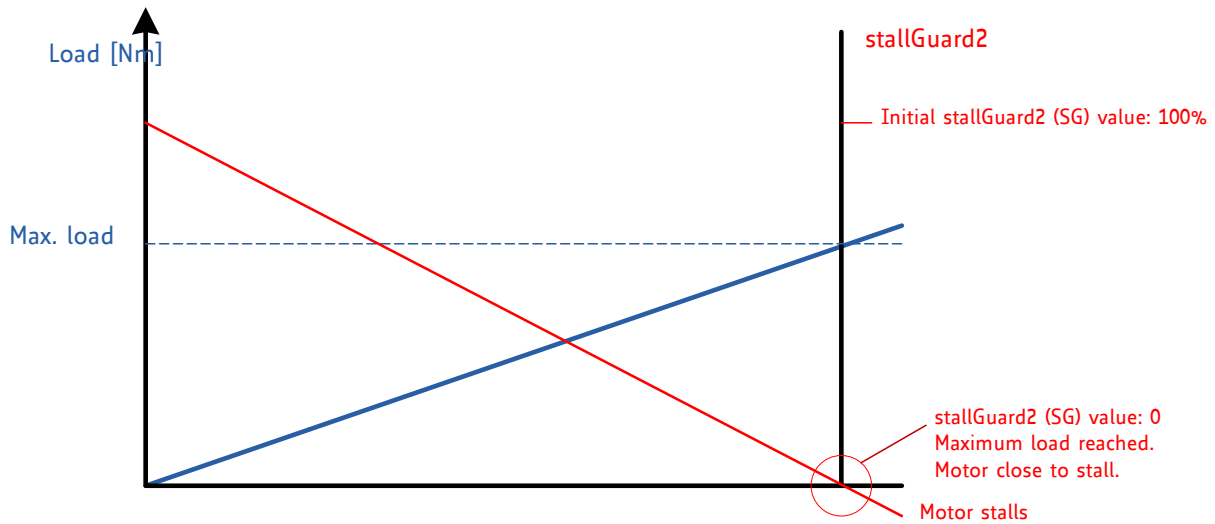


Figure 1: stallGuard2 Load Measurement as a Function of Load

### 1.4.2 CoolStep™

CoolStep™ is a load-adaptive automatic current scaling based on the load measurement via StallGuard2™ adapting the required current to the load. Energy consumption can be reduced by as much as 75%. CoolStep™ allows substantial energy savings, especially for motors which see varying loads or operate at a high duty cycle. Because a stepper motor application needs to work with a torque reserve of 30% to 50%, even a constant-load application allows significant energy savings because CoolStep™ automatically enables torque reserve when required. Reducing power consumption keeps the system cooler, increases motor life, and allows cost reduction.



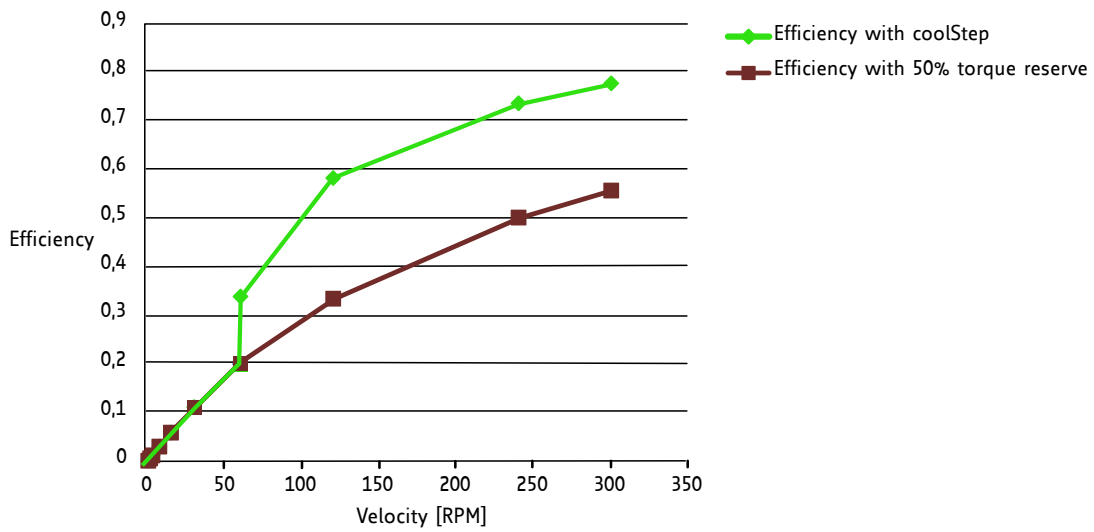


Figure 2: Energy Efficiency Example with CoolStep

### 1.5 Closed-Loop Operation

Together with an external ABN encoder it is possible to operate each axis of the TMCM-3351 as a closed-loop stepper system. Before enabling this feature, some parameters have to be set. The following table shows which objects should be set to which values in order to make closed-loop work. In this example we assume that a 1.8° motor is used together with a 40000cpr (10000lpr) encoder. Before the encoder can be used, the sensor selection code (object 606A<sub>h</sub>) has to be set to 0 (which means that a position encoder is to be used) and the resolution of the encoder ([cpr]) has to be written to object 608F<sub>h</sub> sub-index 1.



Closed-Loop Example Settings			
Parameter	Object	Value	Comment
Maximum current	2003 <sub>h</sub>	85	Set maximum motor current to 1 A.
Standby current	2004 <sub>h</sub>	10	Set standby current to 0.1 A.
Sensor selection code	606A <sub>h</sub>	0	Set to 0 in order to make encoder work. (default: -1 = no encoder)
Encoder resolution	608F <sub>h</sub> /1	10000	Set encoder resolution to 10000cpr.
Field weakening minimum velocity	2034 <sub>h</sub>	300000	Set gamma Vmin.
Field weakening maximum velocity	2035 <sub>h</sub>	1600000	Set gamma Vmax.
Field weakening	2036 <sub>h</sub>	255	(default value)
Closed loop beta	2027 <sub>h</sub>	255	Beta (default value)
Current scaler minimum	2029 <sub>h</sub>	50	
Current scaler maximum	202A <sub>h</sub>	255	
Maximum correction tolerance	2022 <sub>h</sub>	255	
Upscale delay	2030 <sub>h</sub>	1000	
Downscale delay	2031 <sub>h</sub>	10000	
Correction velocity P	202B <sub>h</sub>	3000	
Correction velocity I	202C <sub>h</sub>	20	
Correction velocity I clipping	202D <sub>h</sub>	2000	
Correction velocity DV clock	202E <sub>h</sub>	0	
Correction velocity DV clipping	202F <sub>h</sub>	100000	
Correction position P	2021 <sub>h</sub>	65536	

Table 2: Closed-Loop Example Settings

After these settings have been made, switch the state machine to OPERATIONAL (using the control word). Then, turn on closed-loop operation by setting object 2020<sub>h</sub> to 1. Now, read object 2120<sub>h</sub> until its value is 1 (closed-loop initialization finished).

#### Note

For closed loop mode to work, the encoder has to be mounted directly onto the motor shaft, without any gearing in between. The reason for this is that the encoder is also used for commutating the motor. If the encoder is not mounted directly to the motor shaft, closed loop mode cannot be used. Use the PID mode instead.

### 1.5.1 Closed-Loop Parameters

The closed-loop operation of the TMC3351 is based on Trinamic's closed-loop hardware motion controller IC TMC4361.

The 2-phase closed-loop control of the TMC3351 follows a different approach than PID control cascades to consider stepper motor driver characteristics. The ramp generator which assigns target and velocity is



independent of the position control (commutation angle control) which is also independent of the current control. The closed-loop control scheme is depicted in the following picture.

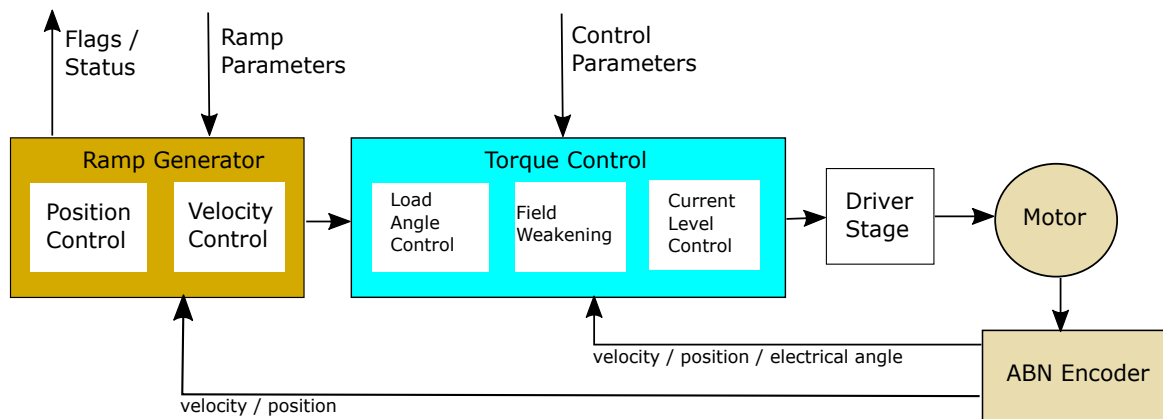


Figure 3: Closed-Loop Control Scheme

Load angle control and current level control will be executed in parallel.

### 1.5.2 Load Angle Control

As typical for stepper motor drivers, phase currents will be assigned directly to the motor drivers. This results in a current vector which should be followed by the rotor. The rotor position will be directly sampled by encoder feedback. The closed-loop motor control monitors the resulting load angle (deviation between driver stage current vector and encoder angle). Further on, the direction of the current vector will track the rotor position if the load angle should impend to exceed a certain limit. The result is a load angle which will never exceed the given limit and as a result no step loss will occur. Thus, the current vector will follow an overpowered load until the load is reduced.

Figure 4 shows the parameters which limit the load angle.



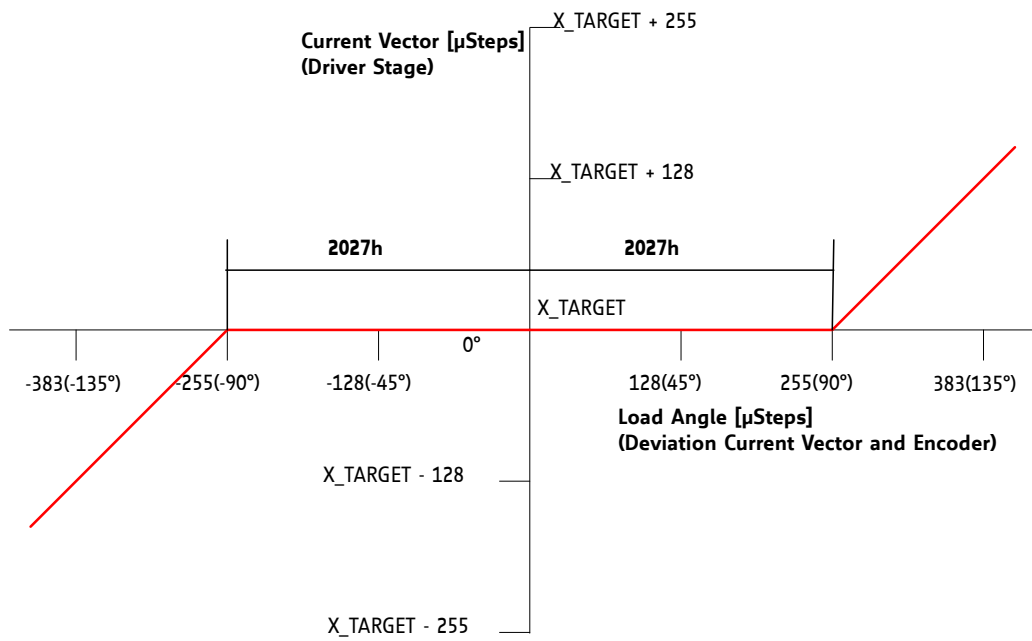


Figure 4: Load Angle Control Parameter

### 1.5.3 Current Level Control

Parallel to the load angle control the TMCM-3351 controls the motor current level (current vector amplitude) depending on the load angle to save energy during no or light load. Figure 5 gives an overview of the current control parameters.

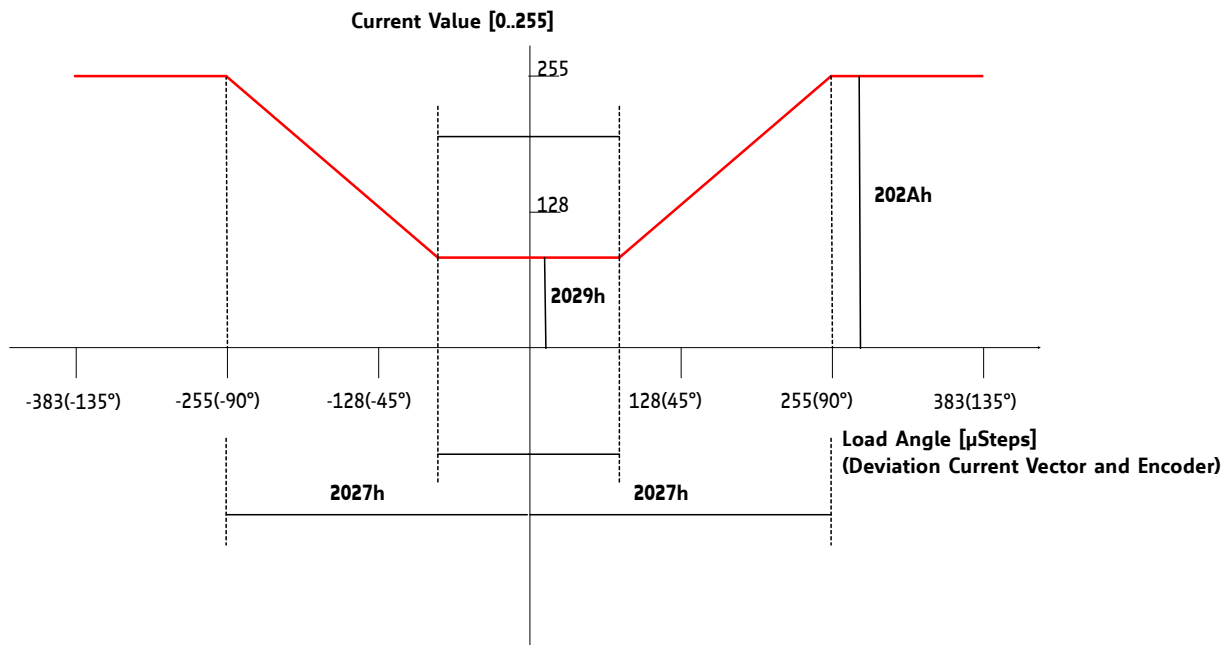


Figure 5: Current Level Control



- Object 2027<sub>h</sub>: Closed-loop beta.
- Object 2029<sub>h</sub>: Minimum closed-loop current scaler.
- Object 202A<sub>h</sub>: Maximum closed-loop current scaler.

Objects 2030<sub>h</sub> and 2031<sub>h</sub> set up the delay which defines how fast the actual current will be increased or decreased and will follow the red marked graph.

### 1.5.4 Field Weakening

With every stepper motor the TMCM-3351 will reach a velocity where it is not possible to maintain the target motor current due to the motor back EMF. Above this velocity load angle (2027<sub>h</sub>, default 90°) and current level control will reach their maximum. To drive the stepper motor faster the back EMF must be compensated by commutating the stepper motor with a commutation angle between 90° and 180°. The parameters for field weakening are described in figure 6.

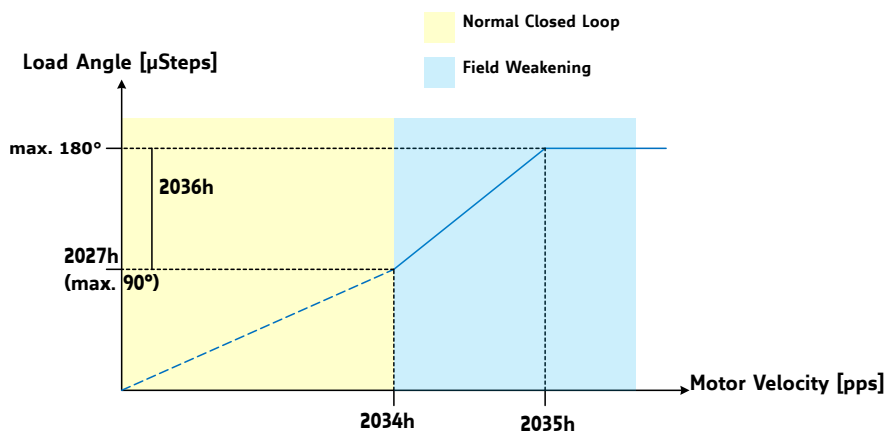


Figure 6: Field Weakening

- Object 2027<sub>h</sub>: Closed-loop beta.
- Object 2036<sub>h</sub>: Field weakening (closed-loop gamma).
- Object 2034<sub>h</sub>: Field weakening minimum velocity (gamma Vmin).
- Object 2035<sub>h</sub>: Field weakening maximum velocity (gamma Vmax).

### 1.5.5 Position Catch up

The TMCM-3351 includes a special feature for closed-loop positioning. Positioning parameters like velocity and acceleration will be calculated to reach a position in a dedicated time. If the target trapezoidal ramp cannot be maintained due to high load peaks the TMCM-3351 includes a special position catch-up mode to ensure that the position will still be reached in time if possible.



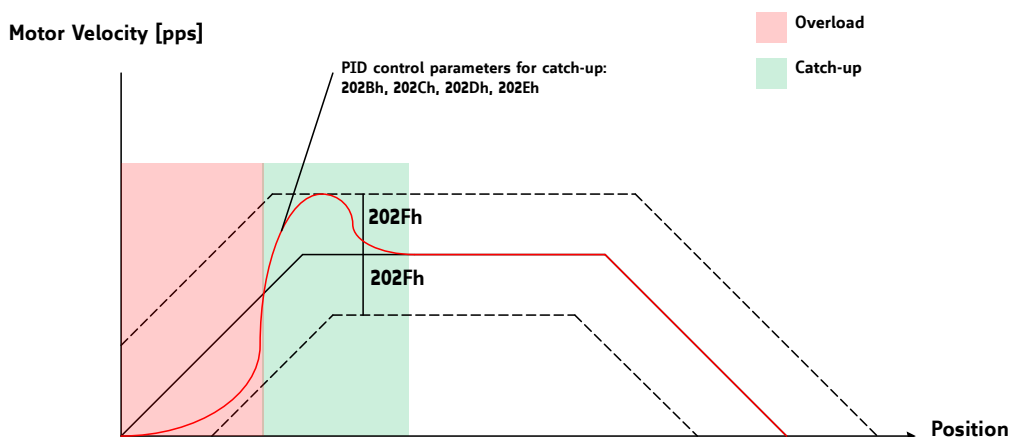


Figure 7: Position Catch up

## 1.6 PID Operation

Together with an external ABN encoder it is possible to operate each axis of the TMC3351 using PID position regulation. In contrast to closed loop mode, the encoder is only used for regulating the position, but not for commutating the motor. This also works when the encoder is not directly mounted to the motor shaft, i.e. with gearing between motor and encoder.

### 1.6.1 PID Parameters

When using PID mode, the following objects are used for setting the PID regulation parameters:

- Object 202B<sub>h</sub> defines the P part of the PID regulator.
- Object 202C<sub>h</sub> defines the I part of the PID regulator.
- Object 202D<sub>h</sub> defines the clipping value for the I part.
- Object 202E<sub>h</sub> sets the clock divider for D part calculation.
- Object 202F<sub>h</sub> defines the clipping value for the D part.
- Object 2021<sub>h</sub> defines the D part of the PID regulator.
- Object 2022<sub>h</sub> defines the maximum tolerated position deviation. There will be no correction as long as the position deviation is within this window.

### 1.6.2 PID Modes

There are two PID modes, which can be activated using object 2020<sub>h</sub>:

- PID mode based on  $v_{\text{actual}}$ : In this mode, the output value of the PID regulator will be added to the actual velocity given by the ramp generator. In most cases, this mode is used. Activate this PID mode by setting object 2020<sub>h</sub> to 2.
- PID mode based on  $v=0$ : In this mode, the output value of the PID regulator will be directly used as velocity value. This mode can be activated by setting 2020<sub>h</sub> to 3.





### 1.6.3 PID Example

The following table shows which values should be written to which objects in order to activate the PID mode. In this example we assume that a 1.8° motor is used together with a 40000cpr (10000lpr) encoder. Before the encoder can be used, the sensor selection code (object 606A<sub>h</sub>) has to be set to 0 (which means that a position encoder is to be used) and the resolution of the encoder ([cpr]) has to be written to object 608F<sub>h</sub> sub-index 1.

Closed-Loop Example Settings			
Parameter	Object	Value	Comment
Maximum current	2003 <sub>h</sub>	85	Set maximum motor current to 1 A.
Standby current	2004 <sub>h</sub>	10	Set standby current to 0.1 A.
Sensor selection code	606A <sub>h</sub>	0	Set to 0 in order to make encoder work. (default: -1 = no encoder)
Encoder resolution	608F <sub>h</sub> /1	10000	Set encoder resolution to 10000cpr.
Maximum correction tolerance	2022 <sub>h</sub>	255	
P part of PID regulator	202B <sub>h</sub>	8000	
I part of PID regulator	202C <sub>h</sub>	500	
I clipping value	202D <sub>h</sub>	400	
Clock divider for D part calculation	202E <sub>h</sub>	2	
D part clipping value	202F <sub>h</sub>	400000	
D part of PID regulator	2021 <sub>h</sub>	10000	

Table 3: PID Mode Example Settings

After these settings have been made, switch the state machine to OPERATIONAL (using the control word). Then, turn on PID operation by writing 2 (or 3) to object 2020<sub>h</sub>. In contrast to closed-loop mode, there is no need to wait for initialization of the PID mode.



## 2 Communication

### 2.1 Reference Model

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

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

The application and the application layer interact with service primitives.

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

*Table 4: Service Primitives*

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



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

*Table 5: Service Types*



## 2.2 NMT State Machine

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

Starting and resetting the device is controlled via the state machine. The NMT state machine consists of the states shown in figure 8.

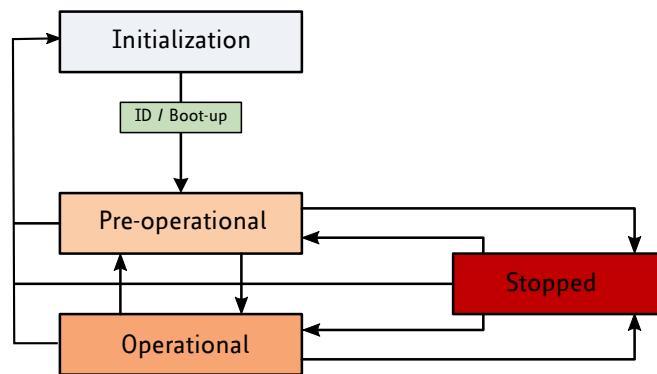


Figure 8: NMT State Machine

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

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

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

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

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

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



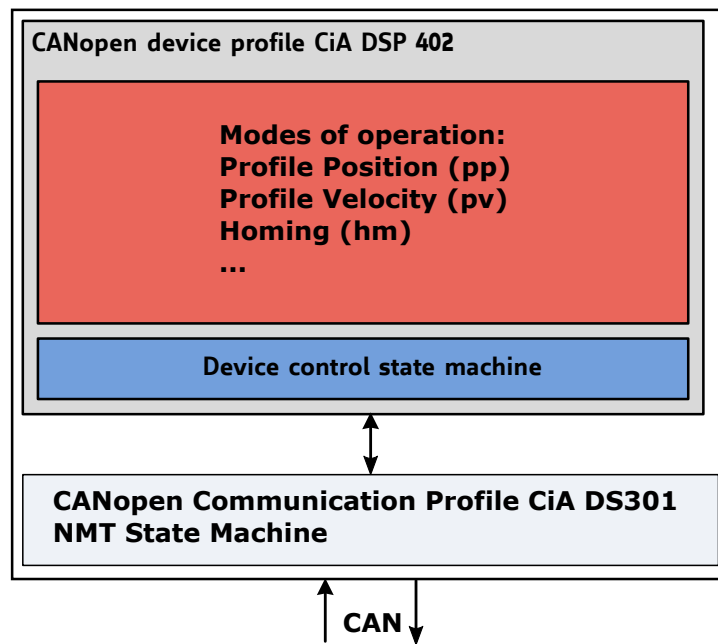


Figure 9: Communication Architecture

## 2.3 Device Model

A CANopen device mainly consists of the following parts:

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



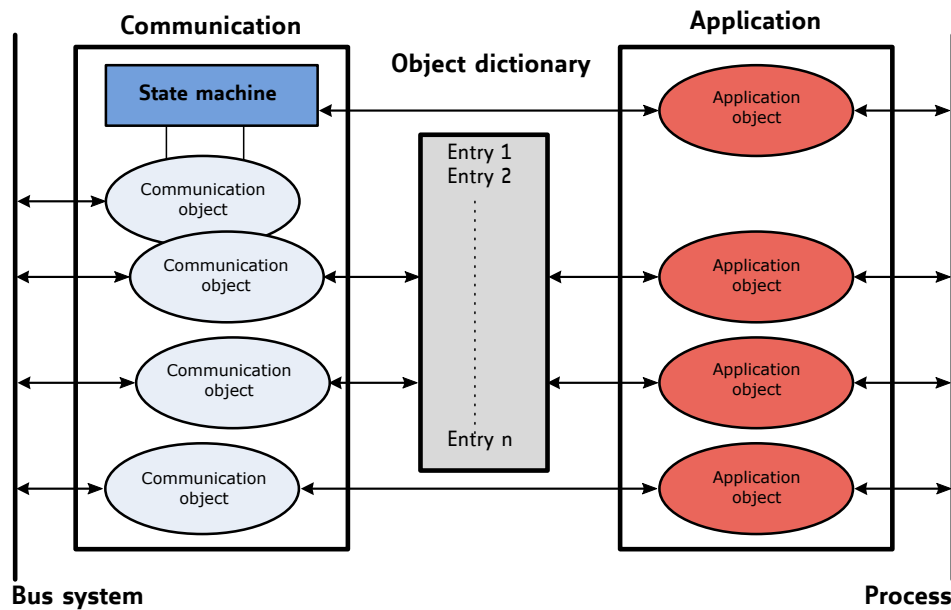


Figure 10: Device Model

## 2.4 Object Dictionary

The most important part of a device profile is the object dictionary description. The object dictionary is essentially a grouping of objects accessible via the network in an ordered pre-defined fashion. Each object within the dictionary is addressed using a 16-bit index. The overall layout of the standard object dictionary is shown in table 6:

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

Table 6: Object Dictionary



The communication profile area at indices 1000<sub>h</sub> through 1FFF<sub>h</sub> contains the communication specific parameters for the CAN network. These entries are common to all devices.

The manufacturer segment at indices 2000<sub>h</sub> through 5FFF<sub>h</sub> contains manufacturer specific objects. These objects control the special features of the Trinamic TMCM-3351 motion control device.

The standardized device profile area at indices 6000<sub>h</sub> through 9FFF<sub>h</sub> contains all data objects common to a class of devices that can be read or written via the network. They describe the device parameters and the device functionality of the device profile.

### 2.4.1 Object Indices on Multi-Axis Modules

On a multi-axis module like the TMCM-3351 each object in the manufacturer area and each object in the profile specific area is available for each motor. In this manual, only the object indices for motor #0 are shown. The objects for the other motors can be accessed by adding offsets to the object indices:

- Add an offset of  $motor\_number \cdot 200_h$  to the index of a manufacturer specific object to get its index for other motors.
- Add an offset of  $motor\_number \cdot 800_h$  to the index of a profile specific object to get its index for other motors.

For example, the control word for motor #1 would be 6840<sub>h</sub> (instead of 6040<sub>h</sub> for motor #0), and the microstep resolution of motor #1 would be 2200<sub>h</sub> for motor #1 (instead of 2000<sub>h</sub> for motor #0).

Multi-Axis Object Indices		
Motor	Manufacturer area	Profile area
Motor #0	2000 <sub>h</sub> – 21FF <sub>h</sub>	6000 <sub>h</sub> – 67FF <sub>h</sub>
Motor #1	2200 <sub>h</sub> – 23FF <sub>h</sub>	6800 <sub>h</sub> – 6FFF <sub>h</sub>
Motor #2	2400 <sub>h</sub> – 25FF <sub>h</sub>	7000 <sub>h</sub> – 77FF <sub>h</sub>

Table 7: Multi-Axis Object Indices





### 3 Communication Area

The communication area contains all objects that define the communication parameters of the CANopen device according to the DS301 standard.

#### 3.1 Detailed Object Specifications

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

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

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

Table 8: Object Description (1000<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	no	UNSIGNED32	FFFC0192 <sub>h</sub>

Table 9: Entry Description (1000<sub>h</sub>)

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

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

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

Table 10: Object Description (1001<sub>h</sub>)

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

Table 11: Entry Description (1001<sub>h</sub>)



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

Table 12: Error Register Bits

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

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

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

Table 13: Value Definition (1005<sub>h</sub>)

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

Table 14: Object Description (1005<sub>h</sub>)

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

Table 15: Entry Description (1005<sub>h</sub>)

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

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

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

Table 16: Object Description (1008<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	no	—	TMCM-3351

Table 17: Entry Description (1008<sub>h</sub>)

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

This object contains the hardware version description.

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

Table 18: Object Description (1009<sub>h</sub>)

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

Table 19: Entry Description (1009<sub>h</sub>)

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

This object contains the software version description.

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

Table 20: Object Description (100A<sub>h</sub>)



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

Table 21: Entry Description (100A<sub>h</sub>)

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

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

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

Table 22: Object Description (100C<sub>h</sub>)

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

Table 23: Entry Description (100C<sub>h</sub>)

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

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

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

Table 24: Object Description (100D<sub>h</sub>)

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

Table 25: Entry Description (100D<sub>h</sub>)

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

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



The TMCM-3351 module supports saving of the following parameter groups:

- Sub-index 1<sub>h</sub>: save all parameters.
- Sub-index 2<sub>h</sub>: save communication parameters 2704<sub>h</sub> and 2705<sub>h</sub>.
- Sub-index 4<sub>h</sub>: save motor 0 parameters.
- Sub-index 5<sub>h</sub>: save motor 1 parameters.
- Sub-index 6<sub>h</sub>: save motor 2 parameters.

**Note**

In order to avoid storage of parameters by mistake, storage is only executed when a specific signature is written to the appropriate sub-Index. This signature is "save" (65766173<sub>h</sub>, see also table 26).

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

Table 26: Save Signature

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

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

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

Table 27: Object Description (1010<sub>h</sub>)



Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	Highest supported sub-index	ro	no	UNSIGNED8	6
01 <sub>h</sub>	Save all parameters	rw	no	UNSIGNED32	—
02 <sub>h</sub>	Save communication parameters	rw	no	UNSIGNED32	—
04 <sub>h</sub>	Save motor 0 parameters	rw	no	UNSIGNED32	—
05 <sub>h</sub>	Save motor 1 parameters	rw	no	UNSIGNED32	—
06 <sub>h</sub>	Save motor 2 parameters	rw	no	UNSIGNED32	—

Table 28: Entry Description (1010<sub>h</sub>)

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

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

The TMC3351 module supports restoring of the following parameter groups:

- Sub-index 1<sub>h</sub>: restore all parameters (factory reset).
- Sub-index 2<sub>h</sub>: restore communication parameters 2704<sub>h</sub> and 2705<sub>h</sub>.
- Sub-index 4<sub>h</sub>: restore motor 0 parameters.
- Sub-index 5<sub>h</sub>: restore motor 1 parameters.
- Sub-index 6<sub>h</sub>: restore motor 2 parameters.

---

**Note** In order to avoid restoring the parameters by mistake, restoring is only executed when a specific signature is written to the appropriate sub-Index. This signature is "load" (64616F6C<sub>h</sub>, see also table 29).

---

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

Table 29: Load Signature

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

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



After the default values have been restored they will become active after the next rest or power cycle of the TMCM-3351.

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

Table 30: Object Description (1011<sub>h</sub>)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	Highest supported sub-index	ro	no	UNSIGNED8	6
01 <sub>h</sub>	Restore all parameters	rw	no	UNSIGNED32	—
02 <sub>h</sub>	Restore communication parameters	rw	no	UNSIGNED32	—
04 <sub>h</sub>	Restore motor 0 parameters	rw	no	UNSIGNED32	—
05 <sub>h</sub>	Restore motor 1 parameters	rw	no	UNSIGNED32	—
06 <sub>h</sub>	Restore motor 2 parameters	rw	no	UNSIGNED32	—

Table 31: Entry Description (1011<sub>h</sub>)

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

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

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

Table 32: Object Description (1014<sub>h</sub>)

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

Table 33: Entry Description (1014<sub>h</sub>)

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

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





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

Table 34: Object Description (1015<sub>h</sub>)

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

Table 35: Entry Description (1015<sub>h</sub>)

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

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

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

Table 36: Value Definition (1016<sub>h</sub>)

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

Table 37: Object Description (1016<sub>h</sub>)

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

Table 38: Entry Description (1016<sub>h</sub>)

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

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

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

Table 39: Object Description (1017<sub>h</sub>)

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

Table 40: Entry Description (1017<sub>h</sub>)

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

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

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

Object Description			
Index	Name	Object Type	Data Type
1018 <sub>h</sub>	Identity object	Record	Identity

Table 41: Object Description (1018<sub>h</sub>)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	Number of entries	ro	no	0...3	3
01 <sub>h</sub>	Vendor ID	ro	no	UNSIGNED32	0286 <sub>h</sub>
02 <sub>h</sub>	Product code	ro	no	UNSIGNED32	3351
03 <sub>h</sub>	Revision number	ro	no	UNSIGNED32	e.g. 20003 <sub>h</sub> for version 2.3

Table 42: Entry Description (1018<sub>h</sub>)



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

After switching to TMCL command mode using object 2700<sub>h</sub>, this object can be used to send TMCL commands to the TMC3351 module. A TMCL command can be written to sub-index 1. After the command has been executed, the status code and the TMCL reply can be read from sub-index 2 and sub-index 3.

Object Description			
Index	Name	Object Type	Data Type
1023 <sub>h</sub>	OS Command	Record	Command

Table 43: Object Description (1023<sub>h</sub>)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
00 <sub>h</sub>	Number of entries	ro	no	—	3
01 <sub>h</sub>	TMCL command	rw	no	Octet string (7 bytes)	—
02 <sub>h</sub>	Reply	ro	no	UNSIGNED8	—
03 <sub>h</sub>	TMCL reply	ro	no	Octet string (7 bytes)	—

Table 44: Entry Description (1023<sub>h</sub>)

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

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

- Communication error
- Application error

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

Table 45: Object Description (1029<sub>h</sub>)

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

Table 46: Entry Description (1029<sub>h</sub>)



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

This object contains the communication parameters for the RPDOs which the device is able to receive. The sub-index 00<sub>h</sub> contains the number of valid entries within the communication record. Its value normally is 2, as this object consists of two other entries.

Sub-index 01<sub>h</sub> contains the COB-ID used by this PDO (in bits 10...0). Bit 30 (RTR bit) defines if this PDO uses RTRs. As RTRs are not supported for PDOs by this CANopen implementation, this bit must always be set in order to turn off RTR support for this PDO. Bit 31 defines if this PDO is active or not. If this bit is set, the PDO is inactive, and if this bit is clear, the PDO is active. Before making any changes to a PDO definition, set this bit to inactivate the PDO.

Sub-Index 02<sub>h</sub> contains the transmission type of the RPDO. This can be FF<sub>h</sub> or FE<sub>h</sub> for event-driven, or 00<sub>h</sub> for synchronous.

The RPDOs defined by objects 1400<sub>h</sub> – 1403<sub>h</sub> (RPDO 1...3) are normally used for motor #0. For the other motors, the following RPDOs are used:

- RPDO 65...68 (objects 1440<sub>h</sub> – 1443<sub>h</sub>) for motor #1.
- RPDO 129...132 (objects 1480<sub>h</sub> – 1483<sub>h</sub>) for motor #2.

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

Table 47: Object Description (1400<sub>h</sub>)

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

Table 48: Entry Description (1400<sub>h</sub>)



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

These objects contain the mapping parameters for the RPDOs the device is able to receive. The sub-index 00<sub>h</sub> contains the number of valid entries within the mapping record. This number of entries is also the number of the application variables which shall be received with the corresponding RPDO. The sub-indices from 01<sub>h</sub> to the number of entries contain the information about the mapped application variables. These entries describe the PDO contents by their index, sub-index and length.

The RPDOs defined by objects 1600<sub>h</sub> – 1603<sub>h</sub> (RPDO 1...3) are normally used for motor #0. For the other motors, the following RPDOs are used:

- RPDO 65...68 (objects 1640<sub>h</sub> – 1643<sub>h</sub>) for motor #1.
- RPDO 129...132 (objects 1680<sub>h</sub> – 1683<sub>h</sub>) for motor #2.

Object Description			
Index	Name	Object Type	Data Type
1600 <sub>h</sub> – 1603 <sub>h</sub>	Receive PDO mapping parameter	RECORD	PDO Mapping
1600 <sub>h</sub>	RPDO 1	RECORD	PDO Mapping
1601 <sub>h</sub>	RPDO 2	RECORD	PDO Mapping
1602 <sub>h</sub>	RPDO 3	RECORD	PDO Mapping
1603 <sub>h</sub>	RPDO 4	RECORD	PDO Mapping

Table 49: Object Description (1600<sub>h</sub>)

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

Table 50: Entry Description (1600<sub>h</sub>)



Before making changes to PDO definitions, first mark the PDO as inactive by setting bit 31 of its COB-ID (see section 3.1.18). Then, set its number of mapped PDO entries to zero (sub-index 0 of the appropriate PDO mapping object). Now, the mappings themselves can be changed. After that, set the number of map objects to the desired value, and finally activate the PDO by clearing bit 31 of its COB-ID.

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

This object contains the communication parameters for the TPDOs which the device is able to transmit. The sub-index 00<sub>h</sub> contains the number of valid entries within the communication record. Its value normally is 5, as this object consists of five other entries.

Sub-index 01<sub>h</sub> contains the COB-ID used by this PDO (in bits 10...0). Bit 30 (RTR bit) defines if this PDO uses RTRs. As RTRs are not supported for PDOs by this CANopen implementation, this bit must always be set in order to turn off RTR support for this PDO. Bit 31 defines if this PDO is active or not. If this bit is set, the PDO is inactive, and if this bit is clear, the PDO is active. Before making any changes to a PDO definition, set this bit to inactivate the PDO.

Sub-index 02<sub>h</sub> contains the transmission type of the RPDO. This can be FF<sub>h</sub> or FE<sub>h</sub> for event-driven, or 00<sub>h</sub> or 01<sub>h</sub> for synchronous.

Sub-index 03<sub>h</sub> contains the inhibit time, given in milliseconds. After a TPDO has been sent, it will not be sent again before the inhibit time has elapsed.

Sub-index 04<sub>h</sub> is not used.

Sub-index 05<sub>h</sub> contains the event timer value in milliseconds. When this is set to a value greater than 0 the TPDO will be sent repeatedly each time the event timer has elapsed. For example, when this value is set to 250, the TPDO will be sent every 250ms.

The TPDOs defined by objects 1800<sub>h</sub> – 1803<sub>h</sub> (TPDO 1...3) are normally used for motor #0. For the other motors, the following TPDOs are used:

- TPDO 65...68 (objects 1840<sub>h</sub> – 1843<sub>h</sub>) for motor #1.
- TPDO 129...132 (objects 1880<sub>h</sub> – 1883<sub>h</sub>) for motor #2.

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

Table 51: Object Description (1800<sub>h</sub>)



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

Table 52: Entry Description (1800<sub>h</sub>)

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

These objects contain the mapping parameters for the TPDOs the device is able to transmit. The sub-index 00<sub>h</sub> contains the number of valid entries within the mapping record. This number of entries is also the number of the application variables which shall be transmitted with the corresponding TPDO. The sub-indices from 01<sub>h</sub> to the number of entries contain the information about the mapped application variables. These entries describe the PDO contents by their index, sub-index and length.

The TPDOs defined by objects 1A00<sub>h</sub> – 1A03<sub>h</sub> (TPDO 1...3) are normally used for motor #0. For the other motors, the following TPDOs are used:

- TPDO 65...68 (objects 1A40<sub>h</sub> – 1A43<sub>h</sub>) for motor #1.
- TPDO 129...132 (objects 1A80<sub>h</sub> – 1A83<sub>h</sub>) for motor #2.

Object Description			
Index	Name	Object Type	Data Type
1A00 <sub>h</sub> – 1A03 <sub>h</sub>	Transmit PDO mapping parameter	RECORD	PDO Mapping
1A00 <sub>h</sub>	TPDO 1	RECORD	PDO Mapping
1A01 <sub>h</sub>	TPDO 2	RECORD	PDO Mapping
1A02 <sub>h</sub>	TPDO 3	RECORD	PDO Mapping
1A03 <sub>h</sub>	TPDO 4	RECORD	PDO Mapping

Table 53: Object Description (1A00<sub>h</sub>)





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

*Table 54: Entry Description (1A00<sub>h</sub>)*

Before making changes to PDO definitions, first mark the PDO as inactive by setting bit 31 of its COB-ID (see section 3.1.20). Then, set its number of mapped PDO entries to zero (sub-index 0 of the appropriate PDO mapping object). Now, the mappings themselves can be changed. After that, set the number of map objects to the desired value, and finally activate the PDO by clearing bit 31 of its COB-ID.



## 4 Manufacturer specific Area

The manufacturer segment contains manufacturer specific objects. These objects control the special features of the Trinamic Motion Control device TMCM-3351.

### **i** Info

This section of the manual only shows the object indices for motor #0. Of course the same objects are also available for the other motors. For the other motors, add an offset of  $motor\_number \cdot 200_h$  to the object index. So for example the microstep resolution (object  $2000_h$  for motor #0) can be accessed as object  $2200_h$  for motor #1 and as object  $2400_h$  for motor #2. Please see also section 2.4.1.

Multi-axis Object Indices	
Motor	Object Index Range
Motor #0	$2000_h - 21FF_h$
Motor #1	$2200_h - 23FF_h$
Motor #2	$2400_h - 25FF_h$

Table 55: Multi-axis Object Indices (Manufacturer specific Area)

### 4.1 Objects related to CoolStep™

Figure 11 shows an overview of the CoolStep™ related objects for motor #0. Please bear in mind that the figure only shows one example for a drive. There are objects which concern the configuration of the current. Other objects are for velocity regulation and for time adjustment. The CoolStep™ feature is sometimes also called SmartEnergy.

The following adjustments have to be made:

- Thresholds for current and velocity have to be identified and set.
- The StallGuard2™ feature has to be adjusted and enabled.
- The reduction or increasing of the current in the CoolStep™ area (depending on the load) has to be configured.



### coolStep™ adjustment points and thresholds

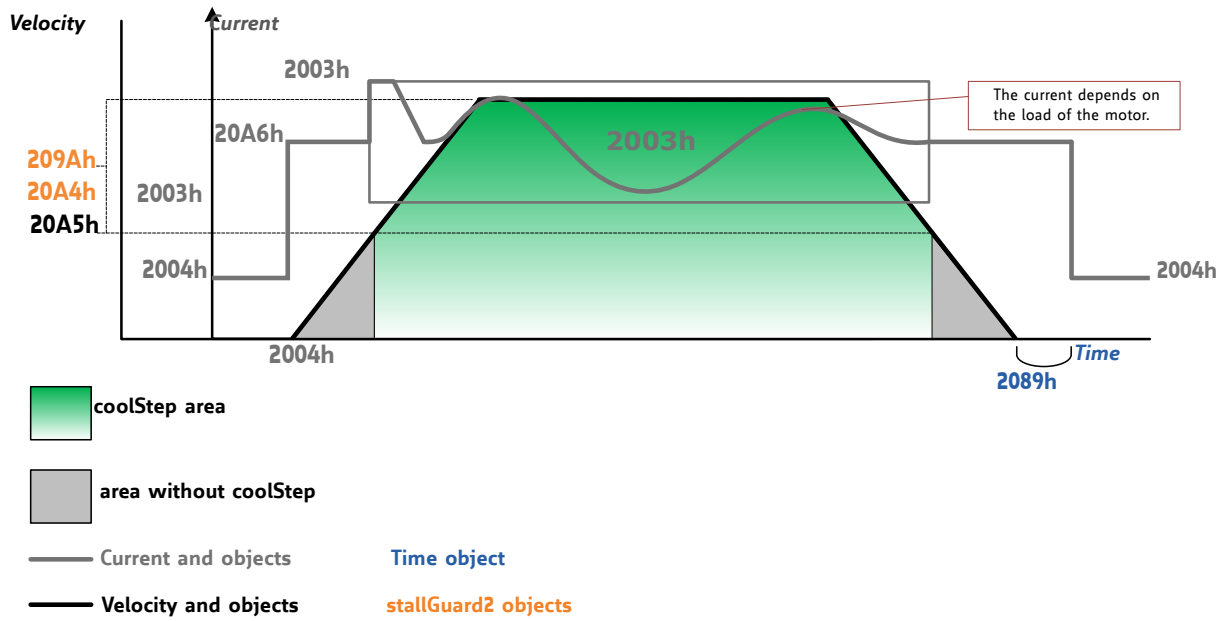


Figure 11: CoolStep Adjustment Points and Thresholds



CoolStep Adjustment Objects		
Object	Name	Description
2003 <sub>h</sub>	Absolute maximum current	The maximum value is 255. This value means 100% of the maximum current of the module. The current adjustment is within the range 0...255 and can be adjusted in 32 steps (0...255 divided by eight; step 0 = 0...7, step 1 = 8...15 and so on). The most important motor setting, since too high values might cause motor damage!
2004 <sub>h</sub>	Standby current	The current limit two seconds after the motor has stopped.
2098 <sub>h</sub>	SmartEnergy current minimum	Sets the lower motor current limit for CoolStep operation by scaling the run current (object 2003 <sub>h</sub> ) value. This can be: 0: for 1/2 of the run current 1: for 1/4 of the run current
2099 <sub>h</sub>	SmartEnergy current down step	Sets the speed of current decrement when the StallGuard2 reading is above the upper threshold. 0: slow decrement 3: fast decrement
209B <sub>h</sub>	SmartEnergy current up step	Sets the current increment step when the StallGuard2 below the lower threshold. 0: slow increment 3: fast increment / fast reaction to rising load
209A <sub>h</sub>	SmartEnergy hysteresis	Sets the distance between the lower and the upper threshold for StallGuard2 reading. Above the upper threshold the motor current becomes decreased.
20A4 <sub>h</sub>	Stop on stall	Below this speed the motor will not be stopped. Above this speed the motor will stop in case StallGuard2 load value reaches zero.
20A5 <sub>h</sub>	SmartEnergy threshold speed.	Above this speed CoolStep becomes enabled.
2089 <sub>h</sub>	Standby delay	Standstill period before the current is changed down to standby current. The standard value is 200 which is 2 seconds.

Table 56: CoolStep related Objects



## 4.2 Detailed Object Specifications

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

This object sets the microstep resolution of the drive. A value of 8 selects 256 (2<sup>8</sup>) microsteps per full step.

Object Description			
Index	Name	Object Type	Data Type
2000 <sub>h</sub>	Microstep Resolution	Variable	UNSIGNED8

Table 57: Object Description (2000<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0...8	8

Table 58: Entry Description (2000<sub>h</sub>)

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

This object sets the fullstep resolution of the motor connected to the drive. Its default value is 200 because most motors are 1.8° motors.

Object Description			
Index	Name	Object Type	Data Type
2001 <sub>h</sub>	Motor full step resolution	Variable	UNSIGNED16

Table 59: Object Description (2001<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0...65535	200

Table 60: Entry Description (2001<sub>h</sub>)

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

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



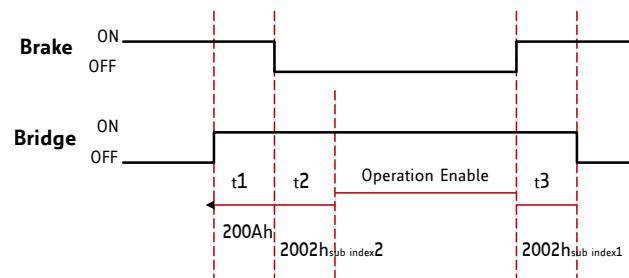


Figure 12: Brake Output Timing

Object Description			
Index	Name	Object Type	Data Type
2002 <sub>h</sub>	Brake delay times	Array	UNSIGNED16

Table 61: Object Description (2002<sub>h</sub>)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
1	Time between applying brake / disabling power stage	rw	no	0...65535	0
2	Time between releasing brake / switching the state machine to operational	rw	no	0..65535	0

Table 62: Entry Description (2002<sub>h</sub>)

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

This object defines the current used when the motor is moving. A value of 255 means 100% of the maximum current of the drive.

Object Description			
Index	Name	Object Type	Data Type
2003 <sub>h</sub>	Maximum current	Variable	UNSIGNED8

Table 63: Object Description (2003<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0...255	128

Table 64: Entry Description (2003<sub>h</sub>)



#### 4.2.5 Object 2004<sub>h</sub>: Standby Current

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

Object Description			
Index	Name	Object Type	Data Type
2004 <sub>h</sub>	Maximum current	Variable	UNSIGNED8

Table 65: Object Description (2004<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0...255	8

Table 66: Entry Description (2004<sub>h</sub>)

#### 4.2.6 Object 2005<sub>h</sub>: Limit Switches

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

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

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

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

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

Table 67: Object Description (2005<sub>h</sub>)

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

Table 68: Entry Description (2005<sub>h</sub>)



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

Table 69: Bit Definitions (2005<sub>h</sub>)

#### 4.2.7 Object 200A<sub>h</sub>: Enable Drive Delay Time

This is an additional delay time (in milliseconds) between enabling the power stage and releasing the brake. It can be used to prevent the brake from being released too early (before the hold current in the motor has been reached). Please see also object 4.2.3.

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

Table 70: Object Description (200A<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0...65535	0

Table 71: Entry Description (200A<sub>h</sub>)

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

This object defines encoder parameters. These are the polarity of the encoder null channel, the direction of rotation (set it to 1 if the direction is reversed compared to the motor) and if the position is to be initialized with the encoder position. It is only writable in SWITCHED\_ON\_DISABLED state.

Object Description			
Index	Name	Object Type	Data Type
200B <sub>h</sub>	Encoder parameters	Array	UNSIGNED8

Table 72: Object Description (200B<sub>h</sub>)





Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
1	Null channel polarity	rw	no	0/1	0
2	Direction of rotation	rw	no	0/1	0
3	Initialize position	rw	no	0/1	1

Table 73: Entry Description (200B<sub>h</sub>)

#### 4.2.9 Object 200C<sub>h</sub>: Brake Current Feed

This object configures how much current has to be fed into the brake to apply and to release it. 0 means 0%, 255 means 100% of the maximum current (this depends on the module). In most cases it is needed to feed current into the brake to release it. Setting both values to 0 disables the automatic brake control. This object is only writable in SWITCHED\_ON\_DISABLED state.

Object Description			
Index	Name	Object Type	Data Type
200C <sub>h</sub>	Brake current feed	Array	UNSIGNED8

Table 74: Object Description (200C<sub>h</sub>)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
1	Apply current	rw	no	0...255	0
2	Release current	rw	no	0...255	0

Table 75: Entry Description (200C<sub>h</sub>)

#### 4.2.10 Object 200F<sub>h</sub>: Encoder N Channel Latch

With this object it is possible to make use of the encoder N channel latch capabilities of the motion controller used on the TMCM-3351 module. Write 1 to sub-index 1 to activate the latch functionality. When the encoder N channel has been passed, the internal position and the encoder position will automatically be stored to sub-index 2 and sub-index 3 of this object, and sub-index 1 will be reset to zero to show that the N channel event has occurred.

The encoder interface must have been configured properly in order to make this function work correctly.

Object Description			
Index	Name	Object Type	Data Type
200F <sub>h</sub>	Encoder N Channel Latch	Variable	RECORD

Table 76: Object Description (200F<sub>h</sub>)



Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
1	Mode	rw	no	0/1	0
2	Latched internal position	ro	no	SIGNED32	0
3	Latched encoder position	ro	no	SIGNED32	0

Table 77: Entry Description (200F<sub>h</sub>)

#### 4.2.11 Object 2010<sub>h</sub>: Profile Start Velocity

This object contains the velocity with which a positioning ramp will be started.

Object Description			
Index	Name	Object Type	Data Type
2010 <sub>h</sub>	Profile Start Velocity	Variable	UNSIGNED32

Table 78: Object Description (2010<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0...268435455	0

Table 79: Entry Description (2010<sub>h</sub>)

#### 4.2.12 Object 2011<sub>h</sub>: Profile Start Acceleration

This object contains the acceleration value used for ramping up from the start velocity (object 2011<sub>h</sub>, see section 4.2.11) to the velocity V1 (object (2012<sub>h</sub>), see section 4.2.13).

Object Description			
Index	Name	Object Type	Data Type
2011 <sub>h</sub>	Profile A1	Variable	UNSIGNED32

Table 80: Object Description (2011<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0...16777215	0

Table 81: Entry Description (2011<sub>h</sub>)



### 4.2.13 Object 2012<sub>h</sub>: Profile Break Velocity

This object contains the velocity used for the first segment of a positioning ramp.

Object Description			
Index	Name	Object Type	Data Type
2012 <sub>h</sub>	Profile V1	Variable	UNSIGNED32

Table 82: Object Description (2012<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0...16777215	0

Table 83: Entry Description (2012<sub>h</sub>)

### 4.2.14 Object 2013<sub>h</sub>: Profile Final Deceleration

This object contains the deceleration value used for decelerating from the maximum positioning velocity to the velocity V1 (object 2012<sub>h</sub>, see section 4.2.13).

Object Description			
Index	Name	Object Type	Data Type
2013 <sub>h</sub>	Profile D1	Variable	UNSIGNED32

Table 84: Object Description (2013<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0...16777215	0

Table 85: Entry Description (2013<sub>h</sub>)

### 4.2.15 Object 2014<sub>h</sub>: Profile Stop Deceleration

This object sets the deceleration used to decelerate from the end velocity to zero velocity.

Object Description			
Index	Name	Object Type	Data Type
2014 <sub>h</sub>	Profile stop deceleration	Variable	UNSIGNED32

Table 86: Object Description (2014<sub>h</sub>)



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

Table 87: Entry Description (2014<sub>h</sub>)

#### 4.2.16 Object 2015<sub>h</sub>: Bow Scaling Factor

The bow or jerk values (object 60A4<sub>h</sub>) will internally be multiplied by this value so that also bow values higher than 2147483647 can be reached. This is only needed in rare cases where very steep S-shaped ramps are required. In most cases this value can be left at its default setting of 1. The bow values multiplied by this factor must not exceed  $467.837 \cdot 10^9$ .

Object Description			
Index	Name	Object Type	Data Type
2015 <sub>h</sub>	Ramp Wait Time	Variable	UNSIGNED8

Table 88: Object Description (2015<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	1...255	1

Table 89: Entry Description (2015<sub>h</sub>)

#### 4.2.17 Object 2020<sub>h</sub>: Closed Loop Mode

Setting this object to 1 selects closed-loop mode, setting it to 0 (default value) selects open-loop mode. Before activating closed-loop mode, all necessary parameters must have been set. After activating closed-loop mode, check object 2120<sub>h</sub> until closed loop initialization has finished (please see also section 4.2.61).

Object Description			
Index	Name	Object Type	Data Type
2020 <sub>h</sub>	Closed-loop mode	Variable	UNSIGNED8

Table 90: Object Description (2020<sub>h</sub>)

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

Table 91: Entry Description (2020<sub>h</sub>)

#### 4.2.18 Object 2021<sub>h</sub>: Correction Position P

Proportional controller P parameter for compensating a detected position deviation. Resulting P parameter is this parameter divided by 65536.

Object Description			
Index	Name	Object Type	Data Type
2021 <sub>h</sub>	Correction position P	Variable	UNSIGNED32

Table 92: Object Description (2021<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0...16777215	0

Table 93: Entry Description (2021<sub>h</sub>)

#### 4.2.19 Object 2022<sub>h</sub>: Maximum Correction Tolerance

This object defines the tolerance range for position deviation.

Object Description			
Index	Name	Object Type	Data Type
2022 <sub>h</sub>	Maximum correction tolerance	Variable	UNSIGNED8

Table 94: Object Description (2022<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0...255	0

Table 95: Entry Description (2022<sub>h</sub>)

#### 4.2.20 Object 2027<sub>h</sub>: Closed Loop Beta

This object sets the maximum commutation angle that can be used to compensate for an evaluated position deviation.

Object Description			
Index	Name	Object Type	Data Type
2027 <sub>h</sub>	Maximum correction tolerance	Variable	UNSIGNED16

Table 96: Object Description (2027<sub>h</sub>)



Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0...511	255

Table 97: Entry Description (2027<sub>h</sub>)

#### 4.2.21 Object 2028<sub>h</sub>: Closed Loop Offset

Offset for closed loop operation. Automatically measured during closed loop initialization. Normally no need to change this value.

Object Description			
Index	Name	Object Type	Data Type
2028 <sub>h</sub>	Closed-loop Offset	Variable	INTEGER32

Table 98: Object Description (2028<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	-2147483648...2147483647	—

Table 99: Entry Description (2028<sub>h</sub>)

#### 4.2.22 Object 2029<sub>h</sub>: Current Scaler Minimum

Minimum current setting for current regulation in closed-loop operation.

Object Description			
Index	Name	Object Type	Data Type
2029 <sub>h</sub>	Current scaler minimum	Variable	UNSIGNED8

Table 100: Object Description (2029<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0...255	—

Table 101: Entry Description (2029<sub>h</sub>)

#### 4.2.23 Object 202A<sub>h</sub>: Current Scaler Maximum

Maximum current setting for current regulation in closed-loop operation.



Object Description			
Index	Name	Object Type	Data Type
202A <sub>h</sub>	Current scaler maximum	Variable	UNSIGNED8

Table 102: Object Description (202A<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0...255	—

Table 103: Entry Description (202A<sub>h</sub>)

#### 4.2.24 Object 202B<sub>h</sub>: Correction Velocity P

This object defines the P parameter of the PI regulator which controls the maximum velocity during closed-loop regulation.

Object Description			
Index	Name	Object Type	Data Type
202B <sub>h</sub>	Correction velocity P	Variable	UNSIGNED32

Table 104: Object Description (202B<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0...16777215	—

Table 105: Entry Description (202B<sub>h</sub>)

#### 4.2.25 Object 202C<sub>h</sub>: Correction Velocity I

This object defines the I parameter of the PI regulator which controls the maximum velocity during closed-loop regulation.

Object Description			
Index	Name	Object Type	Data Type
202C <sub>h</sub>	Correction velocity I	Variable	UNSIGNED32

Table 106: Object Description (202C<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0...16777215	—

Table 107: Entry Description (202C<sub>h</sub>)

#### 4.2.26 Object 202D<sub>h</sub>: Correction Velocity I Clipping

This object defines the clipping of the error sum of the integral part of the PI regulator which controls the maximum velocity during closed loop regulation.

Object Description			
Index	Name	Object Type	Data Type
202D <sub>h</sub>	Correction velocity I clipping	Variable	UNSIGNED16

Table 108: Object Description (202D<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0...32767	—

Table 109: Entry Description (202D<sub>h</sub>)

#### 4.2.27 Object 202E<sub>h</sub>: Correction Velocity DV Clock

This object defines the clock divider for the D part calculation of the regulator that controls the velocity during closed-loop regulation.

Object Description			
Index	Name	Object Type	Data Type
202E <sub>h</sub>	Correction velocity DV clock	Variable	UNSIGNED16

Table 110: Object Description (202E<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0...32767	—

Table 111: Entry Description (202E<sub>h</sub>)



#### 4.2.28 Object 202F<sub>h</sub>: Correction Velocity DV Clipping

The maximum velocity used for position correction in closed-loop mode will be limited to the value set by this object.

Object Description			
Index	Name	Object Type	Data Type
202F <sub>h</sub>	Correction velocity DV clock	Variable	UNSIGNED32

Table 112: Object Description (202F<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	UNSIGNED32	2147483647

Table 113: Entry Description (202F<sub>h</sub>)

#### 4.2.29 Object 2030<sub>h</sub>: Upscale Delay

Delay used when increasing the motor current during closed-loop operation.

Object Description			
Index	Name	Object Type	Data Type
2030 <sub>h</sub>	Upscale delay	Variable	UNSIGNED32

Table 114: Object Description (2030<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0...16777215	—

Table 115: Entry Description (2030<sub>h</sub>)

#### 4.2.30 Object 2031<sub>h</sub>: Downscale Delay

Delay used when decreasing the motor current during closed-loop operation.

Object Description			
Index	Name	Object Type	Data Type
2031 <sub>h</sub>	Downscale delay	Variable	UNSIGNED32

Table 116: Object Description (2031<sub>h</sub>)



Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0...16777215	—

Table 117: Entry Description (2031<sub>h</sub>)

#### 4.2.31 Object 2033<sub>h</sub>: Actual Scaling Factor

The actual current scaling factor (output of the current scaler) can be read from this object.

Object Description			
Index	Name	Object Type	Data Type
2033 <sub>h</sub>	Actual scaling factor	Variable	UNSIGNED8

Table 118: Object Description (2033<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	no	0...255	-

Table 119: Entry Description (2033<sub>h</sub>)

#### 4.2.32 Object 2034<sub>h</sub>: Field Weakening Minimum Velocity

Minimum speed for field weakening operation. For higher velocities the influence of the back EMF may be compensated. At this velocity the compensation will start.

Object Description			
Index	Name	Object Type	Data Type
2034 <sub>h</sub>	Filed weakening minimum velocity	Variable	UNSIGNED32

Table 120: Object Description (2034<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0...16777215	—

Table 121: Entry Description (2034<sub>h</sub>)



### 4.2.33 Object 2035<sub>h</sub>: Field Weakening Maximum Velocity

Maximum speed for field weakening operation. For higher velocities the influence of the back EMF may be compensated. At the field weakening minimum velocity (section 4.2.32) plus this velocity the back EMF compensation will reach its maximum.

Object Description			
Index	Name	Object Type	Data Type
2035 <sub>h</sub>	Field weakening maximum velocity	Variable	UNSIGNED32

Table 122: Object Description (2035<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0...16777215	—

Table 123: Entry Description (2035<sub>h</sub>)

### 4.2.34 Object 2036<sub>h</sub>: Field Weakening

For higher velocities the influence of the back EMF may be compensated. This is the compensation factor that will be added scaled by the current velocity and the minimum and maximum field weakening velocity.

Object Description			
Index	Name	Object Type	Data Type
2036 <sub>h</sub>	Field weakening	Variable	UNSIGNED8

Table 124: Object Description (2036<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0...255	—

Table 125: Entry Description (2036<sub>h</sub>)

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

This object defines the motor current that will be used during the acceleration phase and the deceleration phase in open-loop mode. A value of 255 means 100% of the maximum motor current of the drive. When this object is set to zero the run current (section 4.2.4) will also be used for the acceleration and deceleration phase.



Object Description			
Index	Name	Object Type	Data Type
204E <sub>h</sub>	Field weakening	Variable	UNSIGNED8

Table 126: Object Description (204E<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0...255	—

Table 127: Entry Description (204E<sub>h</sub>)

#### 4.2.36 Object 2089<sub>h</sub>: Setting Delay

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

**Unit:** 10msec

Object Description			
Index	Name	Object Type	Data Type
2089 <sub>h</sub>	Setting Delay	Variable	UNSIGNED16

Table 128: Object Description (2089<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0...65535	200

Table 129: Entry Description (2089<sub>h</sub>)

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

With this object different units can be chosen:

- Writing 0 selects internal units.
- 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 Type	Data Type
208C <sub>h</sub>	Velocity Dimension Index	Variable	UNSIGNED8

Table 130: Object Description (208C<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0/181	181

Table 131: Entry Description (208C<sub>h</sub>)

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

With this object, the unit for acceleration can be read out. The unit can be set using object 208C<sub>h</sub>. Object 208E<sub>h</sub> reads 0 when internal units are selected and 179 when PPS/s is selected.

Object Description			
Index	Name	Object Type	Data Type
208E <sub>h</sub>	Acceleration Dimension Index	Variable	UNSIGNED8

Table 132: Object Description (208E<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	no	0/179	179

Table 133: Entry Description (208E<sub>h</sub>)

#### 4.2.39 Object 2092<sub>h</sub>: Chopper Blank Time

This object serves for selecting the comparator blank time. This time needs to safely cover the switching event and the duration of the ringing on the sense resistor. For low current drivers, a setting of 1 or 2 is good. For higher current applications a setting of 2 or 3 will be required.

Object Description			
Index	Name	Object Type	Data Type
2092 <sub>h</sub>	Chopper Blank Time	Variable	UNSIGNED8

Table 134: Object Description (2092<sub>h</sub>)



Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0...3	2

Table 135: Entry Description (2092<sub>h</sub>)

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

Select the chopper mode using this object:

- 0 – spreadCycle chopper
- 1 – classic constant off time chopper

Object Description			
Index	Name	Object Type	Data Type
2093 <sub>h</sub>	Chopper Mode	Variable	UNSIGNED8

Table 136: Object Description (2093<sub>h</sub>)

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

Table 137: Entry Description (2093<sub>h</sub>)

#### 4.2.41 Object 2094<sub>h</sub>: Chopper Hysteresis Decrement

This object serves for the hysteresis decrement setting. This setting determines the slope of the hysteresis during on time and during fast decay time.

- 0 – fast decrement
- 3 – very slow decrement

Object Description			
Index	Name	Object Type	Data Type
2094 <sub>h</sub>	Chopper Hysteresis Decrement	Variable	UNSIGNED8

Table 138: Object Description (2094<sub>h</sub>)

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

Table 139: Entry Description (2094<sub>h</sub>)



#### 4.2.42 Object 2095<sub>h</sub>: Chopper Hysteresis End

This object provides the setting of the hysteresis end value after a number of decrements. The decrement interval time is controlled by object 2094<sub>h</sub> (section 4.2.41).

Possible values are:

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

Object Description			
Index	Name	Object Type	Data Type
2095 <sub>h</sub>	Chopper Hysteresis End	Variable	SIGNED8

Table 140: Object Description (2095<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	-3...12	0

Table 141: Entry Description (2095<sub>h</sub>)

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

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

Object Description			
Index	Name	Object Type	Data Type
2096 <sub>h</sub>	Chopper Hysteresis Start	Variable	UNSIGNED8

Table 142: Object Description (2096<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0...8	3

Table 143: Entry Description (2096<sub>h</sub>)

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

The off time setting controls the minimum chopper frequency. Under normal circumstances, an off time within the range of 5µs to 20µs is used. Off time setting for constant t<sub>OFF</sub> chopper:  $N_{CLK} = 12 + 32 * t_{OFF}$ .



Minimum is 64 clocks.

Setting this parameter to zero completely disables all driver transistors and so lets the motor free-wheel.

Object Description			
Index	Name	Object Type	Data Type
2097 <sub>h</sub>	Chopper Off Time	Variable	UNSIGNED8

Table 144: Object Description (2097<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0 / 2...15	5

Table 145: Entry Description (2097<sub>h</sub>)

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

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

Possible values are:

- 0 – 1/2 of maximum motor current setting (section 4.2.4)
- 1 – 1/4 of maximum motor current setting (section 4.2.4)

Object Description			
Index	Name	Object Type	Data Type
2098 <sub>h</sub>	Smart Energy Current Minimum	Variable	UNSIGNED8

Table 146: Object Description (2098<sub>h</sub>)

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

Table 147: Entry Description (2098<sub>h</sub>)

#### 4.2.46 Object 2099<sub>h</sub>: Smart Energy Current Down Step

This object provides the setting of the number of StallGuard2 readings above the upper threshold necessary for each current decrement of the motor current.

Possible values are:

- 0 – 32 measurements – slowest decrement
- 1 – 8 measurements





- 2 – 2 measurements
- 3 – 1 measurements – fastest decrement

Object Description			
Index	Name	Object Type	Data Type
2099 <sub>h</sub>	Smart Energy Current Down Step	Variable	UNSIGNED8

Table 148: Object Description (2099<sub>h</sub>)

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

Table 149: Entry Description (2099<sub>h</sub>)

#### 4.2.47 Object 209A<sub>h</sub>: Smart Energy Hysteresis

This object sets the distance between the lower and the upper threshold for StallGuard2 reading. Above the upper threshold the motor current becomes decreased.

Hysteresis: (SmartEnergy hysteresis value + 1) \* 32

Upper StallGuard2 threshold: (SmartEnergy hysteresis start + SmartEnergy hysteresis + 1) \* 32

Object Description			
Index	Name	Object Type	Data Type
209A <sub>h</sub>	Smart Energy Hysteresis	Variable	UNSIGNED8

Table 150: Object Description (209A<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0...15	0

Table 151: Entry Description (209A<sub>h</sub>)

#### 4.2.48 Object 209B<sub>h</sub>: Smart Energy Current Up Step

This object sets the current increment step. The current becomes incremented for each measured StallGuard2 value below the lower threshold (see smart energy hysteresis start (object 209C<sub>h</sub>, section 4.2.49). Possible values are:

- 0 – 1 step – slowest increment
- 1 – 2 steps



- 2 – 4 steps
- 3 – 8 steps – fastest increment

Object Description			
Index	Name	Object Type	Data Type
209B <sub>h</sub>	Smart Energy Current Up Step	Variable	UNSIGNED8

Table 152: Object Description (209B<sub>h</sub>)

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

Table 153: Entry Description (209B<sub>h</sub>)

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

This object serves to set the lower threshold for the StallGuard2 value (see smart Energy current up step (section 4.2.48)). Setting this to 0 disables the CoolStep™ function.

Object Description			
Index	Name	Object Type	Data Type
209C <sub>h</sub>	Smart Energy Hysteresis Start	Variable	UNSIGNED8

Table 154: Object Description (209C<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0...15	0

Table 155: Entry Description (209C<sub>h</sub>)

#### 4.2.50 Object 209D<sub>h</sub>: Smart Energy Filter Enable

This object is used to set the StallGuard2 filter for more precision of the measurement. It reduces the measurement frequency to one measurement per four fullsteps if set.

In most cases it is expedient to set the filtered mode when using CoolStep™. Use the standard mode for step loss detection.

Possible values are:

- 0 – standard mode
- 1 – filtered mode



Object Description			
Index	Name	Object Type	Data Type
209D <sub>h</sub>	Smart Energy Filter Enable	Variable	UNSIGNED8

Table 156: Object Description (209D<sub>h</sub>)

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

Table 157: Entry Description (209D<sub>h</sub>)

#### 4.2.51 Object 209E<sub>h</sub>: StallGuard2 Threshold

This signed value controls the StallGuard2 threshold level for stall output and sets the optimum measurement range for readout. A lower value gives a higher sensitivity. Zero is the starting value. A higher value makes StallGuard2 less sensitive and requires more torque to indicate a stall.

Object Description			
Index	Name	Object Type	Data Type
209E <sub>h</sub>	StallGuard2 Threshold	Variable	SIGNED8

Table 158: Object Description (209E<sub>h</sub>)

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

Table 159: Entry Description (209E<sub>h</sub>)

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

This object is used to enable or to disable the short to ground protection. Normally there is no need to change this. Use the default value.

Possible values are:

- 0 – Short to GND protection enabled
- 1 – Short to GND protection disabled

Object Description			
Index	Name	Object Type	Data Type
20A1 <sub>h</sub>	Short Protection Disable	Variable	UNSIGNED8

Table 160: Object Description (20A1<sub>h</sub>)



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

Table 161: Entry Description (20A1<sub>h</sub>)

#### 4.2.53 Object 20A3<sub>h</sub>: Vsense

This object is used for setting the sense resistor voltage based current scaling. *Use the default value and change only when recommended by Trinamic.*

Possible settings are:

- 0 - Full scale sense resistor voltage is 1/18 VDD
- 1 - Full scale sense resistor voltage is 1/36 VDD

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

Table 162: Object Description (20A3<sub>h</sub>)

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

Table 163: Entry Description (20A3<sub>h</sub>)

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

Below this speed the motor will not be stopped. Above this speed the motor will be stopped in case the StallGuard2 load value reaches zero.

Object Description			
Index	Name	Object Type	Data Type
20A4 <sub>h</sub>	Stop on Stall	Variable	UNSIGNED32

Table 164: Object Description (20A4<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0...2147483647	0

Table 165: Entry Description (20A4<sub>h</sub>)



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

The CoolStep™ functionality will be enabled when the actual speed is above this speed. It will be disabled again when the actual speed drops below this value.

Object Description			
Index	Name	Object Type	Data Type
20A5 <sub>h</sub>	Smart Energy Threshold Speed	Variable	UNSIGNED32

Table 166: Object Description (20A5<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0...2147483647	0

Table 167: Entry Description (20A5<sub>h</sub>)

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

This object shows the home offset. The value is given in microsteps.

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

Table 168: Object Description (2100<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	no	-2147483648...2147483647	0

Table 169: Entry Description (2100<sub>h</sub>)

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

This object shows the actual load value used for stall detection (StallGuard2™).

Object Description			
Index	Name	Object Type	Data Type
2101 <sub>h</sub>	Actual Load Value	Variable	UNSIGNED16

Table 170: Object Description (2101<sub>h</sub>)



Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	no	0...1023	0

Table 171: Entry Description (2101<sub>h</sub>)

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

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

Error Flags		
Bit	Name	Meaning
7	OT	Overtemperature
6	OTPW	Temperature pre-warning
5	UV	Undervoltage
4	OCHS	Overcurrent high side
3	OLB	Open load on bridge B
2	OLA	Open load on bridge A
1	OCB	Overcurrent on bridge B
0	OCA	Overcurrent on bridge A

Table 172: Driver Error Flags (2102<sub>h</sub>)

Object Description			
Index	Name	Object Type	Data Type
2102 <sub>h</sub>	Driver Error Flags	Variable	UNSIGNED8

Table 173: Object Description (2102<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	no	0...255	0

Table 174: Entry Description (2102<sub>h</sub>)

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

This object shows the microstep resolution, set by object 2000<sub>h</sub> (please see section 4.2.1).



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

Table 175: Object Description (2107<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	no	0...8	8

Table 176: Entry Description (2107<sub>h</sub>)

#### 4.2.60 Object 210B<sub>h</sub>: Step Counter

This object shows the overall number of microsteps done by this motor so far. The value can be read as a 64 bit value (sub-index 3) or split into two 32 bit values (sub-index 1 and sub-index 2).

Object Description			
Index	Name	Object Type	Data Type
210B <sub>h</sub>	Step Counter	Variable	RECORD

Table 177: Object Description (210B<sub>h</sub>)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
1	Lower 32 Bits	ro	no	0...ffffffff <sub>h</sub>	0
2	Higher 32 Bits	ro	no	0...ffffffff <sub>h</sub>	0
3	64 Bit Value	ro	no	0...ffffffffffffffff <sub>h</sub>	0

Table 178: Entry Description (210B<sub>h</sub>)

#### 4.2.61 Object 2120<sub>h</sub>: Closed Loop Initialization Flag

The closed-loop initialization flag reads 1 when closed-loop is active and the initialization of closed-loop has finished. Otherwise this object reads 0.

Object Description			
Index	Name	Object Type	Data Type
2120 <sub>h</sub>	Closed Loop Initialization Flag	Variable	UNSIGNED8

Table 179: Object Description (2120<sub>h</sub>)



Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	no	0/1	—

Table 180: Entry Description (2120<sub>h</sub>)

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

After writing the make signature 656b616d<sub>h</sub> to this object the drive switches to TMCL mode. The drive can then only be controlled via TMCL commands written to the OS command object 1023/01<sub>h</sub>. The drive can only be reset to normal CANopen functionality using the DS-301 reset application command.

Object Description			
Index	Name	Object Type	Data Type
2700 <sub>h</sub>	TMCL Direct Communication	Variable	UNSIGNED32

Table 181: Object Description (2700<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	656b616d <sub>h</sub>	0

Table 182: Entry Description (2700)<sub>h</sub>

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

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

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

Object Description			
Index	Name	Object Type	Data Type
2701 <sub>h</sub>	Manufacturer Specific Mode	Variable	UNSIGNED32

Table 183: Object Description (2701<sub>h</sub>)





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

Table 184: Entry Description (2701)<sub>h</sub>

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

Bits 23...16 of this object reflect the states of the general purpose inputs of the module. The number of available inputs depends on the module type.

Bit Definitions	
Bit	Description
16	IN0
17	IN1
18	IN2
19	IN3
20	IN4
21	IN5
22	IN6
23	IN7

Table 185: Bit Definitions (2702)<sub>h</sub>

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

Table 186: Object Description (2702)<sub>h</sub>

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

Table 187: Entry Description (2702)<sub>h</sub>

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

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



number of available digital outputs depends on the module type.

Bit Definitions	
Bit	Description
16	OUT0
17	OUT1
18	OUT2
19	OUT3
20	OUT4
21	OUT5
22	OUT6
23	OUT7

Table 188: Bit Definitions (2703<sub>h</sub>)

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

Table 189: Object Description (2703<sub>h</sub>)

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

Table 190: Entry Description (2703<sub>h</sub>)

---

**Note** Some outputs can also be used for automatically controlling a brake that can be connected to the module via these outputs. In order to be able to control these outputs via this object the automatic brake control function has to be disabled. Do this by writing 0 to sub-index 1 and sub-index 2 of object 200C<sub>h</sub>.

---

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

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



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

Table 191: Object Description (2704<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	20,50,100,125,250,500,800,1000	1000

Table 192: Entry Description (2704)<sub>h</sub>

How to change the bit rate of a module:

- Write new bit rate in object 2704<sub>h</sub>.
- Write the save signature 65766173<sub>h</sub> to sub-index 1 of object 2706<sub>h</sub>.
- Reset module.

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

On modules that do not have address switches the node ID can be selected using this object. On modules with address switches the node ID is normally selected using the address switches. Only when the address switches are set to an invalid value (0 or >127) this object overrides the address switch setting. To change the node ID, first write the new node ID to this object. Then, store the new setting by writing the save signature to object 2706<sub>h</sub>. After that, reset the module. The new setting then becomes active.

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

Table 193: Object Description (2705<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	1...127	1

Table 194: Entry Description (2705)<sub>h</sub>

How to change the node ID of a module:

- Write new node ID to object 2705<sub>h</sub>.
- Write save signature 65766173<sub>h</sub> to sub-index 1 of object 2706<sub>h</sub>.
- Reset the module.



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

Writing the save signature to this object permanently saves changes made to objects 2704<sub>h</sub> and 2705<sub>h</sub>. The save signature is 65766173<sub>h</sub>.

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

Table 195: Object Description (2706<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	save signature: 65766173 <sub>h</sub>	1

Table 196: Entry Description (2706<sub>h</sub>)

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

This object shows the selected CAN bit rate.

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

Table 197: Object Description (2707<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	no	20...1000	depends on bit rate

Table 198: Entry Description (2707<sub>h</sub>)

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

This object shows the selected node ID.

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

Table 199: Object Description (2708<sub>h</sub>)



Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	no	1..127	Depends on node ID setting

Table 200: Entry Description (2708<sub>h</sub>)

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

This object shows the values of the analog inputs of the device.

Object Description			
Index	Name	Object Type	Data Type
270E <sub>h</sub>	Device Analog Inputs	Array	UNSIGNED32

Table 201: Object Description (270E<sub>h</sub>)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
1	Analog input 0	ro	yes	0...1023	-
2	Analog input 1	ro	yes	0...1023	-
3	Analog input 2	ro	yes	0...1023	-
4	Analog input 3	ro	yes	0...1023	-

Table 202: Entry Description (270E<sub>h</sub>)



## 5 Profile specific Area

The profile segment contains CiA-402 standard motion control objects. These objects control the motion control functions of the TMCM-3351. Since it is not possible to operate the modes in parallel, the user is able to activate the required function by selecting a mode of operation. The control device writes to the modes of operation object in order to select the operation mode. The drive device provides the modes of operation display object to indicate the actual activated operation mode. Controlword, statusword, and set-points are used mode-specific. This implies the responsibility of the control device to avoid inconsistencies and erroneous behavior.

The following operating modes (selectable via object 6060<sub>h</sub>, please see 5.1.6) are implemented on the TMCM-3351:

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

### Info

This section of the manual only shows the object indices for motor #0. Of course the same objects are also available for the other motors. For the other motors, add an offset of  $motor\_number \cdot 800_h$  to the object index. So for example the control word (object 6040<sub>h</sub> for motor #0) can be accessed as object 6840<sub>h</sub> for motor #1 and as object 7040<sub>h</sub> for motor #2. Please see also section 2.4.1.

Multi-axis Object Indices	
Motor	Object Index Range
Motor #0	6000 <sub>h</sub> – 67FF <sub>h</sub>
Motor #1	6800 <sub>h</sub> – 6FFF <sub>h</sub>
Motor #2	7000 <sub>h</sub> – 77FF <sub>h</sub>

Table 203: Multi-axis Object Indices (Profile specific Area)

## 5.1 Detailed Object Specifications

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

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



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

Table 204: Value Description (605A<sub>h</sub>)

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

Table 205: Object Description (605A<sub>h</sub>)

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

Table 206: Entry Description (605A<sub>h</sub>)

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

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

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

Table 207: Value Description (605B<sub>h</sub>)

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

Table 208: Object Description (605B<sub>h</sub>)



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

Table 209: Entry Description (605B<sub>h</sub>)

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

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

Value Definition	
Value	Definition
1	Slow down on slow down ramp

Table 210: Value Description (605C<sub>h</sub>)

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

Table 211: Object Description (605C<sub>h</sub>)

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

Table 212: Entry Description (605C<sub>h</sub>)

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

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

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

Table 213: Value Description (605D<sub>h</sub>)



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

Table 214: Object Description (605D<sub>h</sub>)

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

Table 215: Entry Description (605D<sub>h</sub>)

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

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

Value Definition	
Value	Definition
2	Slow down on quick stop ramp

Table 216: Value Description (605E<sub>h</sub>)

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

Table 217: Object Description (605E<sub>h</sub>)

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

Table 218: Entry Description (605E<sub>h</sub>)

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

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



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

Table 219: Value Description (6060<sub>h</sub>)

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

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

Table 220: Object Description (6060<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	refer to CiA-402	see table 219	0

Table 221: Entry Description (6060<sub>h</sub>)

---

**Note** It is not allowed to write the same mode number twice to object 6060<sub>h</sub>. So before writing a new value to object 6060<sub>h</sub>, either check object 6061<sub>h</sub> or object 6060<sub>h</sub> to see if the operating mode has not already been set to that value.

---

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

This object shows the operating mode that is currently set.



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

Table 222: Value Description (6061<sub>h</sub>)

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

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

Table 223: Object Description (6061<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	refer to CiA-402	see table 222	0

Table 224: Entry Description (6061<sub>h</sub>)

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

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

Value Definition	
Value	Mode
0	Encoder used
-1	No encoder

Table 225: Value Description (606A<sub>h</sub>)



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

Table 226: Object Description (606A<sub>h</sub>)

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

Table 227: Entry Description (606A<sub>h</sub>)

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

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

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

All values are dimensionless.

Object Description			
Index	Name	Object Type	Data Type
608F <sub>h</sub>	Position Encoder Resolution	Array	UNSIGNED32

Table 228: Object Description (608F<sub>h</sub>)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
0	Highest sub-index supported	ro	no	2	2
1	Encoder increments	rw	no	0...2147483647	1
2	Motor revolutions	ro	no	1	1

Table 229: Entry Description (608F<sub>h</sub>)

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

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



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

Table 230: Object Description (60FD<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	mappable	UNSIGNED32	0

Table 231: Entry Description (60FD<sub>h</sub>)

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

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

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

Table 232: Value Definition (6502<sub>h</sub>)

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

Table 233: Object Description (6502<sub>h</sub>)



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

*Table 234: Entry Description (6502<sub>h</sub>)*



## 6 Profile Position Mode

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

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

### 6.1 Detailed Object Specifications

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

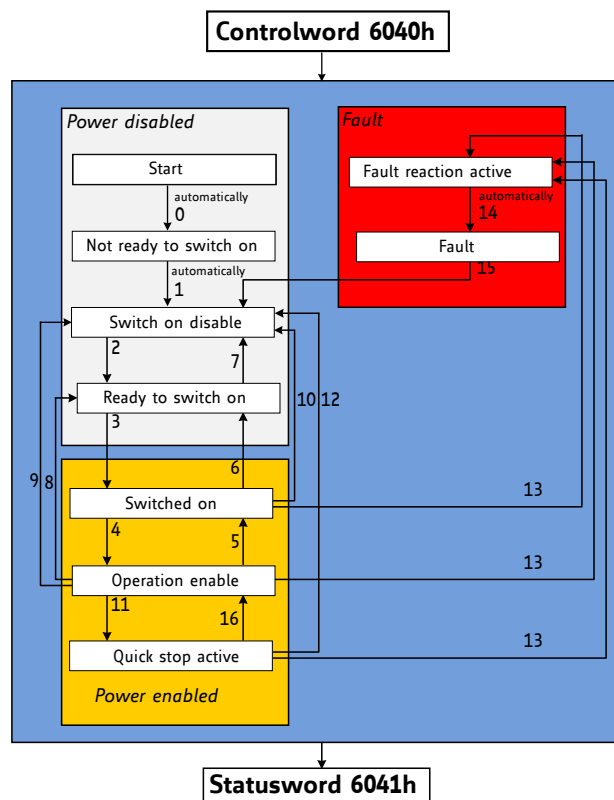


Figure 13: DS402 Finite State Machine

Notes on state transitions:

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



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

This object indicates the received command controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to figure 13 for detailed information.

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

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

Table 235: Structure of the Control Word in pp Mode

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

Table 236: Operation Mode specific Bits in pp Mode

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

Table 237: Command Coding





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

Table 238: Object Description (6040<sub>h</sub> in pp Mode)

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

Table 239: Entry Description (6040<sub>h</sub> in pp Mode)

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

This object provides the status of the PDS FSA. It reflects the status of the CiA-402 state machine. Please refer to figure 13 for detailed information. The object is structured as defined below. For more information about the coding please refer to the CANopen Drives and motion control device profile, part 2.

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

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

Table 240: Structure of the Staus Word in pp Mode

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

Table 241: Trinamic Specific Bits



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

Table 242: Operation Mode specific Bits in pp Mode

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

Table 243: State Coding

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

Table 244: Object Description (6041<sub>h</sub> in pp Mode)

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

Table 245: Entry Description (6041<sub>h</sub> in pp Mode)

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

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



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

Table 246: Object Description (6062<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	Refer to CiA402-3	SIGNED32	no

Table 247: Entry Description (6062<sub>h</sub>)

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

This object provides the actual value of the encoder or the motor. Please use the sensor selection object 606A<sub>h</sub> (see section 5.1.8) for selecting the motor or the encoder first. Object 6063<sub>h</sub> indicates the actual position of the encoder or the motor, re-scaled to the microstep resolution. The value is given in microsteps.

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

Table 248: Object Description (6063<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	Refer to CiA402-3	SIGNED32	no

Table 249: Entry Description (6063<sub>h</sub>)

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

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

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

Table 250: Object Description (6064<sub>h</sub>)



Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	Refer to CiA402-3	SIGNED32	no

Table 251: Entry Description (6064<sub>h</sub>)

### 6.1.6 Object 6065<sub>h</sub>: Following Error Window

This object indicates the configured range of tolerated position values symmetrically to the position demand value. If the position actual value is out of the following error window, a following error occurs. A following error may occur when a drive is blocked, unreachable profile velocity occurs, or at wrong closed-loop coefficients. The value shall be given in microsteps.

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

---

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

---

Object Description			
Index	Name	Object Type	Data Type
6065 <sub>h</sub>	Following Error Window	Variable	UNSIGNED32

Table 252: Object Description (6065<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0...2147483647	0

Table 253: Entry Description (6065<sub>h</sub>)

### 6.1.7 Object 6067<sub>h</sub>: Position Window

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



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

Table 254: Object Description (6067<sub>h</sub>)

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

Table 255: Entry Description (6067<sub>h</sub>)

### 6.1.8 Object 6068<sub>h</sub>: Position Window Time

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

Object Description			
Index	Name	Object Type	Data Type
6068 <sub>h</sub>	Position Window Time	Variable	UNSIGNED16

Table 256: Object Description (6068<sub>h</sub>)

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

Table 257: Entry Description (6068<sub>h</sub>)

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

This object shows the actual velocity value of the motor. The value is given in internal or user-defined velocity units (depending on object 208C<sub>h</sub>, described in section 4.2.37).

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

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

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	Refer to CiA402-3	SIGNED32	no

Table 259: Entry Description (606C<sub>h</sub>)

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

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

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

Table 260: Object Description (607A<sub>h</sub> in pp Mode)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	Refer to CiA402-3	SIGNED32	0

Table 261: Entry Description (607A<sub>h</sub> in pp Mode)

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

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

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

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

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

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



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

Table 263: Entry Description (607D<sub>h</sub>)

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

This object indicates the configured velocity normally attained at the end of the acceleration ramp during a profiled motion and is valid for both directions of motion. The profile velocity is the maximum velocity used when driving to a new position. It is given in internal or user specific units (depending on object 208C<sub>h</sub>, section 4.2.37).

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

Table 264: Object Description (6081<sub>h</sub>)

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

Table 265: Entry Description (6081<sub>h</sub>)

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

This object indicates the configured velocity normally attained at the end of the deceleration ramp during a profiled motion and is valid for both directions of motion. The end velocity is the velocity used when reaching the new position. It is given in internal or user specific units (depending on object 208C<sub>h</sub>, section 4.2.37).

Object Description			
Index	Name	Object Type	Data Type
6082 <sub>h</sub>	End Velocity	Variable	UNSIGNED32

Table 266: Object Description (6082<sub>h</sub>)

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

Table 267: Entry Description (6082<sub>h</sub>)



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

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

The units for object 6083<sub>h</sub> can be chosen with object 208E<sub>h</sub>, described in section 4.2.38.

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

Table 268: Object Description (6083<sub>h</sub>)

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

Table 269: Entry Description (6083<sub>h</sub>)

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

This object indicates the configured deceleration. Object 6084<sub>h</sub> sets the maximum deceleration to be used in profile positioning mode and in profile velocity mode.

The units for object 6084<sub>h</sub> can be chosen with object 208E<sub>h</sub>, described in section 4.2.38.

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

Table 270: Object Description (6084<sub>h</sub>)

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

Table 271: Entry Description (6084<sub>h</sub>)

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

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





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

Table 272: Object Description (6085<sub>h</sub>)

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

Table 273: Entry Description (6085<sub>h</sub>)

### 6.1.17 Object 6086<sub>h</sub>: Motion Profile Type

This object indicates the configured type of motion profile used to perform a profiled motion. Set this object to 0 to select trapezoid ramps or to 1 for S-shaped ramps.

Value Definition	
Value	Definition
0	Linear (six-point) ramp, trapezoidal profile
1	S-shaped ramp, defined by object 60A4 <sub>h</sub> (section 6.1.18)

Table 274: Value Definitions of Object 6086<sub>h</sub>

Object Description			
Index	Name	Object Type	Data Type
6086 <sub>h</sub>	Motion profile type	Variable	UNSIGNED16

Table 275: Object Description (6086<sub>h</sub>)

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

Table 276: Entry Description (6086<sub>h</sub>)

### 6.1.18 Object 60A4<sub>h</sub>: Profile Jerk

This object indicates the configured set of jerk parameters that shall be used during the profile movement. The bow of S-shaped ramps is set here. Object 60A4<sub>h</sub> can only be used when S-shaped ramps are selected by setting object 6086<sub>h</sub> (section 6.1.17) to 1.



**Note** The TMC4361 motion controller used on the TMC3351 module also offers bow values higher than the 32 bit range. Under most circumstances, the normal 32 bit value range is sufficient. Use object 2015<sub>h</sub> if higher values are required.

Object Description			
Index	Name	Object Type	Data Type
60A4 <sub>h</sub>	Profile Jerk	Array	UNSIGNED32

Table 277: Object Description (60A4<sub>h</sub>)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
0	Highest sub-index supported	ro	no	4	4
1	Profile jerk 1	rw	no	0...2147483647	0
2	Profile jerk 2	rw	no	0...2147483647	0
3	Profile jerk 3	rw	no	0...2147483647	0
4	Profile jerk 4	rw	no	0...2147483647	0

Table 278: Entry Description (60A4<sub>h</sub>)

### 6.1.19 Object 60F2<sub>h</sub>: Positioning Option Code

This object indicates the positioning behaviour in profile position mode. Only bits 0 and 1 (relative option) are supported.

Bit Definitions		
Bit 1	Bit 0	Definition
0	0	Positioning moves shall be performed relative to the preceding (internal absolute) target position.
0	1	Positioning moves shall be performed relative to the actual position demand value (object 6063 <sub>h</sub> ).
1	0	Positioning moves shall be performed relative to the position actual value (object 6064 <sub>h</sub> ).
1	1	reserved

Table 279: Bit Definitions of Object 60F2<sub>h</sub>



Object Description			
Index	Name	Object Type	Data Type
60F2 <sub>h</sub>	Positioning option code	Variable	UNSIGNED16

Table 280: Object Description (60F2<sub>h</sub>)

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

Table 281: Entry Description (60F2<sub>h</sub>)

## 6.2 How to move a Motor in pp Mode

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

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



## 7 Profile Velocity Mode

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

The profile velocity mode covers the following sub-functions:

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

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

- Profile velocity
- Profile acceleration
- Profile deceleration
- Emergency stop
- Motion profile type

### 7.1 Detailed Object Specifications

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

This object indicates the received command controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to figure 13 for detailed information.

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

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

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

Table 282: Structure of the Control Word in pv Mode



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

Table 283: Command Coding

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

Table 284: Object Description (6040<sub>h</sub> in pv Mode)

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

Table 285: Entry Description (6040<sub>h</sub> in pv Mode)

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

This object provides the status of the PDS FSA. It reflects the status of the CiA-402 state machine. Please refer to figure 13 for detailed information. The object is structured as defined below. For more information about the coding please refer to the CANopen Drives and motion control device profile, part 2.

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

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

Table 286: Structure of the Status Word in pv Mode



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

Table 287: Trinamic Specific Bits

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

Table 288: Operation Mode specific Bits in pv Mode

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

Table 289: State Coding

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

Table 290: Object Description (6041<sub>h</sub> in pv Mode)



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

Table 291: Entry Description (6041<sub>h</sub> in pv Mode)

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

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

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

Table 292: Object Description (6062<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	Refer to CiA402-3	SIGNED32	no

Table 293: Entry Description (6062<sub>h</sub>)

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

This object provides the actual value of the encoder or the motor. Please use the sensor selection object 606A<sub>h</sub> (see section 5.1.8) for selecting the motor or the encoder first. Object 6063<sub>h</sub> indicates the actual position of the encoder or the motor, re-scaled to the microstep resolution. The value is given in microsteps.

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

Table 294: Object Description (6063<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	Refer to CiA402-3	SIGNED32	no

Table 295: Entry Description (6063<sub>h</sub>)



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

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

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

Table 296: Object Description (6064<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	Refer to CiA402-3	SIGNED32	no

Table 297: Entry Description (6064<sub>h</sub>)

### 7.1.6 Object 6065<sub>h</sub>: Following Error Window

This object indicates the configured range of tolerated position values symmetrically to the position demand value. If the position actual value is out of the following error window, a following error occurs. A following error may occur when a drive is blocked, unreachable profile velocity occurs, or at wrong closed-loop coefficients. The value shall be given in microsteps.

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

---

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

---

Object Description			
Index	Name	Object Type	Data Type
6065 <sub>h</sub>	Following Error Window	Variable	UNSIGNED32

Table 298: Object Description (6065<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0...2147483647	0

Table 299: Entry Description (6065<sub>h</sub>)





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

This object shows the actual velocity value of the motor. The value is given in internal or user-defined velocity units (depending on object 208C<sub>h</sub>, described in section 4.2.37).

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

Table 300: Object Description (606C<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	Refer to CiA402-3	SIGNED32	no

Table 301: Entry Description (606C<sub>h</sub>)

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

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

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

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

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

Table 302: Object Description (607D<sub>h</sub>)

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

Table 303: Entry Description (607D<sub>h</sub>)



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

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

The units for object 6083<sub>h</sub> can be chosen with object 208E<sub>h</sub>, described in section 4.2.38.

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

Table 304: Object Description (6083<sub>h</sub>)

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

Table 305: Entry Description (6083<sub>h</sub>)

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

This object indicates the configured deceleration. Object 6084<sub>h</sub> sets the maximum deceleration to be used in profile positioning mode and in profile velocity mode.

The units for object 6084<sub>h</sub> can be chosen with object 208E<sub>h</sub>, described in section 4.2.38.

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

Table 306: Object Description (6084<sub>h</sub>)

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

Table 307: Entry Description (6084<sub>h</sub>)

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

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



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

Table 308: Object Description (6085<sub>h</sub>)

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

Table 309: Entry Description (6085<sub>h</sub>)

### 7.1.12 Object 6086<sub>h</sub>: Motion Profile Type

This object indicates the configured type of motion profile used to perform a profiled motion. Set this object to 0 to select trapezoid ramps or to 1 for S-shaped ramps.

Value Definition	
Value	Definition
0	Linear (six-point) ramp, trapezoidal profile
1	S-shaped ramp, defined by object 60A4 <sub>h</sub> (section 7.1.13)

Table 310: Value Definitions of Object 6086<sub>h</sub>

Object Description			
Index	Name	Object Type	Data Type
6086 <sub>h</sub>	Motion profile type	Variable	UNSIGNED16

Table 311: Object Description (6086<sub>h</sub>)

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

Table 312: Entry Description (6086<sub>h</sub>)

### 7.1.13 Object 60A4<sub>h</sub>: Profile Jerk

This object indicates the configured set of jerk parameters that shall be used during the profile movement. The bow of S-shaped ramps is set here. Object 60A4<sub>h</sub> can only be used when S-shaped ramps are selected by setting object 6086<sub>h</sub> (section 7.1.12) to 1.



**Note** The TMC4361 motion controller used on the TMCM-3351 module also offers bow values higher than the 32 bit range. Under most circumstances, the normal 32 bit value range is sufficient. Use object 2015<sub>h</sub> if higher values are required.

Object Description			
Index	Name	Object Type	Data Type
60A4 <sub>h</sub>	Profile Jerk	Array	UNSIGNED32

Table 313: Object Description (60A4<sub>h</sub>)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
0	Highest sub-index supported	ro	no	4	4
1	Profile jerk 1	rw	no	0...2147483647	0
2	Profile jerk 2	rw	no	0...2147483647	0
3	Profile jerk 3	rw	no	0...2147483647	0
4	Profile jerk 4	rw	no	0...2147483647	0

Table 314: Entry Description (60A4<sub>h</sub>)

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

This object indicates the configured target velocity and is used as input for the trajectory generator. Object 60FF<sub>h</sub> sets the target velocity when using profile velocity mode. The drive then accelerates or decelerates to that velocity using the acceleration and deceleration set by objects 6083<sub>h</sub> and 6084<sub>h</sub>. The values are given in units which can be selected with object 208C<sub>h</sub>, described in section 4.2.37.

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

Table 315: Object Description (60FF<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	see CiA402-3	SIGNED32	0

Table 316: Entry Description (60FF<sub>h</sub>)



## 7.2 How to move a Motor in pv Mode

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

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



## 8 Homing Mode

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

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

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

Figure 14 shows the defined input objects as well as the output objects. The user can specify the speeds, acceleration and method of homing. The home offset object 607C<sub>h</sub> allows displacing the zero in point the coordinate system for the home position.

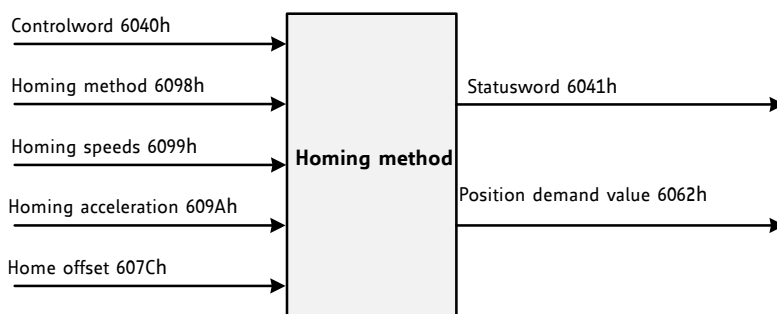


Figure 14: Homing Mode Function

Choosing a homing mode determines the following things:

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

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

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

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

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



## 8.1 Homing Methods

The TMCM-3351 supports a subset of different standard CANopen homing methods. The homing method that is to be used can be chosen via object 6098<sub>h</sub> (section 8.2.5).

Supported Homing Methods	
Method	Description
0	No homing (default value for object 6098 <sub>h</sub> ).
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 6062h, 6063h, and 6064h) are set to zero, but the motor will not move.

Table 317: Supported CANopen Homing Methods

When using homing methods that need end switch inputs or home switch inputs please take care of their configuration (object 2005<sub>h</sub>, section 4.2.6).

### 8.1.1 Homing 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.

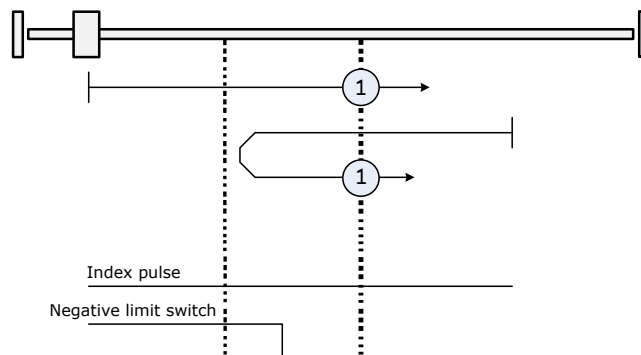


Figure 15: Homing Method 1



### 8.1.2 Homing 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.

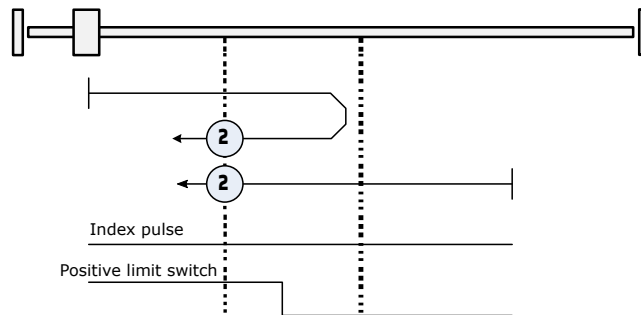


Figure 16: Homing Method 2

### 8.1.3 Homing 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.

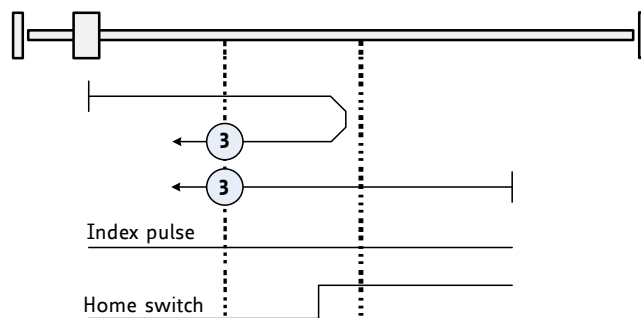


Figure 17: Homing Method 3

### 8.1.4 Homing 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.





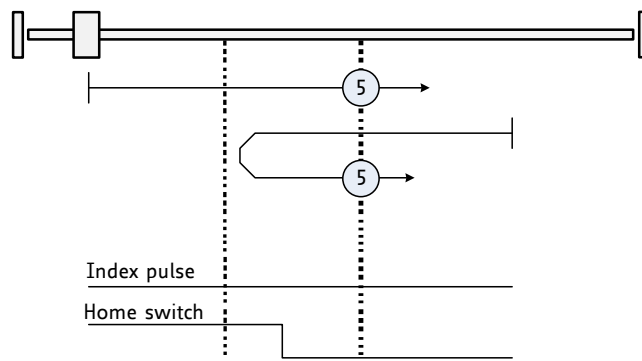


Figure 18: Homing Method 5

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

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

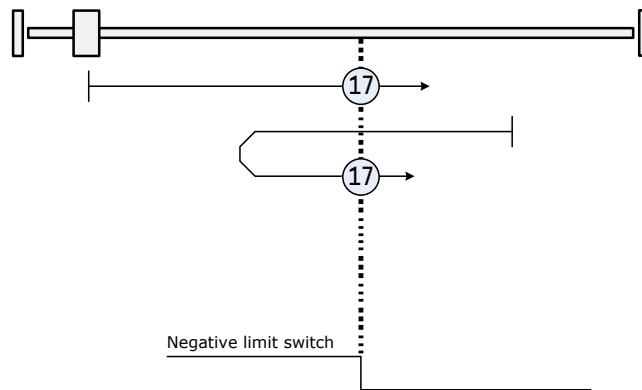


Figure 19: Homing Method 17

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

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

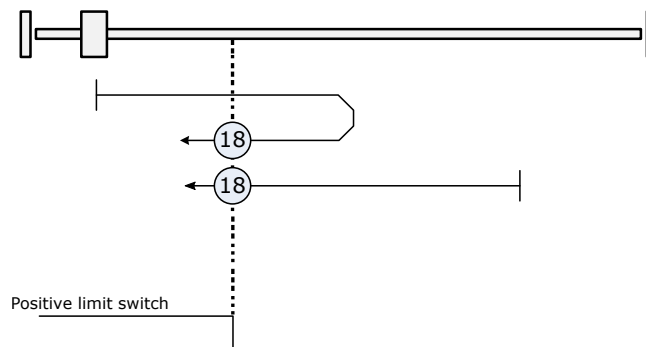


Figure 20: Homing Method 18



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

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

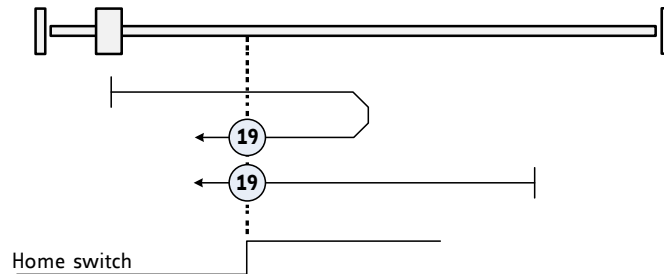


Figure 21: Homing Method 19

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

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

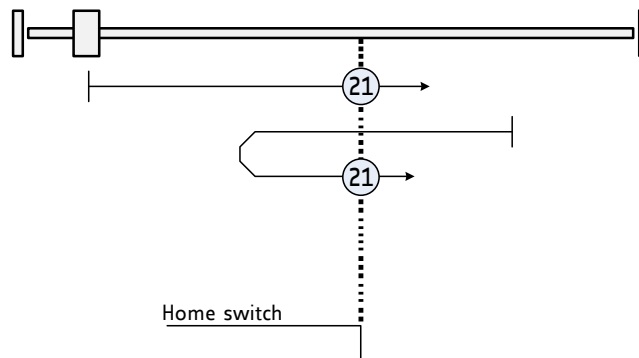


Figure 22: Homing Method 21

### 8.1.9 Homing 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.



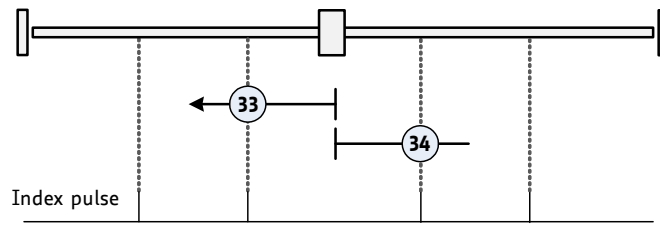


Figure 23: Homing Methods 33 and 34

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

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



## 8.2 Detailed Object Specifications

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

This object indicates the received command controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to figure 13 for detailed information.

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

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

Table 318: Structure of the Control Word in hm Mode

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

Table 319: Operation Mode specific Bits in hm Mode

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

Table 320: Command Coding



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

Table 321: Object Description (6040<sub>h</sub> in hm Mode)

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

Table 322: Entry Description (6040<sub>h</sub> in hm Mode)

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

This object provides the status of the PDS FSA. It reflects the status of the CiA-402 state machine. Please refer to figure 13 for detailed information. The object is structured as defined below. For more information about the coding please refer to the CANopen Drives and motion control device profile, part 2.

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

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

Table 323: Structure of the Status Word in hm Mode

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

Table 324: Trinamic Specific Bits



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

Table 325: Operation Mode specific Bits in hm Mode

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

Table 326: State Coding

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

Table 327: Object Description (6041<sub>h</sub> in hm Mode)

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

Table 328: Entry Description (6041<sub>h</sub> in hm Mode)

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

This object shows the actual velocity value of the motor. The value is given in internal or user-defined velocity units (depending on object 208C<sub>h</sub>, described in section 4.2.37).



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

Table 329: Object Description (606C<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	Refer to CiA402-3	SIGNED32	no

Table 330: Entry Description (606C<sub>h</sub>)

### 8.2.4 Object 607C<sub>h</sub>: Home Offset

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

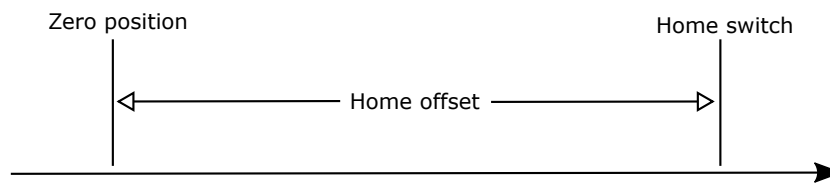


Figure 24: Home Offset

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

Table 331: Object Description (607C<sub>h</sub>)

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

Table 332: Entry Description (607C<sub>h</sub>)



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

The homing method to be used can be selected by writing to this object. Please see table 317 for a list of homing methods supported by the current version of the TMCM-3351 CANopen firmware.

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

Table 333: Object Description (6098<sub>h</sub>)

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

Table 334: Entry Description (6098<sub>h</sub>)

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

This object indicates the configured speeds used during homing procedure. The values are given in pps units or internal units selectable with object 208C<sub>h</sub> (section 4.2.37). Using object 6099<sub>h</sub> a fast and a slow homing speed can be set. In most homing modes, the home switch is searched with the fast speed first. When the home switch has been found, the motor will be decelerated to the slow speed (using the homing acceleration, object 609A<sub>h</sub>) to search for the exact switch point. When the switch point has been found the motor will be stopped at that point.

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

Table 335: Object Description (6099<sub>h</sub>)

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

Table 336: Entry Description (6099<sub>h</sub>)

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

This object indicates the configured acceleration and deceleration to be used during homing operation. The value is given in units selected by object 208E<sub>h</sub> (section 4.2.38).





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

Table 337: Object Description (609A<sub>h</sub>)

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

Table 338: Entry Description (609A<sub>h</sub>)

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

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

- Select hm mode by writing 6 to object 6060<sub>h</sub>.
- Write 6 to object 6040<sub>h</sub> to switch to READY\_TO\_SWITCH\_ON state.
- Write 7 to object 6040<sub>h</sub> to switch to SWITCHED\_ON state.
- Write 15 to object 6040<sub>h</sub> to switch to OPERATION\_ENABLED state.
- Select homing method 19 by writing 19 to object 6098<sub>h</sub>.
- Set the homing speeds by writing e.g. 50000 to object 6099<sub>h</sub> sub index 1 and e.g. 10000 to object 6099<sub>h</sub> sub index 2.
- Write 31 to object 6040<sub>h</sub> to start the homing process.
- Press and release the home switch.
- When homing has finished, write 15 to object 6040<sub>h</sub> again.



## 9 Cyclic synchronous Position Mode

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

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

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

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

### 9.1 Detailed Object Specifications

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

This object indicates the received command controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to figure 13 for detailed information. The cyclic synchronous position mode does not use any mode specific bits of the control word.

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

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

Table 339: Structure of the Control Word in csp Mode



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

Table 340: Command Coding

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

Table 341: Object Description (6040<sub>h</sub> in csp Mode)

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

Table 342: Entry Description (6040<sub>h</sub> in csp Mode)

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

This object provides the status of the PDS FSA. It reflects the status of the CiA-402 state machine. Please refer to figure 13 for detailed information. The object is structured as defined below. For more information about the coding please refer to the CANopen Drives and motion control device profile, part 2.

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

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

Table 343: Structure of the Status Word in csp Mode



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

Table 344: Trinamic Specific Bits

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

Table 345: Operation Mode specific Bits in csp Mode

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

Table 346: State Coding

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

Table 347: Object Description (6041<sub>h</sub> in csp Mode)



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

Table 348: Entry Description (6041<sub>h</sub> in csp Mode)

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

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

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

Table 349: Object Description (6062<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	Refer to CiA402-3	SIGNED32	no

Table 350: Entry Description (6062<sub>h</sub>)

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

This object provides the actual value of the encoder or the motor. Please use the sensor selection object 606A<sub>h</sub> (see section 5.1.8) for selecting the motor or the encoder first. Object 6063<sub>h</sub> indicates the actual position of the encoder or the motor, re-scaled to the microstep resolution. The value is given in microsteps.

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

Table 351: Object Description (6063<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	Refer to CiA402-3	SIGNED32	no

Table 352: Entry Description (6063<sub>h</sub>)



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

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

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

Table 353: Object Description (6064<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	Refer to CiA402-3	SIGNED32	no

Table 354: Entry Description (6064<sub>h</sub>)

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

This object shows the actual velocity value of the motor. The value is given in internal or user-defined velocity units (depending on object 208C<sub>h</sub>, described in section 4.2.37).

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

Table 355: Object Description (606C<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	Refer to CiA402-3	SIGNED32	no

Table 356: Entry Description (606C<sub>h</sub>)

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

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

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

Table 357: Object Description (607A<sub>h</sub> in csp Mode)



Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	Refer to CiA402-3	SIGNED32	0

Table 358: Entry Description (607A<sub>h</sub> in csp Mode)

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

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

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

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

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

Table 359: Object Description (607D<sub>h</sub>)

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

Table 360: Entry Description (607D<sub>h</sub>)

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

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

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

Table 361: Object Description (60B0<sub>h</sub>)

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

Table 362: Entry Description (60B0<sub>h</sub>)

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

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

Object Description			
Index	Name	Object Type	Data Type
60C2 <sub>h</sub>	Offset Torque	Vecord	Interpolation time period record (0080 <sub>h</sub> )

Table 363: Object Description (60C2<sub>h</sub>)

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

Table 364: Entry Description (60C2<sub>h</sub>)



## 10 Cyclic synchronous Velocity Mode

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

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

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

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

### 10.1 Detailed Object Specifications

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

This object indicates the received command controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to figure 13 for detailed information. The cyclic synchronous velocity mode does not use any mode specific bits of the control word.

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

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

Table 365: Structure of the Control Word in csv Mode



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

Table 366: Command Coding

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

Table 367: Object Description (6040<sub>h</sub> in csv Mode)

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

Table 368: Entry Description (6040<sub>h</sub> in csv Mode)

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

This object provides the status of the PDS FSA. It reflects the status of the CiA-402 state machine. Please refer to figure 13 for detailed information. The object is structured as defined below. For more information about the coding please refer to the CANopen Drives and motion control device profile, part 2.

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

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

Table 369: Structure of the Status Word in csv Mode



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

Table 370: Trinamic Specific Bits

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

Table 371: Operation Mode specific Bits in csv Mode

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

Table 372: State Coding

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

Table 373: Object Description (6041<sub>h</sub> in csv Mode)



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

Table 374: Entry Description (6041<sub>h</sub> in csv Mode)

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

This object shows the actual velocity value of the motor. The value is given in internal or user-defined velocity units (depending on object 208C<sub>h</sub>, described in section 4.2.37).

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

Table 375: Object Description (606C<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	Refer to CiA402-3	SIGNED32	no

Table 376: Entry Description (606C<sub>h</sub>)

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

In csv mode the target velocity specifies the velocity that is to be reached within the interpolation time period. The values are given in units which can be selected with object 208C<sub>h</sub>, described in section 4.2.37.

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

Table 377: Object Description (60FF<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	see CiA402-3	SIGNED32	0

Table 378: Entry Description (60FF<sub>h</sub>)

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

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



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

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

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

Table 379: Object Description (607D<sub>h</sub>)

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

Table 380: Entry Description (607D<sub>h</sub>)

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

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

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

Table 381: Object Description (60B1<sub>h</sub>)

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

Table 382: Entry Description (60B1<sub>h</sub>)

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

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



Object Description			
Index	Name	Object Type	Data Type
60C2 <sub>h</sub>	Offset Torque	Vecord	Interpolation time period record (0080 <sub>h</sub> )

Table 383: Object Description (60C2<sub>h</sub>)

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

Table 384: Entry Description (60C2<sub>h</sub>)



## 11 Cyclic synchronous Torque Mode

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

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

### 11.1 Detailed Object Specifications

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

This object indicates the received command controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to figure 13 for detailed information. The cyclic synchronous torque mode does not use any mode specific bits of the control word.

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

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

Table 385: Structure of the Control Word in cst Mode

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

Table 386: Command Coding



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

Table 387: Object Description (6040<sub>h</sub> in cst Mode)

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

Table 388: Entry Description (6040<sub>h</sub> in cst Mode)

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

This object provides the status of the PDS FSA. It reflects the status of the CiA-402 state machine. Please refer to figure 13 for detailed information. The object is structured as defined below. For more information about the coding please refer to the CANopen Drives and motion control device profile, part 2.

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

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

Table 389: Structure of the Status Word in cst Mode

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

Table 390: Trinamic Specific Bits





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

Table 391: Operation Mode specific Bits in cst Mode

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

Table 392: State Coding

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

Table 393: Object Description (6041<sub>h</sub> in cst Mode)

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

Table 394: Entry Description (6041<sub>h</sub> in cst Mode)

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

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



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

Table 395: Object Description (6062<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	Refer to CiA402-3	SIGNED32	no

Table 396: Entry Description (6062<sub>h</sub>)

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

This object provides the actual value of the encoder or the motor. Please use the sensor selection object 606A<sub>h</sub> (see section 5.1.8) for selecting the motor or the encoder first. Object 6063<sub>h</sub> indicates the actual position of the encoder or the motor, re-scaled to the microstep resolution. The value is given in microsteps.

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

Table 397: Object Description (6063<sub>h</sub>)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	Refer to CiA402-3	SIGNED32	no

Table 398: Entry Description (6063<sub>h</sub>)

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

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

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

Table 399: Object Description (6064<sub>h</sub>)



Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	Refer to CiA402-3	SIGNED32	no

Table 400: Entry Description (6064<sub>h</sub>)

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

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

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

Table 401: Object Description (6071<sub>h</sub>)

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

Table 402: Entry Description (6071<sub>h</sub>)

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

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

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

Table 403: Object Description (6077<sub>h</sub>)

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

Table 404: Entry Description (6077<sub>h</sub>)

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

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



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

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

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

Table 405: Object Description (607D<sub>h</sub>)

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

Table 406: Entry Description (607D<sub>h</sub>)

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

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

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

Table 407: Object Description (60B2<sub>h</sub>)

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

Table 408: Entry Description (60B2<sub>h</sub>)

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

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



Object Description			
Index	Name	Object Type	Data Type
60C2 <sub>h</sub>	Offset Torque	Vecord	Interpolation time period record (0080 <sub>h</sub> )

*Table 409: Object Description (60C2<sub>h</sub>)*

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

*Table 410: Entry Description (60C2<sub>h</sub>)*



## 12 Emergency Messages (EMCY)

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

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

Emergency Messages (EMCY) of the TMCM-3351						
Error code	Additional byte					Description
	1	2	3	4	5	
0000 <sub>h</sub>	0	0...2	0	0	0	<b>Fault reset</b> The fault reset command has been executed.
1000 <sub>h</sub>	1	0...2	0	0	0	<b>Generic error: open load bridge A</b> The motor driver indicates open load on bridge A. It is possible that the motor cable is broken or that there is an error in the power amplifier itself.
1000 <sub>h</sub>	2	0...2	0	0	0	<b>Generic error: open load bridge B</b> The motor driver indicates open load on bridge B. It is possible that the motor cable is broken or that there is an error in the power amplifier itself.
2310 <sub>h</sub>	0	0...2	0	0	0	<b>Overcurrent high side</b> The motor driver indicates an overcurrent on the high side. This can be caused by a short circuit in the driver stage.
2311 <sub>h</sub>	0	0...2	0	0	0	<b>Overcurrent bridge B</b> The motor driver indicates that there is overcurrent on bridge B. This can be caused by a short circuit in the motor itself or in the motor driver stage.
2312 <sub>h</sub>	0	0...2	0	0	0	<b>Overcurrent bridge A</b> The motor driver indicates that there is overcurrent on bridge A. This can be caused by a short circuit in the motor itself or in the motor driver stage.
3230 <sub>h</sub>	0	0...2	0	0	0	<b>stallGuard2 error</b> The actual load value exceeds the stallGuard2 limit.
4310 <sub>h</sub>	1	0...2	0	0	0	<b>Overtemperature pre-warning</b> The temperature in the motor driver exceeds the pre-warning limit.
4310 <sub>h</sub>	2	0...2	0	0	0	<b>Overtemperature error</b> The motor driver has been switched off because the temperature limit has been exceeded.
5441 <sub>h</sub>	0	255	0	0	0	<b>Shutdown switch active</b> The enable signal is missing (due to the shutdown switch) and the motor driver has been switched off.
6320 <sub>h</sub>	0	255	0	0	0	<b>Parameter error</b> The data in the received PDO is either wrong or cannot be accepted due to the internal state of the drive.



Error code	Additional byte					Description
	1	2	3	4	5	
8100 <sub>h</sub>	0	255	0	0	0	<b>Communication error</b> General CAN bus communication error.
8110 <sub>h</sub>	1	255	0	0	0	<b>CAN controller overflow</b> The receive message buffer of the CAN controller hardware is full and some CAN messages are lost.
8110 <sub>h</sub>	2	255	0	0	0	<b>CAN Tx buffer overflow</b> The software CAN transmit buffer is full and thus some CAN messages are lost.
8110 <sub>h</sub>	3	255	0	0	0	<b>CAN Rx buffer overflow</b> The software CAN receive buffer is full and so some CAN messages are lost.
8120 <sub>h</sub>	0	255	0	0	0	<b>CAN error passive</b> The CAN controller has detected communication errors and has entered the CAN Error passive state.
8130 <sub>h</sub>	0	255	0	0	0	<b>Heartbeat or lifeguard error</b> The module did not receive a heartbeat or lifeguard message in time.
8140 <sub>h</sub>	0	255	0	0	0	<b>CAN controller recovered from bus-off state</b> The CAN controller has detected too many errors and has changed into the bus-off state. The drive has been stopped and disabled. This message is sent after the CAN controller has recovered from bus-off state and is bus-on again.
8210 <sub>h</sub>	0	255	0	0	0	<b>PDO not processed due to length error</b> A PDO sent to the module could not be processed because too few bytes were supplied.
8220 <sub>h</sub>	0	255	0	0	0	<b>PDO length exceeded</b> A PDO sent to the module could not be processed because too many bytes were supplied.
8611 <sub>h</sub>	0	0...2	0	0	0	<b>Following error</b> The deviation between motor position counter and encoder position counter has exceeded the following error window.
ff00 <sub>h</sub>	0	0...2	0	0	0	<b>Undervoltage</b> The supply voltage is too low to drive a motor.
ff01 <sub>h</sub>	1	0...2	0	0	0	<b>Positive software limit</b> The actual position is outside the range defined by object 607d <sub>h</sub> .
ff01 <sub>h</sub>	2	0...2	0	0	0	<b>Negative software limit</b> The actual position is outside the range defined by object 607d <sub>h</sub> .
ff01 <sub>h</sub>	3	0...2	0	0	0	<b>Positive limit switch</b> The positive limit switch has been touched outside of the homing function.



Error code	Additional byte					Description
	1	2	3	4	5	
ff01 <sub>h</sub>	4	0...2	0	0	0	<b>Negative limit switch</b> The negative limit switch has been touched outside of the homing function.

Table 411: Emergency Messages (EMCY)





## 13 Figures Index

1	stallGuard2 Load Measurement as a Function of Load . . . . .	9	12	Brake Output Timing . . . . .	42
2	Energy Efficiency Example with CoolStep . . . . .	10	13	DS402 Finite State Machine . . . . .	83
3	Closed-Loop Control Scheme . . . . .	12	14	Homing Mode Function . . . . .	106
4	Load Angle Control Parameter . . . . .	13	15	Homing Method 1 . . . . .	107
5	Current Level Control . . . . .	13	16	Homing Method 2 . . . . .	108
6	Field Weakening . . . . .	14	17	Homing Method 3 . . . . .	108
7	Position Catch up . . . . .	15	18	Homing Method 5 . . . . .	109
8	NMT State Machine . . . . .	19	19	Homing Method 17 . . . . .	109
9	Communication Architecture . . . . .	20	20	Homing Method 18 . . . . .	109
10	Device Model . . . . .	21	21	Homing Method 19 . . . . .	110
11	CoolStep Adjustment Points and Thresholds . . . . .	39	22	Homing Method 21 . . . . .	110
			23	Homing Methods 33 and 34 . . . . .	111
			24	Home Offset . . . . .	115



## 14 Tables Index

1	Abbreviations used in this Manual . . .	8	53	Object Description (1A00 <sub>h</sub> ) . . . . .	36
2	Closed-Loop Example Settings . . . . .	11	54	Entry Description (1A00 <sub>h</sub> ) . . . . .	37
3	PID Mode Example Settings . . . . .	16	55	Multi-axis Object Indices (Manufac- turer specific Area) . . . . .	38
4	Service Primitives . . . . .	17	56	CoolStep related Objects . . . . .	40
5	Service Types . . . . .	18	57	Object Description (2000 <sub>h</sub> ) . . . . .	41
6	Object Dictionary . . . . .	21	58	Entry Description (2000 <sub>h</sub> ) . . . . .	41
7	Multi-Axis Object Indices . . . . .	22	59	Object Description (2001 <sub>h</sub> ) . . . . .	41
8	Object Description (1000 <sub>h</sub> ) . . . . .	23	60	Entry Description (2001 <sub>h</sub> ) . . . . .	41
9	Entry Description (1000 <sub>h</sub> ) . . . . .	23	61	Object Description (2002 <sub>h</sub> ) . . . . .	42
10	Object Description (1001 <sub>h</sub> ) . . . . .	23	62	Entry Description (2002 <sub>h</sub> ) . . . . .	42
11	Entry Description (1001 <sub>h</sub> ) . . . . .	23	63	Object Description (2003 <sub>h</sub> ) . . . . .	42
12	Error Register Bits . . . . .	24	64	Entry Description (2003 <sub>h</sub> ) . . . . .	42
13	Value Definition (1005 <sub>h</sub> ) . . . . .	24	65	Object Description (2004 <sub>h</sub> ) . . . . .	43
14	Object Description (1005 <sub>h</sub> ) . . . . .	24	66	Entry Description (2004 <sub>h</sub> ) . . . . .	43
15	Entry Description (1005 <sub>h</sub> ) . . . . .	24	67	Object Description (2005 <sub>h</sub> ) . . . . .	43
16	Object Description (1008 <sub>h</sub> ) . . . . .	25	68	Entry Description (2005 <sub>h</sub> ) . . . . .	43
17	Entry Description (1008 <sub>h</sub> ) . . . . .	25	69	Bit Definitions (2005 <sub>h</sub> ) . . . . .	44
18	Object Description (1009 <sub>h</sub> ) . . . . .	25	70	Object Description (200A <sub>h</sub> ) . . . . .	44
19	Entry Description (1009 <sub>h</sub> ) . . . . .	25	71	Entry Description (200A <sub>h</sub> ) . . . . .	44
20	Object Description (100A <sub>h</sub> ) . . . . .	25	72	Object Description (200B <sub>h</sub> ) . . . . .	44
21	Entry Description (100A <sub>h</sub> ) . . . . .	26	73	Entry Description (200B <sub>h</sub> ) . . . . .	45
22	Object Description (100C <sub>h</sub> ) . . . . .	26	74	Object Description (200C <sub>h</sub> ) . . . . .	45
23	Entry Description (100C <sub>h</sub> ) . . . . .	26	75	Entry Description (200C <sub>h</sub> ) . . . . .	45
24	Object Description (100D <sub>h</sub> ) . . . . .	26	76	Object Description (200F <sub>h</sub> ) . . . . .	45
25	Entry Description (100D <sub>h</sub> ) . . . . .	26	77	Entry Description (200F <sub>h</sub> ) . . . . .	46
26	Save Signature . . . . .	27	78	Object Description (2010 <sub>h</sub> ) . . . . .	46
27	Object Description (1010 <sub>h</sub> ) . . . . .	27	79	Entry Description (2010 <sub>h</sub> ) . . . . .	46
28	Entry Description (1010 <sub>h</sub> ) . . . . .	28	80	Object Description (2011 <sub>h</sub> ) . . . . .	46
29	Load Signature . . . . .	28	81	Entry Description (2011 <sub>h</sub> ) . . . . .	46
30	Object Description (1011 <sub>h</sub> ) . . . . .	29	82	Object Description (2012 <sub>h</sub> ) . . . . .	47
31	Entry Description (1011 <sub>h</sub> ) . . . . .	29	83	Entry Description (2012 <sub>h</sub> ) . . . . .	47
32	Object Description (1014 <sub>h</sub> ) . . . . .	29	84	Object Description (2013 <sub>h</sub> ) . . . . .	47
33	Entry Description (1014 <sub>h</sub> ) . . . . .	29	85	Entry Description (2013 <sub>h</sub> ) . . . . .	47
34	Object Description (1015 <sub>h</sub> ) . . . . .	30	86	Object Description (2014 <sub>h</sub> ) . . . . .	47
35	Entry Description (1015 <sub>h</sub> ) . . . . .	30	87	Entry Description (2014 <sub>h</sub> ) . . . . .	48
36	Value Definition (1016 <sub>h</sub> ) . . . . .	30	88	Object Description (2015 <sub>h</sub> ) . . . . .	48
37	Object Description (1016 <sub>h</sub> ) . . . . .	30	89	Entry Description (2015 <sub>h</sub> ) . . . . .	48
38	Entry Description (1016 <sub>h</sub> ) . . . . .	30	90	Object Description (2020 <sub>h</sub> ) . . . . .	48
39	Object Description (1017 <sub>h</sub> ) . . . . .	31	91	Entry Description (2020 <sub>h</sub> ) . . . . .	48
40	Entry Description (1017 <sub>h</sub> ) . . . . .	31	92	Object Description (2021 <sub>h</sub> ) . . . . .	49
41	Object Description (1018 <sub>h</sub> ) . . . . .	31	93	Entry Description (2021 <sub>h</sub> ) . . . . .	49
42	Entry Description (1018 <sub>h</sub> ) . . . . .	31	94	Object Description (2022 <sub>h</sub> ) . . . . .	49
43	Object Description (1023 <sub>h</sub> ) . . . . .	32	95	Entry Description (2022 <sub>h</sub> ) . . . . .	49
44	Entry Description (1023 <sub>h</sub> ) . . . . .	32	96	Object Description (2027 <sub>h</sub> ) . . . . .	49
45	Object Description (1029 <sub>h</sub> ) . . . . .	32	97	Entry Description (2027 <sub>h</sub> ) . . . . .	50
46	Entry Description (1029 <sub>h</sub> ) . . . . .	32	98	Object Description (2028 <sub>h</sub> ) . . . . .	50
47	Object Description (1400 <sub>h</sub> ) . . . . .	33	99	Entry Description (2028 <sub>h</sub> ) . . . . .	50
48	Entry Description (1400 <sub>h</sub> ) . . . . .	33	100	Object Description (2029 <sub>h</sub> ) . . . . .	50
49	Object Description (1600 <sub>h</sub> ) . . . . .	34	101	Entry Description (2029 <sub>h</sub> ) . . . . .	50
50	Entry Description (1600 <sub>h</sub> ) . . . . .	34	102	Object Description (202A <sub>h</sub> ) . . . . .	51
51	Object Description (1800 <sub>h</sub> ) . . . . .	35	103	Entry Description (202A <sub>h</sub> ) . . . . .	51
52	Entry Description (1800 <sub>h</sub> ) . . . . .	36			



104	Object Description (202B <sub>h</sub> ) . . . . .	51	158	Object Description (209E <sub>h</sub> ) . . . . .	63
105	Entry Description (202B <sub>h</sub> ) . . . . .	51	159	Entry Description (209E <sub>h</sub> ) . . . . .	63
106	Object Description (202C <sub>h</sub> ) . . . . .	51	160	Object Description (20A1 <sub>h</sub> ) . . . . .	63
107	Entry Description (202C <sub>h</sub> ) . . . . .	52	161	Entry Description (20A1 <sub>h</sub> ) . . . . .	64
108	Object Description (202D <sub>h</sub> ) . . . . .	52	162	Object Description (20A3 <sub>h</sub> ) . . . . .	64
109	Entry Description (202D <sub>h</sub> ) . . . . .	52	163	Entry Description (20A3 <sub>h</sub> ) . . . . .	64
110	Object Description (202E <sub>h</sub> ) . . . . .	52	164	Object Description (20A4 <sub>h</sub> ) . . . . .	64
111	Entry Description (202E <sub>h</sub> ) . . . . .	52	165	Entry Description (20A4 <sub>h</sub> ) . . . . .	64
112	Object Description (202F <sub>h</sub> ) . . . . .	53	166	Object Description (20A5 <sub>h</sub> ) . . . . .	65
113	Entry Description (202F <sub>h</sub> ) . . . . .	53	167	Entry Description (20A5 <sub>h</sub> ) . . . . .	65
114	Object Description (2030 <sub>h</sub> ) . . . . .	53	168	Object Description (2100 <sub>h</sub> ) . . . . .	65
115	Entry Description (2030 <sub>h</sub> ) . . . . .	53	169	Entry Description (2100 <sub>h</sub> ) . . . . .	65
116	Object Description (2031 <sub>h</sub> ) . . . . .	53	170	Object Description (2101 <sub>h</sub> ) . . . . .	65
117	Entry Description (2031 <sub>h</sub> ) . . . . .	54	171	Entry Description (2101 <sub>h</sub> ) . . . . .	66
118	Object Description (2033 <sub>h</sub> ) . . . . .	54	172	Driver Error Flags (2102 <sub>h</sub> ) . . . . .	66
119	Entry Description (2033 <sub>h</sub> ) . . . . .	54	173	Object Description (2102 <sub>h</sub> ) . . . . .	66
120	Object Description (2034 <sub>h</sub> ) . . . . .	54	174	Entry Description (2102 <sub>h</sub> ) . . . . .	66
121	Entry Description (2034 <sub>h</sub> ) . . . . .	54	175	Object Description (2107 <sub>h</sub> ) . . . . .	67
122	Object Description (2035 <sub>h</sub> ) . . . . .	55	176	Entry Description (2107 <sub>h</sub> ) . . . . .	67
123	Entry Description (2035 <sub>h</sub> ) . . . . .	55	177	Object Description (210B <sub>h</sub> ) . . . . .	67
124	Object Description (2036 <sub>h</sub> ) . . . . .	55	178	Entry Description (210B <sub>h</sub> ) . . . . .	67
125	Entry Description (2036 <sub>h</sub> ) . . . . .	55	179	Object Description (2120 <sub>h</sub> ) . . . . .	67
126	Object Description (204E <sub>h</sub> ) . . . . .	56	180	Entry Description (2120 <sub>h</sub> ) . . . . .	68
127	Entry Description (204E <sub>h</sub> ) . . . . .	56	181	Object Description (2700 <sub>h</sub> ) . . . . .	68
128	Object Description (2089 <sub>h</sub> ) . . . . .	56	182	Entry Description (2700 <sub>h</sub> ) . . . . .	68
129	Entry Description (2089 <sub>h</sub> ) . . . . .	56	183	Object Description (2701 <sub>h</sub> ) . . . . .	68
130	Object Description (208C <sub>h</sub> ) . . . . .	57	184	Entry Description (2701 <sub>h</sub> ) . . . . .	69
131	Entry Description (208C <sub>h</sub> ) . . . . .	57	185	Bit Definitions (2702 <sub>h</sub> ) . . . . .	69
132	Object Description (208E <sub>h</sub> ) . . . . .	57	186	Object Description (2702 <sub>h</sub> ) . . . . .	69
133	Entry Description (208E <sub>h</sub> ) . . . . .	57	187	Entry Description (2702 <sub>h</sub> ) . . . . .	69
134	Object Description (2092 <sub>h</sub> ) . . . . .	57	188	Bit Definitions (2703 <sub>h</sub> ) . . . . .	70
135	Entry Description (2092 <sub>h</sub> ) . . . . .	58	189	Object Description (2703 <sub>h</sub> ) . . . . .	70
136	Object Description (2093 <sub>h</sub> ) . . . . .	58	190	Entry Description (2703 <sub>h</sub> ) . . . . .	70
137	Entry Description (2093 <sub>h</sub> ) . . . . .	58	191	Object Description (2704 <sub>h</sub> ) . . . . .	71
138	Object Description (2094 <sub>h</sub> ) . . . . .	58	192	Entry Description (2704 <sub>h</sub> ) . . . . .	71
139	Entry Description (2094 <sub>h</sub> ) . . . . .	58	193	Object Description (2705 <sub>h</sub> ) . . . . .	71
140	Object Description (2095 <sub>h</sub> ) . . . . .	59	194	Entry Description (2705 <sub>h</sub> ) . . . . .	71
141	Entry Description (2095 <sub>h</sub> ) . . . . .	59	195	Object Description (2706 <sub>h</sub> ) . . . . .	72
142	Object Description (2096 <sub>h</sub> ) . . . . .	59	196	Entry Description (2706 <sub>h</sub> ) . . . . .	72
143	Entry Description (2096 <sub>h</sub> ) . . . . .	59	197	Object Description (2707 <sub>h</sub> ) . . . . .	72
144	Object Description (2097 <sub>h</sub> ) . . . . .	60	198	Entry Description (2707 <sub>h</sub> ) . . . . .	72
145	Entry Description (2097 <sub>h</sub> ) . . . . .	60	199	Object Description (2708 <sub>h</sub> ) . . . . .	72
146	Object Description (2098 <sub>h</sub> ) . . . . .	60	200	Entry Description (2708 <sub>h</sub> ) . . . . .	73
147	Entry Description (2098 <sub>h</sub> ) . . . . .	60	201	Object Description (270E <sub>h</sub> ) . . . . .	73
148	Object Description (2099 <sub>h</sub> ) . . . . .	61	202	Entry Description (270E <sub>h</sub> ) . . . . .	73
149	Entry Description (2099 <sub>h</sub> ) . . . . .	61	203	Multi-axis Object Indices (Profile specific Area) . . . . .	74
150	Object Description (209A <sub>h</sub> ) . . . . .	61	204	Value Description (605A <sub>h</sub> ) . . . . .	75
151	Entry Description (209A <sub>h</sub> ) . . . . .	61	205	Object Description (605A <sub>h</sub> ) . . . . .	75
152	Object Description (209B <sub>h</sub> ) . . . . .	62	206	Entry Description (605A <sub>h</sub> ) . . . . .	75
153	Entry Description (209B <sub>h</sub> ) . . . . .	62	207	Value Description (605B <sub>h</sub> ) . . . . .	75
154	Object Description (209C <sub>h</sub> ) . . . . .	62	208	Object Description (605B <sub>h</sub> ) . . . . .	75
155	Entry Description (209C <sub>h</sub> ) . . . . .	62	209	Entry Description (605B <sub>h</sub> ) . . . . .	76
156	Object Description (209D <sub>h</sub> ) . . . . .	63	210	Value Description (605C <sub>h</sub> ) . . . . .	76
157	Entry Description (209D <sub>h</sub> ) . . . . .	63			



211	Object Description (605C <sub>h</sub> ) . . . . .	76	262	Object Description (607D <sub>h</sub> ) . . . . .	90
212	Entry Description (605C <sub>h</sub> ) . . . . .	76	263	Entry Description (607D <sub>h</sub> ) . . . . .	91
213	Value Description (605D <sub>h</sub> ) . . . . .	76	264	Object Description (6081 <sub>h</sub> ) . . . . .	91
214	Object Description (605D <sub>h</sub> ) . . . . .	77	265	Entry Description (6081 <sub>h</sub> ) . . . . .	91
215	Entry Description (605D <sub>h</sub> ) . . . . .	77	266	Object Description (6082 <sub>h</sub> ) . . . . .	91
216	Value Description (605E <sub>h</sub> ) . . . . .	77	267	Entry Description (6082 <sub>h</sub> ) . . . . .	91
217	Object Description (605E <sub>h</sub> ) . . . . .	77	268	Object Description (6083 <sub>h</sub> ) . . . . .	92
218	Entry Description (605E <sub>h</sub> ) . . . . .	77	269	Entry Description (6083 <sub>h</sub> ) . . . . .	92
219	Value Description (6060 <sub>h</sub> ) . . . . .	78	270	Object Description (6084 <sub>h</sub> ) . . . . .	92
220	Object Description (6060 <sub>h</sub> ) . . . . .	78	271	Entry Description (6084 <sub>h</sub> ) . . . . .	92
221	Entry Description (6060 <sub>h</sub> ) . . . . .	78	272	Object Description (6085 <sub>h</sub> ) . . . . .	93
222	Value Description (6061 <sub>h</sub> ) . . . . .	79	273	Entry Description (6085 <sub>h</sub> ) . . . . .	93
223	Object Description (6061 <sub>h</sub> ) . . . . .	79	274	Value Definitions of Object 6086 <sub>h</sub> . . . . .	93
224	Entry Description (6061 <sub>h</sub> ) . . . . .	79	275	Object Description (6086 <sub>h</sub> ) . . . . .	93
225	Value Description (606A <sub>h</sub> ) . . . . .	79	276	Entry Description (6086 <sub>h</sub> ) . . . . .	93
226	Object Description (606A <sub>h</sub> ) . . . . .	80	277	Object Description (60A4 <sub>h</sub> ) . . . . .	94
227	Entry Description (606A <sub>h</sub> ) . . . . .	80	278	Entry Description (60A4 <sub>h</sub> ) . . . . .	94
228	Object Description (608F <sub>h</sub> ) . . . . .	80	279	Bit Definitions of Object 60F2 <sub>h</sub> . . . . .	94
229	Entry Description (608F <sub>h</sub> ) . . . . .	80	280	Object Description (60F2 <sub>h</sub> ) . . . . .	95
230	Object Description (60FD <sub>h</sub> ) . . . . .	81	281	Entry Description (60F2 <sub>h</sub> ) . . . . .	95
231	Entry Description (60FD <sub>h</sub> ) . . . . .	81	282	Structure of the Control Word in pv Mode . . . . .	96
232	Value Definition (6502 <sub>h</sub> ) . . . . .	81	283	Command Coding . . . . .	97
233	Object Description (6502 <sub>h</sub> ) . . . . .	81	284	Object Description (6040 <sub>h</sub> in pv Mode) . . . . .	97
234	Entry Description (6502 <sub>h</sub> ) . . . . .	82	285	Entry Description (6040 <sub>h</sub> in pv Mode) . . . . .	97
235	Structure of the Control Word in pp Mode . . . . .	84	286	Structure of the Status Word in pv Mode . . . . .	97
236	Operation Mode specific Bits in pp Mode . . . . .	84	287	Trinamic Specific Bits . . . . .	98
237	Command Coding . . . . .	84	288	Operation Mode specific Bits in pv Mode . . . . .	98
238	Object Description (6040 <sub>h</sub> in pp Mode) . . . . .	85	289	State Coding . . . . .	98
239	Entry Description (6040 <sub>h</sub> in pp Mode) . . . . .	85	290	Object Description (6041 <sub>h</sub> in pv Mode) . . . . .	98
240	Structure of the Staus Word in pp Mode . . . . .	85	291	Entry Description (6041 <sub>h</sub> in pv Mode) . . . . .	99
241	Trinamic Specific Bits . . . . .	85	292	Object Description (6062 <sub>h</sub> ) . . . . .	99
242	Operation Mode specific Bits in pp Mode . . . . .	86	293	Entry Description (6062 <sub>h</sub> ) . . . . .	99
243	State Coding . . . . .	86	294	Object Description (6063 <sub>h</sub> ) . . . . .	99
244	Object Description (6041 <sub>h</sub> in pp Mode) . . . . .	86	295	Entry Description (6063 <sub>h</sub> ) . . . . .	99
245	Entry Description (6041 <sub>h</sub> in pp Mode) . . . . .	86	296	Object Description (6064 <sub>h</sub> ) . . . . .	100
246	Object Description (6062 <sub>h</sub> ) . . . . .	87	297	Entry Description (6064 <sub>h</sub> ) . . . . .	100
247	Entry Description (6062 <sub>h</sub> ) . . . . .	87	298	Object Description (6065 <sub>h</sub> ) . . . . .	100
248	Object Description (6063 <sub>h</sub> ) . . . . .	87	299	Entry Description (6065 <sub>h</sub> ) . . . . .	100
249	Entry Description (6063 <sub>h</sub> ) . . . . .	87	300	Object Description (606C <sub>h</sub> ) . . . . .	101
250	Object Description (6064 <sub>h</sub> ) . . . . .	87	301	Entry Description (606C <sub>h</sub> ) . . . . .	101
251	Entry Description (6064 <sub>h</sub> ) . . . . .	88	302	Object Description (607D <sub>h</sub> ) . . . . .	101
252	Object Description (6065 <sub>h</sub> ) . . . . .	88	303	Entry Description (607D <sub>h</sub> ) . . . . .	101
253	Entry Description (6065 <sub>h</sub> ) . . . . .	88	304	Object Description (6083 <sub>h</sub> ) . . . . .	102
254	Object Description (6067 <sub>h</sub> ) . . . . .	89	305	Entry Description (6083 <sub>h</sub> ) . . . . .	102
255	Entry Description (6067 <sub>h</sub> ) . . . . .	89	306	Object Description (6084 <sub>h</sub> ) . . . . .	102
256	Object Description (6068 <sub>h</sub> ) . . . . .	89	307	Entry Description (6084 <sub>h</sub> ) . . . . .	102
257	Entry Description (6068 <sub>h</sub> ) . . . . .	89	308	Object Description (6085 <sub>h</sub> ) . . . . .	103
258	Object Description (606C <sub>h</sub> ) . . . . .	89	309	Entry Description (6085 <sub>h</sub> ) . . . . .	103
259	Entry Description (606C <sub>h</sub> ) . . . . .	90	310	Value Definitions of Object 6086 <sub>h</sub> . . . . .	103
260	Object Description (607A <sub>h</sub> in pp Mode) . . . . .	90	311	Object Description (6086 <sub>h</sub> ) . . . . .	103
261	Entry Description (607A <sub>h</sub> in pp Mode) . . . . .	90	312	Entry Description (6086 <sub>h</sub> ) . . . . .	103
			313	Object Description (60A4 <sub>h</sub> ) . . . . .	104
			314	Entry Description (60A4 <sub>h</sub> ) . . . . .	104



315	Object Description (60FF <sub>h</sub> ) . . . . .	104	362	Entry Description (60B0 <sub>h</sub> ) . . . . .	124
316	Entry Description (60FF <sub>h</sub> ) . . . . .	104	363	Object Description (60C2 <sub>h</sub> ) . . . . .	124
317	Supported CANopen Homing Methods	107	364	Entry Description (60C2 <sub>h</sub> ) . . . . .	124
318	Structure of the Control Word in hm Mode . . . . .	112	365	Structure of the Control Word in csv Mode . . . . .	125
319	Operation Mode specific Bits in hm Mode . . . . .	112	366	Command Coding . . . . .	126
320	Command Coding . . . . .	112	367	Object Description (6040 <sub>h</sub> in csv Mode)	126
321	Object Description (6040 <sub>h</sub> in hm Mode)	113	368	Entry Description (6040 <sub>h</sub> in csv Mode)	126
322	Entry Description (6040 <sub>h</sub> in hm Mode)	113	369	Structure of the Status Word in csv Mode . . . . .	126
323	Structure of the Status Word in hm Mode . . . . .	113	370	Trinamic Specific Bits . . . . .	127
324	Trinamic Specific Bits . . . . .	113	371	Operation Mode specific Bits in csv Mode . . . . .	127
325	Operation Mode specific Bits in hm Mode . . . . .	114	372	State Coding . . . . .	127
326	State Coding . . . . .	114	373	Object Description (6041 <sub>h</sub> in csv Mode)	127
327	Object Description (6041 <sub>h</sub> in hm Mode)	114	374	Entry Description (6041 <sub>h</sub> in csv Mode)	128
328	Entry Description (6041 <sub>h</sub> in hm Mode)	114	375	Object Description (606C <sub>h</sub> ) . . . . .	128
329	Object Description (606C <sub>h</sub> ) . . . . .	115	376	Entry Description (606C <sub>h</sub> ) . . . . .	128
330	Entry Description (606C <sub>h</sub> ) . . . . .	115	377	Object Description (60FF <sub>h</sub> ) . . . . .	128
331	Object Description (607C <sub>h</sub> ) . . . . .	115	378	Entry Description (60FF <sub>h</sub> ) . . . . .	128
332	Entry Description (607C <sub>h</sub> ) . . . . .	115	379	Object Description (607D <sub>h</sub> ) . . . . .	129
333	Object Description (6098 <sub>h</sub> ) . . . . .	116	380	Entry Description (607D <sub>h</sub> ) . . . . .	129
334	Entry Description (6098 <sub>h</sub> ) . . . . .	116	381	Object Description (60B1 <sub>h</sub> ) . . . . .	129
335	Object Description (6099 <sub>h</sub> ) . . . . .	116	382	Entry Description (60B1 <sub>h</sub> ) . . . . .	129
336	Entry Description (6099 <sub>h</sub> ) . . . . .	116	383	Object Description (60C2 <sub>h</sub> ) . . . . .	130
337	Object Description (609A <sub>h</sub> ) . . . . .	117	384	Entry Description (60C2 <sub>h</sub> ) . . . . .	130
338	Entry Description (609A <sub>h</sub> ) . . . . .	117	385	Structure of the Control Word in cst Mode . . . . .	131
339	Structure of the Control Word in csp Mode . . . . .	118	386	Command Coding . . . . .	131
340	Command Coding . . . . .	119	387	Object Description (6040 <sub>h</sub> in cst Mode)	132
341	Object Description (6040 <sub>h</sub> in csp Mode)	119	388	Entry Description (6040 <sub>h</sub> in cst Mode)	132
342	Entry Description (6040 <sub>h</sub> in csp Mode)	119	389	Structure of the Status Word in cst Mode	132
343	Structure of the Status Word in csp Mode . . . . .	119	390	Trinamic Specific Bits . . . . .	132
344	Trinamic Specific Bits . . . . .	120	391	Operation Mode specific Bits in cst Mode . . . . .	133
345	Operation Mode specific Bits in csp Mode . . . . .	120	392	State Coding . . . . .	133
346	State Coding . . . . .	120	393	Object Description (6041 <sub>h</sub> in cst Mode)	133
347	Object Description (6041 <sub>h</sub> in csp Mode)	120	394	Entry Description (6041 <sub>h</sub> in cst Mode)	133
348	Entry Description (6041 <sub>h</sub> in csp Mode)	121	395	Object Description (6062 <sub>h</sub> ) . . . . .	134
349	Object Description (6062 <sub>h</sub> ) . . . . .	121	396	Entry Description (6062 <sub>h</sub> ) . . . . .	134
350	Entry Description (6062 <sub>h</sub> ) . . . . .	121	397	Object Description (6063 <sub>h</sub> ) . . . . .	134
351	Object Description (6063 <sub>h</sub> ) . . . . .	121	398	Entry Description (6063 <sub>h</sub> ) . . . . .	134
352	Entry Description (6063 <sub>h</sub> ) . . . . .	121	399	Object Description (6064 <sub>h</sub> ) . . . . .	134
353	Object Description (6064 <sub>h</sub> ) . . . . .	122	400	Entry Description (6064 <sub>h</sub> ) . . . . .	135
354	Entry Description (6064 <sub>h</sub> ) . . . . .	122	401	Object Description (6071 <sub>h</sub> ) . . . . .	135
355	Object Description (606C <sub>h</sub> ) . . . . .	122	402	Entry Description (6071 <sub>h</sub> ) . . . . .	135
356	Entry Description (606C <sub>h</sub> ) . . . . .	122	403	Object Description (6077 <sub>h</sub> ) . . . . .	135
357	Object Description (607A <sub>h</sub> in csp Mode)	122	404	Entry Description (6077 <sub>h</sub> ) . . . . .	135
358	Entry Description (607A <sub>h</sub> in csp Mode)	123	405	Object Description (607D <sub>h</sub> ) . . . . .	136
359	Object Description (607D <sub>h</sub> ) . . . . .	123	406	Entry Description (607D <sub>h</sub> ) . . . . .	136
360	Entry Description (607D <sub>h</sub> ) . . . . .	123	407	Object Description (60B2 <sub>h</sub> ) . . . . .	136
361	Object Description (60B0 <sub>h</sub> ) . . . . .	123	408	Entry Description (60B2 <sub>h</sub> ) . . . . .	136
			409	Object Description (60C2 <sub>h</sub> ) . . . . .	137
			410	Entry Description (60C2 <sub>h</sub> ) . . . . .	137



411 Emergency Messages (EMCY) . . . . .	140	413 Document Revision . . . . .	149
412 Firmware Revision . . . . .	149		



## 15 Supplemental Directives

### 15.1 Producer Information

### 15.2 Copyright

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

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

### 15.3 Trademark Designations and Symbols

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

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

### 15.4 Target User

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

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

### 15.5 Disclaimer: Life Support Systems

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

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

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

### 15.6 Disclaimer: Intended Use

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





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

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

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

## 15.7 Collateral Documents & Tools

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





## 16 Revision History

### 16.1 Firmware Revision

Version	Date	Author	Description
3.21	2017-DEC-14	OK	First release.
3.22	2019-FEB-11	OK	CSV mode added. Internal performance optimized. Automatic brake control switched off by default (default value of object 200C <sub>h</sub> sub-index 2 now 0).

Table 412: Firmware Revision

### 16.2 Document Revision

Version	Date	Author	Description
1.00	2017-DEC-14	OK	First release.
1.01	2019-FEB-14	OK	CSV mode added. Changes of firmware V3.22 added.

Table 413: Document Revision

