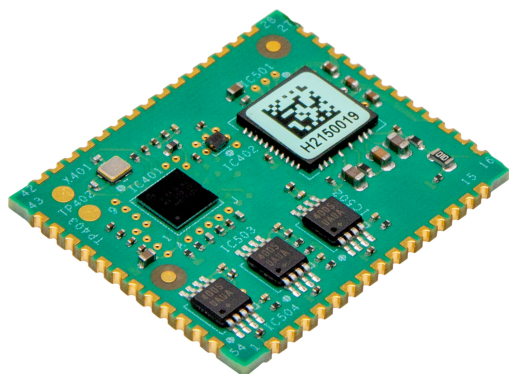


TMCM-1690 CANopen® Firmware Manual

Firmware Version V1.01 | YYY-XXXXXX; Rev 0; 07/23

The TMCM-1690 is a single axis FOC servo controller gate driver module for 3-phase BLDC and DC motors with up to 1.5A gate drive current and +60V (+48 V nominal) supply. It offers UART (RS232-/RS485-ready), CAN and EtherCAT® interfaces with TMCL, CANopen, or CANopen-over-EtherCAT protocol support for communication. TMCM-1690 supports incremental encoders, digital hall sensors, and absolute encoder as position feedback.



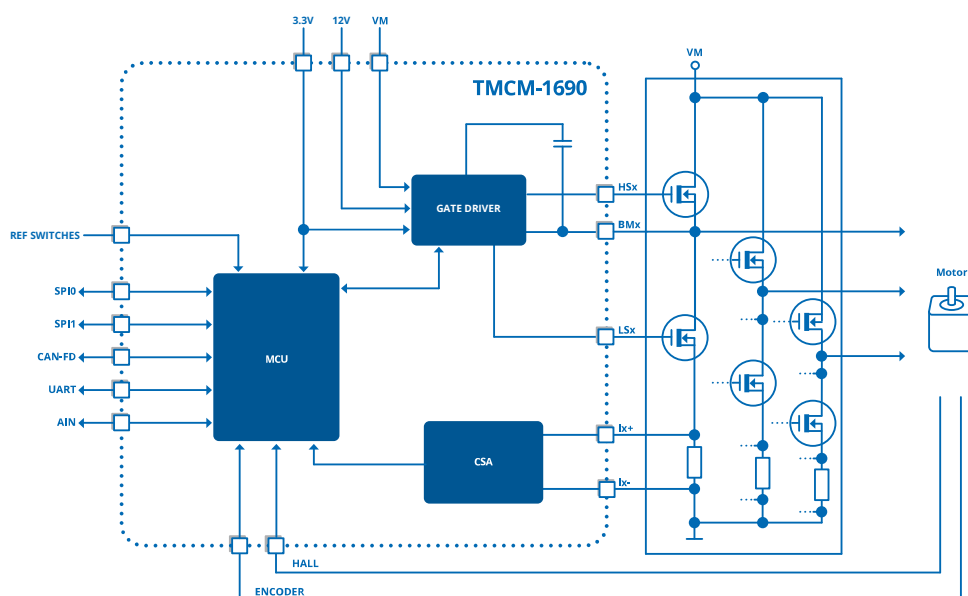
Features

- Supply voltage +10 to +60V DC
- FOC servo controller gate driver module for BLDC and DC motors
- 0.5A/1.0A/1.5A gate drive current
- Up to 120kHz PWM frequency
- Onboard current-sense amplifiers
- Supports UART (RS232/RS485-ready), CAN and EtherCAT™ interface
- Supports incremental encoders (ABN), digital HALL sensors, absolute SPI encoders
- Reference switch inputs
- Compact size 27mm x 22.5mm

Applications

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

Simplified Block Diagram



Contents

1	Preface	5
1.1	General Features of this CANopen Implementation	5
1.2	Abbreviations used in this Manual	6
1.3	Firmware Update	6
2	Motor Regulation	7
2.1	Structure of Motor Control Regulation Modes	7
2.2	Current Regulation	7
2.2.1	Structure of the Current Regulator	8
2.3	Velocity Regulation	9
2.3.1	Structure of the Velocity Regulator	9
2.4	Velocity Ramp Generator	10
2.5	Position Regulation	10
2.5.1	Structure of the Position Regulator	10
2.5.2	Correlation of Target Position and the Position End Flag	11
3	Ramps	12
3.1	Linear Ramp	12
3.2	Sine shaped ramp	12
4	Module specific functions	14
4.1	Filters	14
4.1.1	Biquad filter (For future use)	15
4.2	Mechanical Brake	15
4.3	Brake chopper	16
4.3.1	PWM braking	16
4.3.2	Resistive/Shunt braking	16
4.4	IIT	17
4.5	Lower Velocity PI	18
5	Communication	19
5.1	Reference Model	19
5.2	NMT State Machine	21
5.3	Device Model	22
5.4	Object Dictionary	23
6	Communication Area	25
6.1	Detailed Object Specifications	25
6.1.1	Object 1000 _h : Device Type	25
6.1.2	Object 1001 _h : Error Register	25
6.1.3	Object 1005 _h : COB-ID SYNC Message	26
6.1.4	Object 1008 _h : Manufacturer Device Name	27
6.1.5	Object 1009 _h : Manufacturer Hardware Version	27
6.1.6	Object 100A _h : Manufacturer Software Version	27
6.1.7	Object 100C _h : Guard Time	28
6.1.8	Object 100D _h : Life Time Factor	28
6.1.9	Object 1010 _h : Store Parameters	28
6.1.10	Object 1011 _h : Restore Parameters	30
6.1.11	Object 1014 _h : COB-ID Emergency Object	31
6.1.12	Object 1015 _h : Inhibit Time EMCY	31
6.1.13	Object 1016 _h : Consumer Heartbeat Time	32
6.1.14	Object 1017 _h : Producer Heartbeat Time	33
6.1.15	Object 1018 _h : Identity Object	33

6.1.16	Object 1029 _h : Error Behaviour	34
6.1.17	Objects 1400 _h – 1403 _h : Receive PDO Communication Parameter	34
6.1.18	Objects 1600 _h – 1603 _h : Receive PDO Mapping Parameter	35
6.1.19	Objects 1800 _h – 1803 _h : Transmit PDO Communication Parameter	36
6.1.20	Objects 1A00 _h – 1A03 _h : Transmit PDO Mapping Parameter	37

7 Manufacturer specific Area 39

7.1	Detailed Object Specifications	39
7.1.1	Object 2000 _h : Motor settings	39
7.1.2	Object 2003 _h : Drive settings	40
7.1.3	Object 2005 _h : ADC settings	41
7.1.4	Object 200A _h : Open loop settings	42
7.1.5	Object 200F _h : Digital hall settings	42
7.1.6	Object 2014 _h : ABN encoder settings	44
7.1.7	Object 2015 _h : ABN encoder 2 settings	46
7.1.8	Object 2019 _h : Absolute encoder settings	47
7.1.9	Object 201E _h : Torque mode	48
7.1.10	Object 2023 _h : Velocity mode	49
7.1.11	Object 2028 _h : Position mode	50
7.1.12	Object 202B _h : Gearbox	51
7.1.13	Object 202D _h : IIT	52
7.1.14	Object 2032 _h : Windows	53
7.1.15	Object 2037 _h : Ramp	54
7.1.16	Object 203C _h : Homing	54
7.1.17	Object 2041 _h : Filter target torque	55
7.1.18	Object 2046 _h : Filter actual torque	56
7.1.19	Object 204B _h : Filter target velocity	57
7.1.20	Object 2050 _h : Filter actual velocity	57
7.1.21	Object 2055 _h : Filter target position	58
7.1.22	Object 205A _h : Mechanical brake	59
7.1.23	Object 205F _h : Brake chopper	60
7.1.24	Object 2064 _h : Reference switch	61
7.1.25	Object 2069 _h : Monitoring	62
7.1.26	Object 206B _h : Lower velocity PI	62
7.1.27	Object 206C _h : Linear drive settings	63
7.1.28	Object 206D _h : Driver status	63
7.1.29	Object 206E _h : Controller initialize	64
7.1.30	Object 207D _h : Switch parameters	64
7.1.31	Object 2082 _h : Home offset display	64
7.1.32	Object 2701 _h : Device digital IO direction	65
7.1.33	Object 2702 _h : Device digital inputs	65
7.1.34	Object 2703 _h : Device digital outputs	66
7.1.35	Object 2704 _h : CAN bit rate	67
7.1.36	Object 2705 _h : Node ID	67
7.1.37	Object 2707 _h : CAN bit rate load	68
7.1.38	Object 2708 _h : Node ID load	68
7.1.39	Object 270B _h : HAL version	69
7.1.40	Object 270E _h : Device analog inputs	69

8 Profile specific Area 70

8.1	Detailed Object Specifications	70
8.1.1	Object 605A _h : Quick Stop Option Code	70
8.1.2	Object 605B _h : Shutdown Option Code	71
8.1.3	Object 605C _h : Disable Operation Option Code	71

8.1.4	Object 605D _h : Halt Option Code	72
8.1.5	Object 605E _h : Fault Reaction Option Code	72
8.1.6	Object 6060 _h : Modes of Operation	73
8.1.7	Object 6061 _h : Modes of Operation Display	74
8.1.8	Object 60FD _h : Digital Inputs	75
8.1.9	Object 6502 _h : Supported Drive Modes	75
8.1.10	Object 67FF _h : Single Device Type	76
9	Profile Position Mode	77
9.1	Detailed Object Specifications	77
9.1.1	Object 6040 _h : Controlword	78
9.1.2	Object 6041 _h : Statusword	80
9.1.3	Object 6062 _h : Position Demand Value	81
9.1.4	Object 6063 _h : Position Actual Internal Value	82
9.1.5	Object 6064 _h : Position Actual Value	82
9.1.6	Object 6067 _h : Position Window	82
9.1.7	Object 6068 _h : Position Window Time	83
9.1.8	Object 606C _h : Velocity Actual Value	83
9.1.9	Object 607A _h : Target Position	84
9.1.10	Object 607D _h : Software Position Limit	84
9.1.11	Object 6081 _h : Profile Velocity	85
9.1.12	Object 6083 _h : Profile Acceleration	85
9.1.13	Object 6084 _h : Profile Deceleration	86
9.1.14	Object 6085 _h : Quick Stop Deceleration	86
9.2	How to move a Motor in pp Mode	87
10	Profile Velocity Mode	88
10.0.1	Object 6040 _h : Controlword	88
10.0.2	Object 6041 _h : Statusword	89
10.0.3	Object 6062 _h : Position Demand Value	91
10.0.4	Object 6063 _h : Position Actual Internal Value	91
10.0.5	Object 6064 _h : Position Actual Value	92
10.0.6	Object 606C _h : Velocity Actual Value	92
10.0.7	Object 607D _h : Software Position Limit	92
10.0.8	Object 6083 _h : Profile Acceleration	93
10.0.9	Object 6085 _h : Quick Stop Deceleration	94
10.0.10	Object 60FF _h : Target Velocity	94
10.1	How to move a Motor in pv Mode	94
11	Homing Mode	96
11.1	Homing Methods	97
11.1.1	Homing Method 17: Homing on negative Limit Switch	97
11.1.2	Homing Method 18: Homing on positive Limit Switch	97
11.1.3	Homing Method 19: Homing on positive Home Switch	98
11.1.4	Homing Method 21: Homing on negative Home Switch	98
11.1.5	Homing Method 35: Current Position as Home Position	99
11.2	Detailed Object Specifications	100
11.2.1	Object 6040 _h : Controlword	100
11.2.2	Object 6041 _h : Statusword	101
11.2.3	Object 606C _h : Velocity Actual Value	102
11.2.4	Object 607C _h : Home Offset	103
11.2.5	Object 6098 _h : Homing Method	104
11.2.6	Object 6099 _h : Homing Speeds	104
11.2.7	Object 609A _h : Homing Acceleration	104

11.3 How to start a Homing in hm Mode	105
12 Cyclic synchronous Position Mode	106
12.1 Detailed Object Specifications	106
12.1.1 Object 6040 _h : Controlword	106
12.1.2 Object 6041 _h : Statusword	107
12.1.3 Object 6062 _h : Position Demand Value	109
12.1.4 Object 6063 _h : Position Actual Internal Value	109
12.1.5 Object 6064 _h : Position Actual Value	110
12.1.6 Object 606C _h : Velocity Actual Value	110
12.1.7 Object 607A _h : Target Position	110
12.1.8 Object 607D _h : Software Position Limit	111
12.1.9 Object 60B0 _h : Position Offset	111
12.1.10 Object 60C2 _h : Interpolation Time Period	112
13 Cyclic synchronous Velocity Mode	113
13.1 Detailed Object Specifications	113
13.1.1 Object 6040 _h : Controlword	113
13.1.2 Object 6041 _h : Statusword	114
13.1.3 Object 606C _h : Velocity Actual Value	116
13.1.4 Object 60FF _h : Target Velocity	116
13.1.5 Object 607D _h : Software Position Limit	117
13.1.6 Object 60B1 _h : Velocity Offset	117
13.1.7 Object 60C2 _h : Interpolation Time Period	118
14 Cyclic synchronous Torque Mode	119
14.1 Detailed Object Specifications	119
14.1.1 Object 6040 _h : Controlword	119
14.1.2 Object 6041 _h : Statusword	120
14.1.3 Object 6062 _h : Position Demand Value	121
14.1.4 Object 6063 _h : Position Actual Internal Value	122
14.1.5 Object 6064 _h : Position Actual Value	122
14.1.6 Object 6071 _h : Target Torque	123
14.1.7 Object 6077 _h : Torque actual Value	123
14.1.8 Object 607D _h : Software Position Limit	123
14.1.9 Object 60B2 _h : Torque Offset	124
14.1.10 Object 60C2 _h : Interpolation Time Period	124
15 Emergency Messages (EMCY)	126
15.1 Format of EMCY Messages	126
15.2 Error Codes used by the TMC1690 Module	126
16 SDO Abort Codes	128
17 Firmware Update	130
18 Figures Index	131
19 Tables Index	132
20 Supplemental Directives	133
20.1 Producer Information	133
20.2 Copyright	133
20.3 Trademark Designations and Symbols	133
20.4 Target User	133

20.5 Disclaimer: Life Support Systems133

20.6 Disclaimer: Intended Use133

20.7 Collateral Documents & Tools134

21 Revision History135

21.1 Firmware Revision135

21.2 Document Revision135

1 Preface

This document specifies objects and modes of operation of the Trinamic TMCM-1690 BLDC/PMSM 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-1690 TMCL firmware again through the CAN interface or the UART(RS232/RS485-ready) 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 server

SDO Communication

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

PDO Communication

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

Further Characteristics

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

1.2 Abbreviations used in this Manual

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

Table 1: Abbreviations used in this Manual

1.3 Firmware Update

The software running on the microprocessor consists of two parts, a boot loader and the CANopen firmware itself. Whereas the boot loader is installed during production and testing at TRINAMIC and remains untouched throughout the whole lifetime, the CANopen firmware can easily be updated by the user. The new firmware can be loaded into the module via the firmware update function of the TMCL-IDE, using the CAN interface or the RS232 interface of the module.

2 Motor Regulation

2.1 Structure of Motor Control Regulation Modes

The TMC1690 supports a current, velocity, and position PI regulation mode for motor control in different application areas. These regulation modes are shown in Figure 1. Individual modes are explained in the following subsections.

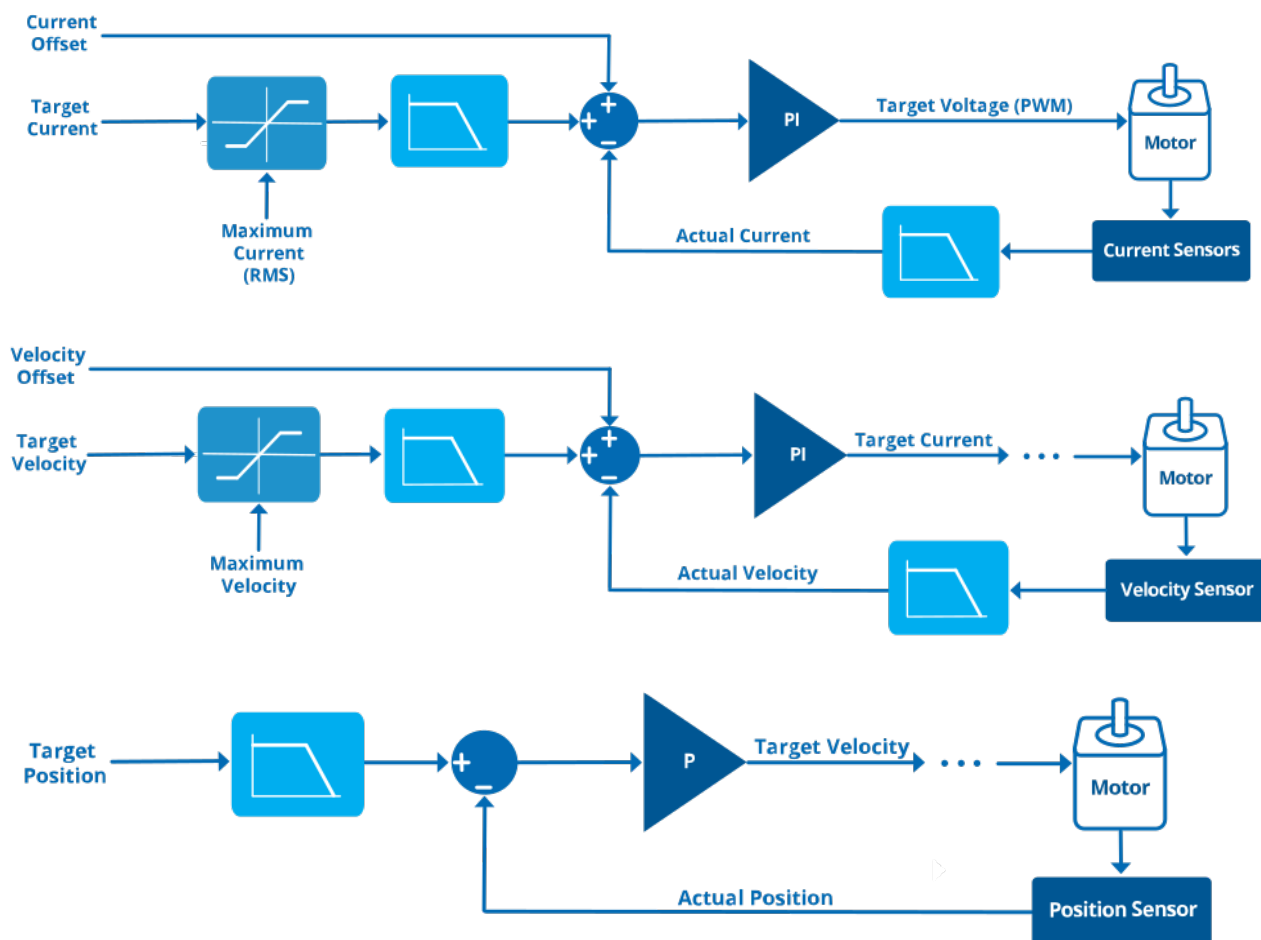


Figure 1: Motion Regulation

It can be seen that the target current signal is first limited by maximum current. There is an option for having a filter (average or biquad) to this signal. This is followed by having an impact of P and I parameters to the signal and is fed to the motor. There is a feedback coming from the actual current forming a closed loop. In the same fashion the velocity and position regulation is performed.

2.2 Current Regulation

The current regulation mode uses a FOC current and flux regulator to adjust a desired motor current. The current loop is running typically at 20KHz and the PWM loop is variable and can be set using object 2003 sub-object **PWM frequency**. The PWM frequency ranges from 20KHz to 120KHz. The target current can be set by object 0x6071 (**Target torque**). The maximal target current is limited by object 0x2003 (**Drive**

settings) sub-object **Maximum current**. The current regulation uses two basic parameters: The P and I parameters

2.2.1 Structure of the Current Regulator

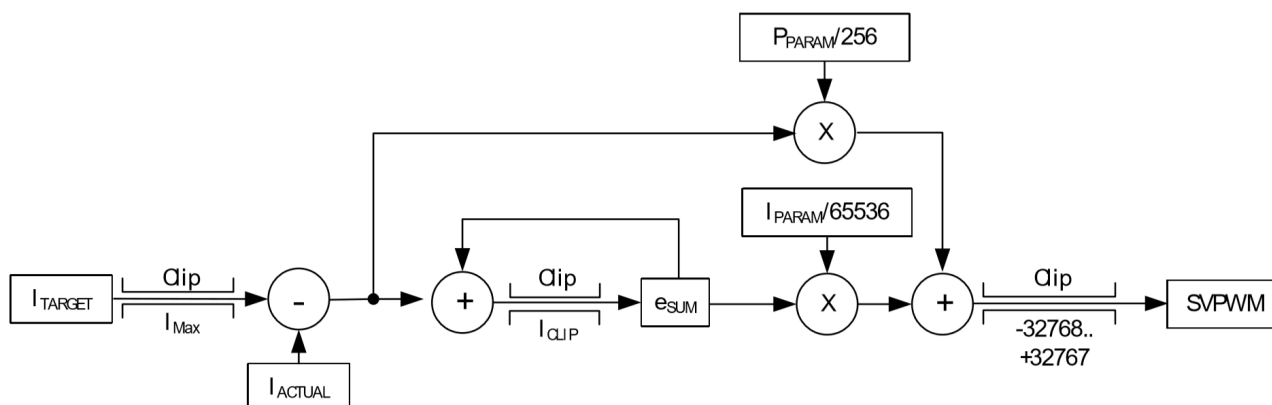


Figure 2: Current Regulation (See parameter descriptions in Table 2)

Current Regulation Parameters	
Parameter	Description
I_{ACTUAL}	Actual current (object 0x6077)
I_{TARGET}	Target current (object 0x6071)
I_{Max}	Max. target current (object 0x2003, sub-object Maximum current)
e_{SUM}	Error sum for integral calculation
P_{PARAM}	Current P parameter (Object 0x201E, sub-object P)
I_{PARAM}	Current I parameter (Object 0x201E, sub-object I)

Table 2: Current Regulation Parameters

2.2.1.1 Parametrizing the Current Regulator Set

Parametrizing the current regulator set is preferably done with PI tuning tool using auto tuning. Further details can be seen in the TMCL-IDE documentation under the section PI tuning. However, it can be done manually as well without the tool. In order to parameterize properly the current regulator set, do as follows:

1. Set the P parameter and the I parameter to zero.
2. Start the motor by using a low target flux (e.g. 1000mA).
3. Modify the current P parameter. Start from a low value and go to a higher value, until the actual current nearly reaches 70% of the desired target current.
4. Do the same with the current I parameter.

For all tests, set the motor current limitation to a realistic value, so that your power supply does not become overloaded during acceleration phases. If your power supply reaches current limitation, the unit may reset or undetermined regulation results may occur.

2.3 Velocity Regulation

Based on the current regulation the motor velocity can be controlled by the velocity PI regulator. The velocity regulation runs in the velocity loop clocked at 2KHz

2.3.1 Structure of the Velocity Regulator

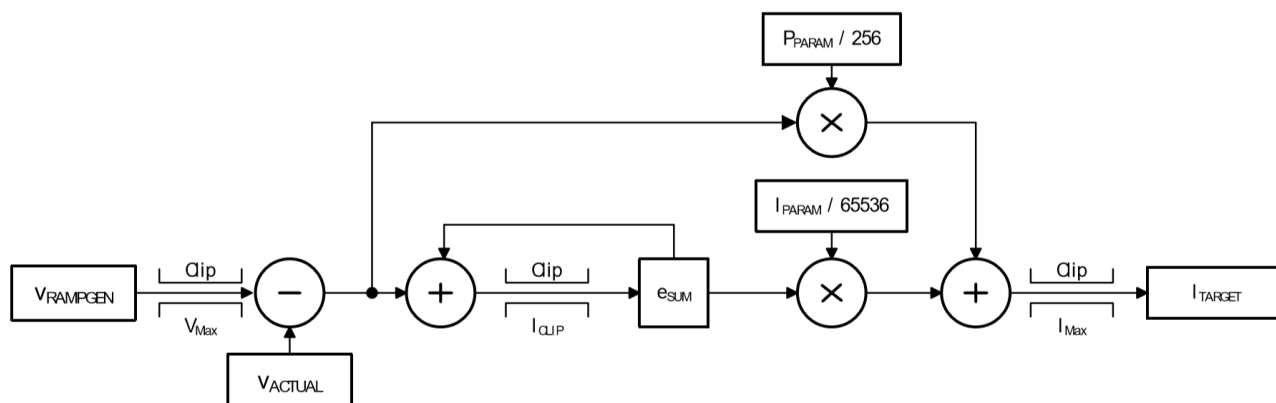


Figure 3: Velocity Regulation (See parameter descriptions in Table 3)

Velocity Regulation Parameters	
Parameter	Description
V_{ACTUAL}	Actual motor velocity (Object 0x606C)
$V_{RAMPGEN}$	Target velocity of ramp generator (Object 0x60FF)
V_{Max}	Max. target velocity (Object 0x607F)
e_{SUM}	Error sum for integral calculation
P_{PARAM}	Velocity P parameter (Object 0x2023, sub-object P)
I_{PARAM}	Velocity I parameter (Object 0x2023, sub-object I)
I_{Max}	Max. target current (object 0x2003, sub-object Maximum current)

Table 3: Velocity Regulation Parameters

2.3.1.1 Parametrizing the Velocity Regulator Set

Parametrizing the velocity regulator set is preferably done with PI tuning tool using auto tuning. Further details can be seen in the TMCL-IDE documentation under the section PI tuning. However, it can be done manually as well without the tool. In order to parameterize properly the velocity regulator set, do as follows:

1. Set the velocity I parameter to zero.
2. Start the motor by using a medium target velocity (e.g. 2000 rpm).
3. Modify the current P parameter.
 - (a) Start from a low value and go to a higher value, until the actual motor speed reaches 80 or 90% of the target velocity.

- (b) The lasting 10 or 20% speed difference can be reduced by slowly increasing the velocity I parameter.

2.4 Velocity Ramp Generator

For a controlled startup of the motor's velocity, a velocity ramp generator can be activated/deactivated by object 0x2037, sub-object **Enable**. The ramp generator uses the maximal allowed motor velocity (Object 0x607F), the acceleration (Object 0x6083) and the desired target velocity (Object 0x60FF) to calculate a ramp generator velocity for the following velocity PI regulator.

2.5 Position Regulation

Based on current and velocity regulators, the TMC-1690 supports a positioning mode based on encoder (abn or abs) or hall sensor position. During positioning the velocity ramp generator can be activated to enable motor positioning with controlled acceleration or it can be disabled to support motor positioning with max allowed speed.

The position regulation uses only one basic parameter: the P regulation parameter

2.5.1 Structure of the Position Regulator

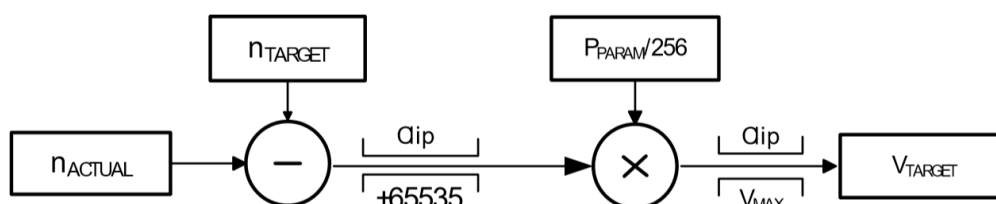


Figure 4: Positioning Regulation (See parameter descriptions in Table 4)

Position Regulation Parameters	
Parameter	Description
n_{ACTUAL}	Actual motor position (Object 0x6064)
n_{TARGET}	Target motor position (Object 0x607A)
P_{PARAM}	Position P parameter (Object 0x2028, sub-object P)
V_{MAX}	Max. allowed velocity (Object 0x607F)
V_{TARGET}	New target velocity for the ramp generator

Table 4: Position Regulation Parameters

2.5.1.1 Parametrizing the Position Regulation

Based on the velocity regulator, only the position regulator P has to be parameterized.

Parametrizing the position regulator is preferably done with PI tuning tool using auto tuning. Further details can be seen in the TMCL-IDE documentation under the section PI tuning. However, it can be done manually as well without the tool. In order to parameterize the position regulator, do as follows:

1. Disable the velocity ramp generator and set position P parameter to zero.
2. Choose a target position and increase the position P parameter until the motor reaches the target position approximately.
3. Switch on the velocity ramp generator. Based on the max. positioning velocity (Object 0x607F) and the acceleration value (Object 0x6083) the ramp generator automatically calculates the slow down point, i.e. the point at which the velocity has to be reduced in order to stop at the desired target position.
4. Reaching the target position is signaled by setting the position end flag.

In order to minimize the time until this flag becomes set, the positioning tolerance MVP target reached distance can be chosen with object 0x2028 sub-object **Position reached distance**.

Since the motor typically is assumed not to signal target reached when the target was just passed in a short moment at a high velocity, additionally the maximum target reached velocity (MVP target reached velocity) can be defined by object 0x6082.

A value of zero for object 0x6082 is the most universal, since it implies that the motor stands still at the target. But when a fast rising of the position end flag is desired, a higher value for the MVP target reached velocity parameter will save a lot of time. The best value should be tried out in the actual application.

2.5.2 Correlation of Target Position and the Position End Flag

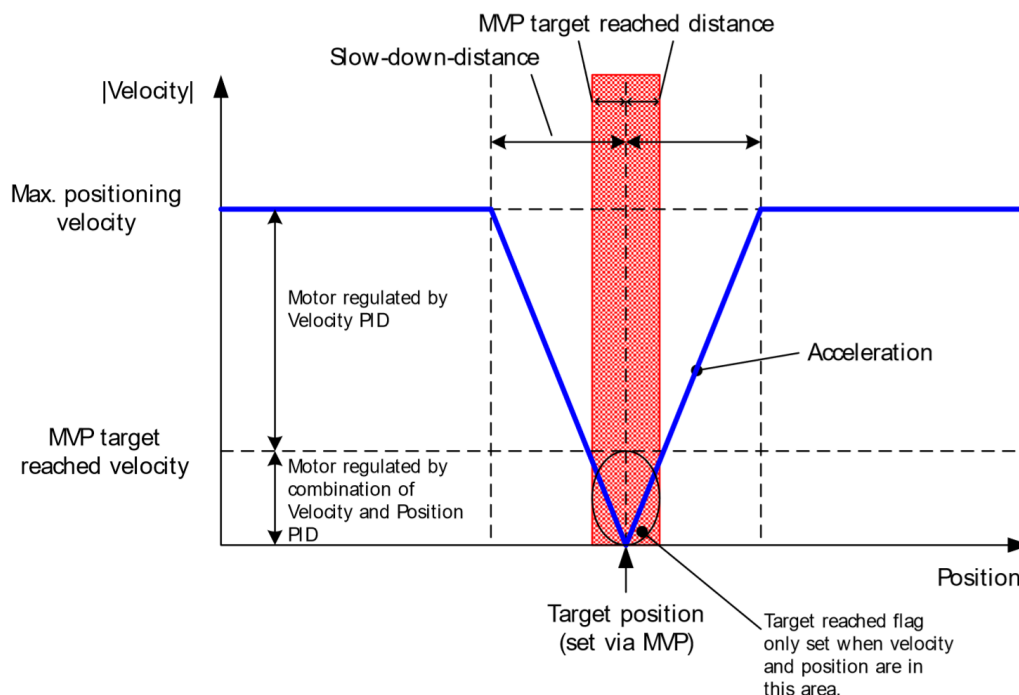


Figure 5: Positioning Algorithm

Depending on motor and mechanics, a low oscillation is normal. This can be reduced to at least ± 1 steps. Without oscillation the regulation cannot keep the position!

3 Ramps

TMC1690 supports two forms of ramps for motion control which are:

- Linear ramp
- Sine shaped ramp

3.1 Linear Ramp

A linear ramp or a trapezoidal ramp predicts the acceleration rates by using a constant gradient. This constant gradient results in constant increase in the velocity. The acceleration and velocity are limited by the object 0x6083 (**Profile acceleration**) and 0x607F (**Max profile velocity**) respectively. This ramp can be used to perform positioning tasks as well. There would be some resonances in the system but for many systems however they can be ignored. The linear ramp is selected by setting 0 to object 0x2003 sub-object **Ramp type**

3.2 Sine shaped ramp

In contrast to linear shaped ramps, the sine shape ramps follow a constant increase in the acceleration as it is one order higher. This would result in a smoother response with far less resonances in the system. Sine shaped ramp is selected setting by 1 to object 0x2003 sub-object **Ramp type**. Figure 6 below gives an impression of how s shaped ramp looks like and how it is different than linear ramp. In the figure, desired position P_t is reached with the maximum acceleration and maximum velocity.

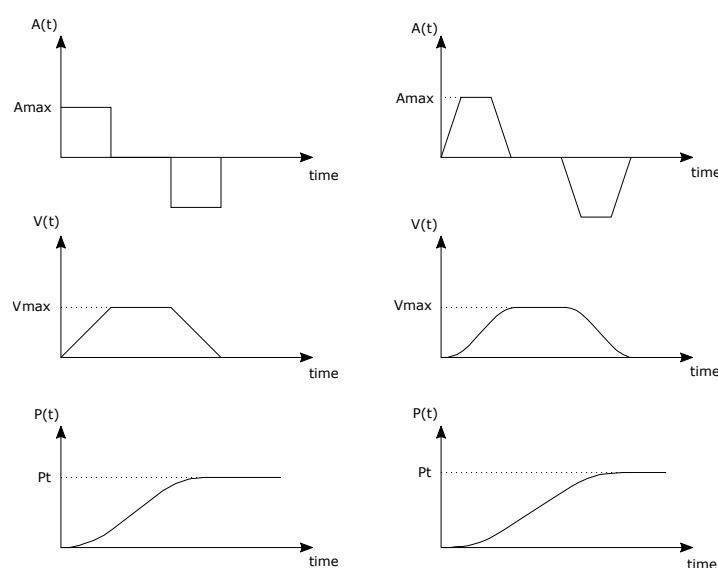


Figure 6: Linear shaped ramp (left) S Shaped ramp (Right)

It can be seen that s shaped ramps provide smoother operation as the acceleration is changed linearly compared to abrupt change of acceleration as in the linear ramp. This results from controlling a finite amount of jerk at the time when the ramp is accelerating or decelerating

After the ramp type is selected, the motor can be turned in both velocity and position mode. For turning the motor in velocity mode, the target velocity object 0x60FF is set to the desired velocity. The ramp will

try to reach its desired value with fastest acceleration limited by the object 0x6083. For turning the motor with position mode, the target position is set with the object 0x607A. The ramp will be calculated that it would try to reach its desired position with the fastest acceleration (which is also limited by object 0x6083) and fastest velocity (limited by maximum velocity object 0x6081).

TMCM-1690 supports all three modes (Torque mode, Velocity mode and Position mode) with both linear and sine shaped ramps. The parameters such as maximum velocity, maximum acceleration, target velocity and target position can be changed on the fly and the ramp would be adjusted accordingly.

4 Module specific functions

4.1 Filters

This module supports software filters for 5 signals, as shown in Figure 7. Each signal can be filtered using either a biquad filter or an averaging filter. Of course there is the option to not filter at all. The averaging filter averages over the last number of samples, in exponents of 2, up to 2^8 . The biquad filter, on the other hand, can be configured as a lowpass filter of the 1st and the 2nd orders, and also as resonance-antiresonance filter. As the equation determining the biquad filter is the same, only a different set of coefficients is required to alter its behaviour. Refer to section 4.1.1 for more details.

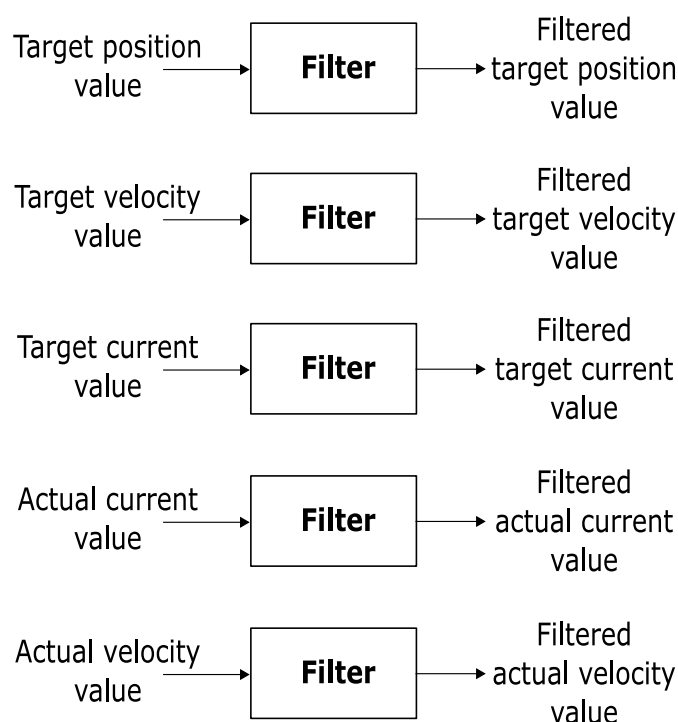


Figure 7: Filter blocks

The filter for the target position value is intended to be used for smoothening position input to the control structure. It is evaluated at every 10th PWM cycle. The filter for the target velocity value is intended to be used for smoothening velocity input from an external master or the position controller. It is also evaluated at every 10th PWM cycle. The filter for the target torque value is intended to be used for smoothening torque input from an external master or the velocity controller. It is also evaluated at every 10th PWM cycle. The filter for the actual current value is intended to be used for smoothening the measured actual current value. It is evaluated at every PWM cycle. The filter for the actual velocity value is intended to be used for smoothening the measured actual velocity value. It is evaluated at every 10th PWM cycle. It can be used as a low-pass filter for bandwidth limitation and noise suppression. Moreover, it can be designed to suppress a resonance or anti-resonance. Same statements are correct for all the filters.

There are a total of seven sub-objects for each signal. To set the type of filter for Target Torque, for example, sub-object **Filter type** in object 0x2041 (**Filter target torque**) can be set to values 0, 1 or 2. Where 0 disables the filter, 1 enables the averaging filter and 2 enables the biquad filter. Sub-object **Average filter size** of object 0x2041 sets the sample size for the averaging filter. Sub-objects 3 through

7 of object 0x2041 correspond to the biquad filter coefficients in Q3.29 format. A similar layout of filter settings is followed by all five signals. Refer to the object table for the filter under consideration

4.1.1 Biquad filter (For future use)

The biquad filters implemented inside of TMCM-1690 use standard biquad filters (standard IIR filter of second order, [Wikipedia Article](#)) in the following structure.

$$Y(n) = X(n) \cdot b_0 + X(n-1) \cdot b_1 + X(n-2) \cdot b_2 + Y(n-1) \cdot a_1 + Y(n-2) \cdot a_2 \quad (1)$$

In this equation $X(n)$ is the actual input sample, while $Y(n-1)$ is the filter output of the last cycle. All coefficients are S32 values and are normalized to a Q3.29 format. Users must take care of correct parametrization of the filter. There is no built-in plausibility or stability check. Biquad state variables are reset when parameters are changed. The TRINAMIC IDE supports parametrization with the Control Settings tool. A standard biquad filter has the following transfer function in the Laplace-Domain:

$$G(s) = \frac{b_2_cont \cdot s^2 + b_1_cont \cdot s + b_0_cont}{a_2_cont \cdot s^2 + a_1_cont \cdot s + a_0_cont} \quad (2)$$

The transfer function needs to be transformed to time discrete domain by Z-Transformation and coefficients need to be normalized. This is done by the following equations.

$$b_2_z = (b_0_cont \cdot T^2 + 2 \cdot b_1_cont \cdot T + 4 \cdot b_2_cont) / (T^2 - 2 \cdot a_1_cont \cdot T + 4 \cdot a_2_cont) \quad (3)$$

$$b_1_z = (2 \cdot b_0_cont \cdot T^2 - 8 \cdot b_2_cont) / (T^2 - 2 \cdot a_1_cont \cdot T + 4 \cdot a_2_cont) \quad (4)$$

$$b_0_z = (b_0_cont \cdot T^2 - 2 \cdot b_1_cont \cdot T + 4 \cdot b_2_cont) / (T^2 - 2 \cdot a_1_cont \cdot T + 4 \cdot a_2_cont) \quad (5)$$

$$a_2_z = (T^2 + 2 \cdot a_1_cont \cdot T + 4 \cdot a_2_cont) / (T^2 - 2 \cdot a_1_cont \cdot T + 4 \cdot a_2_cont) \quad (6)$$

$$a_1_z = (2 \cdot T^2 - 8 \cdot a_2_cont) / (T^2 - 2 \cdot a_1_cont \cdot T + 4 \cdot a_2_cont) \quad (7)$$

$$b_0 = \text{round}(b_0_z \cdot 2^{29}) \quad (8)$$

$$b_1 = \text{round}(b_1_z \cdot 2^{29}) \quad (9)$$

$$b_2 = \text{round}(b_2_z \cdot 2^{29}) \quad (10)$$

$$a_1 = \text{round}(-a_1_z \cdot 2^{29}) \quad (11)$$

$$a_2 = \text{round}(-a_2_z \cdot 2^{29}) \quad (12)$$

while T is the sampling time according to $\text{PWM_MAX_COUNT} \cdot 10 \text{ ns}$ and variables with index z are auxiliary variables.

A standard second order lowpass filter with given cutoff frequency ω_c and damping factor D has the following transfer function in the Laplace-Domain:

$$G_{LP}(s) = \frac{1}{\frac{1}{\omega_c^2} \cdot s^2 + \frac{2D}{\omega_c} \cdot s + 1} \quad (13)$$

Users can determine filter coefficients with the equations above by comparing coefficients of both transfer functions.

4.2 Mechanical Brake

The TMCM-1690 also supports a mechanical brake that can have a separate power source. As the brake is operated by mosfet on the baseboard, there is an upper limit to the current it can handle. Thus, the user has to provide the brake's electrical resistance and supply voltage to set the upper limit of PWM duty cycle.

The object used for Mechanical brake for TMCM-1690 is 0x205A (**Mechanical brake**). For setup, first provide the supply voltage with sub-object **Supply voltage** and brake's resistance with sub-object **Resistance**. Then provide the duty cycles that the brake would have on release with sub-object **Releasing duty cycle** and as hold with sub-object **Holding duty cycle**, while also providing release duration with sub-object **Releasing duration**. Setting sub-object **Enable brake output** enables the brake to be used. Now the brake can be released by setting sub-object **Release brake**. This causes the brake line to output a PWM signal with release duty cycle for the release duration. After the release duration, the PWM signal goes to holding duty cycle. See Figure 8.

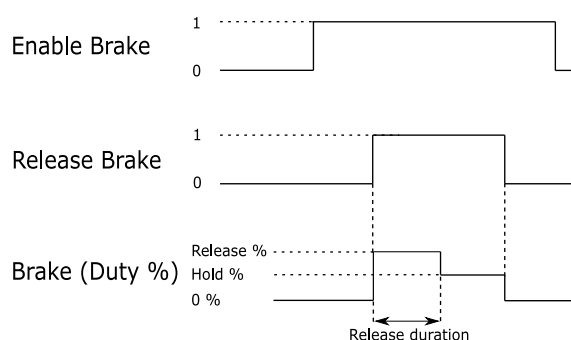


Figure 8: Mechanical brake settings

4.3 Brake chopper

A servo system feeds back energy to the power supply line during deceleration and load control. The energy can lead to a voltage rise on the power supply system if it is not dissipated. The voltage overshoot of a system without brake chopper depends on the motor deceleration time, kinetic energy and the servo module buffer capacity. The brake chopper dissipates this energy from the system, and thus avoids system damage. TMCM-1690 supports two kinds of brake chopper

- PWM braking
- Resistive/Shunt braking

4.3.1 PWM braking

In PWM braking, the motor coils are used to perform the dissipation of the energy from the system. This is done by applying 50 % duty cycle on all three phases of the motor. The object used for brake chopper in TMCM-1690 is 0x205F. For enabling PWM braking, sub-object **Type** is set to 0. Additionally the voltage limits (sub-object **Voltage limit**) is selected such that the supply voltage is larger than it. After that the brake is enabled by setting sub-object **Enable**. This will result in the extra regenerative energy produced due to slowing down dissipated through the coils of the motor.

4.3.2 Resistive/Shunt braking

TMCM-1690 provides a continuous motor voltage monitoring as well as brake chopper output. The brake resistor is connected between the supply voltage and the brake output. For setup, first provide the supply voltage sub-object **Supply voltage** and brake's resistance (sub-object **Resistance**). Shunt braking is selected by setting sub-object **Type** to 1. The voltage limits and hysteresis are selected using the sub-objects

Voltage limit and **Type** respectively. Lastly the brake is enabled by setting sub-object **Enable**. For a full speed ramp stop, the brake resistor should be able to dissipate the complete kinetic energy which is fed back during deceleration phase. The following figure shows an example of brake chopper schematic

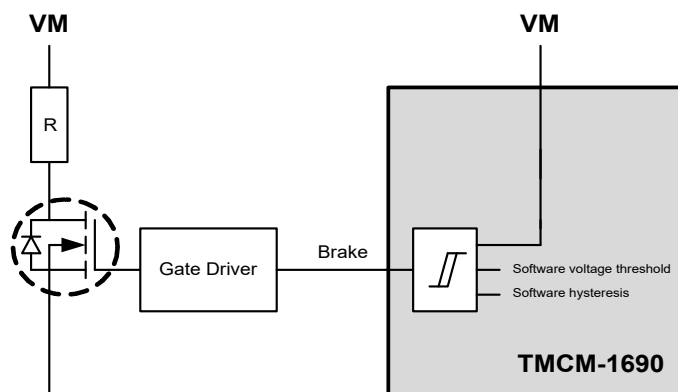


Figure 9: Brake chopper example schematic

4.4 IIT

The IIT monitor provides a check on the energy consumption. The actual current is being monitored, and these values are being squared and summed up periodically over the configured winding time using a 1ms cycle. If one of the limits gets exceeded during this time, the motor will be stopped and the IIT error flag will be set. The object corresponding to IIT settings is 0x202D (**IIT**). The IIT error flag can be reset by writing any value to sub-object **Clear exceed flags**. There are two IIT windows (see figure 10). The first one directly uses the actual current, and the second one uses the actual current divided by $\sqrt{2}$ (less power over longer time). Sub-objects **Limit 1** and **Limit 2** are limits for the two windows. Sub-objects **Sum 1** and **Sum 2** show the actual integration sums.

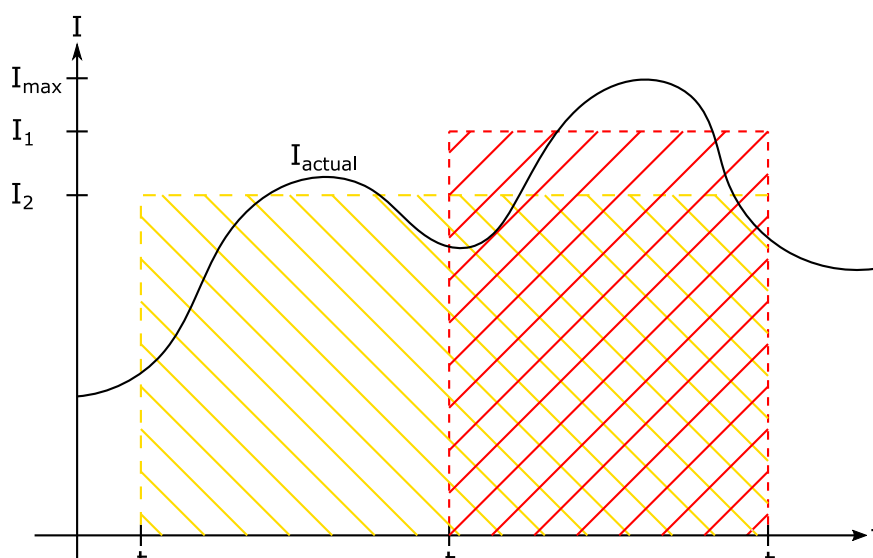


Figure 10: IIT Monitor Windows

4.5 Lower Velocity PI

The TMCM-1690 supports having different PI parameters for lower velocities. This provides a better control and smoother operation. It is possible to have a stronger set of PI parameters for the lower velocities and a relatively weaker set for the higher velocities. It is good to have the set of both PI parameters and the velocity where the user wants to perform the switch (to get the parameters, PI tuning tool can be used). The object used for lower velocity PI parameters is 0x206B (**Lower velocity PI**) The velocity at which the switch is performed can be set using the sub-object **Switch over speed** (default value is 0). After that, P and I parameters can be set using the sub-objects **Lower velocity P** and **Lower velocity I** respectively. These will be the PI parameters for the velocities less than the switch over velocity. Parameters for higher velocities are set by writing on the standard PI velocity parameters

5 Communication

5.1 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 5: 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 6: Service Types

5.2 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 11.

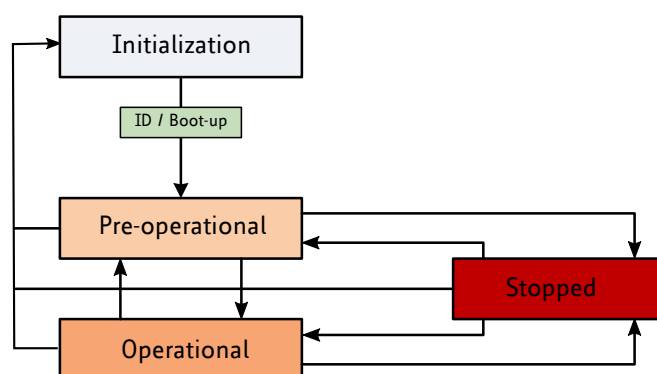


Figure 11: NMT State Machine

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.

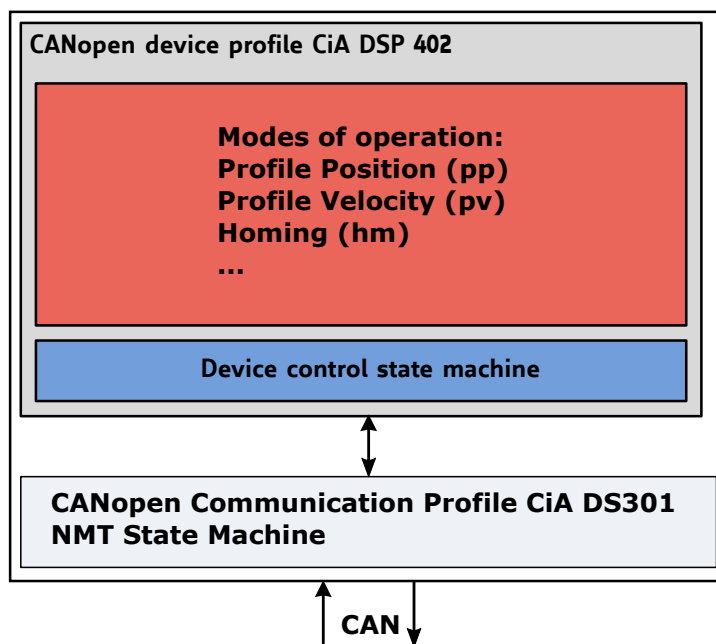


Figure 12: Communication Architecture

5.3 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.

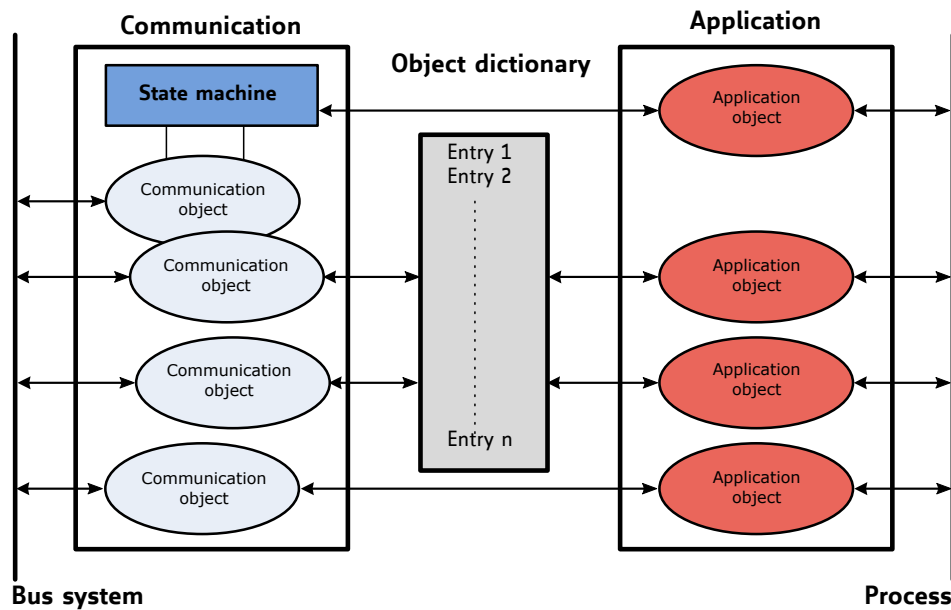


Figure 13: Device Model

5.4 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 7:

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

Table 7: Object Dictionary

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

The manufacturer segment at indices 2000_h through 5FFF_h contains manufacturer specific objects. These objects control the special features of the Trinamic TMC-1690 motion control device.

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

6 Communication Area

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

6.1 Detailed Object Specifications

6.1.1 Object 1000_h: Device Type

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

Object Description			
Index	Name	Object Type	Data Type
1000 _h	Device type	Variable	UNSIGNED32

Table 8: Object Description (1000_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	no	UNSIGNED32	FFFC0192 _h

Table 9: Entry Description (1000_h)

6.1.2 Object 1001_h: Error Register

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

Object Description			
Index	Name	Object Type	Data Type
1001 _h	Error register	Variable	UNSIGNED8

Table 10: Object Description (1001_h)

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

Table 11: Entry Description (1001_h)

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

Table 12: Error Register Bits

6.1.3 Object 1005_h: COB-ID SYNC Message

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

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

Table 13: Value Definition (1005_h)

Object Description			
Index	Name	Object Type	Data Type
1005 _h	COB-ID SYNC message	Variable	UNSIGNED32

Table 14: Object Description (1005_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	UNSIGNED32	80 _h

Table 15: Entry Description (1005_h)

6.1.4 Object 1008_h: Manufacturer Device Name

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

Object Description			
Index	Name	Object Type	Data Type
1008 _h	Manufacturer Device Name	Variable	Visible String

Table 16: Object Description (1008_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	no	—	TMC-1690

Table 17: Entry Description (1008_h)

6.1.5 Object 1009_h: Manufacturer Hardware Version

This object contains the hardware version description.

Object Description			
Index	Name	Object Type	Data Type
1009 _h	Manufacturer Hardware Version	Variable	Visible String

Table 18: Object Description (1009_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	no	—	Depends on device, e.g. 1.00

Table 19: Entry Description (1009_h)

6.1.6 Object 100A_h: Manufacturer Software Version

This object contains the software version description.

Object Description			
Index	Name	Object Type	Data Type
100A _h	Manufacturer Software Version	Variable	Visible String

Table 20: Object Description (100A_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	no	—	Depends on device, e.g. 1.00.

Table 21: Entry Description (100A_h)

6.1.7 Object 100C_h: Guard Time

The objects at index 100C_h and 100D_h shall indicate the configured guard time respectively the life time factor. The life time factor multiplied with the guard time gives the life time for the life guarding protocol.

Object Description			
Index	Name	Object Type	Data Type
100C _h	Guard Time	Variable	UNSIGNED16

Table 22: Object Description (100C_h)

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

Table 23: Entry Description (100C_h)

6.1.8 Object 100D_h: Life Time Factor

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

Object Description			
Index	Name	Object Type	Data Type
100D _h	Life Time Factor	Variable	UNSIGNED8

Table 24: Object Description (100D_h)

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

Table 25: Entry Description (100D_h)

6.1.9 Object 1010_h: Store Parameters

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

The TMC1690 module supports saving of the following parameter groups:

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

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

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

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

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

Object Description			
Index	Name	Object Type	Data Type
1010 _h	Store Parameters	Array	UNSIGNED32

Table 27: Object Description (1010_h)

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

Table 28: Entry Description (1010_h)

6.1.10 Object 1011_h: Restore Parameters

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

The TMC1690 module supports restoring of the following parameter groups:

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

Note

In order to avoid restoring the parameters by mistake, restoring is only executed when a specific signature is written to the appropriate sub-Index. This signature is "load" (64616F6C_h, see also table 29).

Load Signature			
d	a	o	l
64 _h	61 _h	6F _h	6C _h

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

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

After the default values have been restored they will become active after the next rest or power cycle of the TMCM-1690.

Object Description			
Index	Name	Object Type	Data Type
1011 _h	Restore parameters	Array	UNSIGNED32

Table 30: Object Description (1011_h)

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

Table 31: Entry Description (1011_h)

6.1.11 Object 1014_h: COB-ID Emergency Object

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

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

Table 32: Object Description (1014_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	UNSIGNED32	80 _h + Node ID

Table 33: Entry Description (1014_h)

6.1.12 Object 1015_h: Inhibit Time EMCY

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

Object Description			
Index	Name	Object Type	Data Type
1015 _h	COB-ID emergency object	Variable	UNSIGNED16

Table 34: Object Description (1015_h)

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

Table 35: Entry Description (1015_h)

6.1.13 Object 1016_h: Consumer Heartbeat Time

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

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

Table 36: Value Definition (1016_h)

Object Description			
Index	Name	Object Type	Data Type
1016 _h	Consumer heartbeat time	Array	UNSIGNED32

Table 37: Object Description (1016_h)

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_h)

6.1.14 Object 1017_h: Producer Heartbeat Time

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

Object Description			
Index	Name	Object Type	Data Type
1017 _h	Producer heartbeat time	Variable	UNSIGNED16

Table 39: Object Description (1017_h)

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

Table 40: Entry Description (1017_h)

6.1.15 Object 1018_h: Identity Object

The object 1018_h contains general information about the device:

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

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

Table 41: Object Description (1018_h)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
0	Number of entries	ro	no	0...3	3
1	Vendor ID	ro	no	UNSIGNED32	0286 _h
2	Product code	ro	no	UNSIGNED32	1690
3	Revision number	ro	no	UNSIGNED32	e.g. 20003 _h for version 2.3

Table 42: Entry Description (1018_h)

6.1.16 Object 1029_h: Error Behaviour

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

- Communication error
- Application error

Object Description			
Index	Name	Object Type	Data Type
1029 _h	Error behaviour	Array	UNSIGNED8

Table 43: Object Description (1029_h)

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

Table 44: Entry Description (1029_h)

6.1.17 Objects 1400_h – 1403_h: Receive PDO Communication Parameter

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

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

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

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

Table 45: Object Description (1400_h)

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

Table 46: Entry Description (1400_h)

6.1.18 Objects 1600_h – 1603_h: Receive PDO Mapping Parameter

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

Object Description			
Index	Name	Object Type	Data Type
1600 _h – 1603 _h	Receive PDO mapping parameter	RECORD	PDO Mapping
1600 _h	RPDO 1	RECORD	PDO Mapping
1601 _h	RPDO 2	RECORD	PDO Mapping
1602 _h	RPDO 3	RECORD	PDO Mapping
1603 _h	RPDO 4	RECORD	PDO Mapping

Table 47: Object Description (1600_h)

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

Table 48: Entry Description (1600_h)

Before making changes to PDO definitions, first mark the PDO as inactive by setting bit 31 of its COB-ID (see section 6.1.17). Then, set its number of mapped PDO entries to zero (sub-index 0 of the appropriate PDO mapping object). Now, the mappings themselves can be changed. After that, set the number of mapped objects to the desired value, and finally activate the PDO by clearing bit 31 of its COB-ID.

6.1.19 Objects 1800_h – 1803_h: Transmit PDO Communication Parameter

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

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

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

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

Sub-index 4 is not used.

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

Object Description			
Index	Name	Object Type	Data Type
1800 _h – 1803 _h	Transmit PDO communication parameter	RECORD	TPDO CommPar
1800 _h	TPDO 1	RECORD	TPDO CommPar
1801 _h	TPDO 2	RECORD	TPDO CommPar
1802 _h	TPDO 3	RECORD	TPDO CommPar
1803 _h	TPDO 4	RECORD	TPDO CommPar

Table 49: Object Description (1800_h)

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

Table 50: Entry Description (1800_h)

6.1.20 Objects 1A00_h – 1A03_h: Transmit PDO Mapping Parameter

These objects contain the mapping parameters for the TPDOs the device is able to transmit. The sub-index 0 contains the number of valid entries within the mapping record. This number of entries is also the number of the application variables which shall be transmitted with the corresponding TPDO. The sub-indices from 1 to the number of entries contain the information about the mapped application variables. These entries describe the PDO contents by their index, sub-index and length.

Object Description			
Index	Name	Object Type	Data Type
1A00 _h – 1A03 _h	Transmit PDO mapping parameter	RECORD	PDO Mapping
1A00 _h	TPDO 1	RECORD	PDO Mapping
1A01 _h	TPDO 2	RECORD	PDO Mapping
1A02 _h	TPDO 3	RECORD	PDO Mapping
1A03 _h	TPDO 4	RECORD	PDO Mapping

Table 51: Object Description (1A00_h)

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

Table 52: Entry Description (1A00_h)

Before making changes to PDO definitions, first mark the PDO as inactive by setting bit 31 of its COB-ID (see section 6.1.19). Then, set its number of mapped PDO entries to zero (sub-index 0 of the appropriate PDO mapping object). Now, the mappings themselves can be changed. After that, set the number of mapped objects to the desired value, and finally activate the PDO by clearing bit 31 of its COB-ID.

7 Manufacturer specific Area

The manufacturer segment contains manufacturer specific objects. These objects control the special features of the Trinamic Motion Control device TMC1690.

7.1 Detailed Object Specifications

7.1.1 Object 2000_h: Motor settings

This object provides information about motor settings which can be deduced from the datasheet.

Object Description			
Index	Name	Object Type	Data Type
2000 _h	Motor settings	Variable	Record

Table 53: Object Description (2000_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Motor type	no	0	1	0	—	RW
2	Motor family	no	0	1	0	—	RW
3	Pole pairs	no	0	255	4	—	RW
4	Pole pair distance	no	1	65535	10	μm	RW
5	Nominal current	no	0	10000	3470	mA	RW
6	Peak current	no	0	10000	10000	mA	RW
7	Line to line resistance	no	1	65535	720	mΩ	RW
8	Line to line inductance	no	1	65535	1200	μH	RW
9	Torque constant	no	1	65535	30	mNm/A	RW
10	Inertia	no	1	65535	480	gcm ²	RW

Table 54: Entry Description (2000_h)

Parameter	Description
Motor type	Select a motor type connected to the module 0. 3-phase BLDC 1. DC motor
Motor family	Select your motor family 0. Rotary 1. Linear

Table 55: Parameter description (2000_h)

7.1.2 Object 2003_h: Drive settings

This object is used to configure the drive settings. It has parameters such as ramp type, commutation mode, sensors, maximum current which are used to initialize the drive

Object Description			
Index	Name	Object Type	Data Type
2003 _h	Drive settings	Variable	Record

Table 56: Object Description (2003_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Ramp type	no	0	1	0	—	RW
2	Motor direction	no	0	1	0	—	RW
3	Commutation mode	no	0	5	0	—	RW
4	Peripherals	no	0	63	34	—	RW
5	Current sensor selection	no	0	0	0	—	RO
6	Maximum current	no	0	10000	4000	mA	RW
7	PWM scheme	no	0	1	0	—	RW
8	PWM frequency	no	0	5	0	—	RW
9	PWM dead time	no	0	255	1	ms	RW

Table 57: Entry Description (2003_h)

Parameter	Description
Ramp type	Select a ramp type between linear and s shaped ramp 0. Linear ramp 1. S shaped ramp
Motor direction	Used for reversing motor shaft direction 0. Rotate clockwise 1. Rotate counter clockwise
Commutation mode	Select commutation mode based on the sensor used 0: Disabled 1: Open loop 2: Digital hall(foc) 3: Digital hall (Block) 4: ABN encoder 5: ABS encoder
PWM scheme	Select the PWM scheme 0. Center aligned 1. Down counting
PWM frequency	Select the desired PWM frequency(KHz) 0. 20000 1. 40000 2. 60000 3. 80000 4. 100000 5. 120000

Table 58: Parameter Description (2003_h)

7.1.3 Object 2005_h: ADC settings

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

Object Description			
Index	Name	Object Type	Data Type
2005 _h	ADC settings	Variable	Record

Table 59: Object Description (2005_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	ADC_u_raw	no	0	4095	0	—	RO
2	ADC_v_raw	no	0	4095	0	—	RO
3	ADC_w_raw	no	0	4095	0	—	RO
4	ADC_u_offset	no	0	4095	2047	—	RW
5	ADC_v_offset	no	0	4095	2047	—	RW
6	ADC_w_offset	no	0	4095	2047	—	RW
7	ADC_u	no	-32768	32767	0	—	RO
8	ADC_v	no	-32768	32767	0	—	RO
9	ADC_w	no	-32768	32767	0	—	RO

Table 60: Entry Description (2005_h)

7.1.4 Object 200A_h: Open loop settings

This object deals with the open loop settings such as commutation angle, current velocity and position.

Object Description			
Index	Name	Object Type	Data Type
200A _h	Open loop settings	Variable	Record

Table 61: Object Description (200A_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Commutation angle	no	-32768	32767	0	—	RO
2	Current	no	0	10000	2000	mA	RW
3	Velocity	no	-2147483648	2147483648	0	—	RO
4	Position	no	-2147483648	2147483648	0	—	RW

Table 62: Entry Description (200A_h)

7.1.5 Object 200F_h: Digital hall settings

This object relates to the digital hall sensor settings. It has parameters which are used to initialize the hall sensor and run the motor using hall sensor as commutation mode.

Object Description			
Index	Name	Object Type	Data Type
200F _h	Digital hall settings	Variable	Record

Table 63: Object Description (200F_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Sector offset	no	0	5	0	—	RW
2	Direction	no	0	1	0	—	RW
3	Interpolation	no	0	1	0	—	RW
4	Offset	no	-32768	32767	0	—	RW
5	Inputs	no	0	7	0	—	RO
6	Identify trigger configuration	no	0	2	0	—	RW
7	Commutation angle	no	-32768	32767	0	—	RO
8	Open Loop angle difference	no	-32768	32767	0	—	RO
9	Velocity	no	-2147483648	2147483648	0	—	RO
10	Position	no	-2147483648	2147483648	0	—	RW

Table 64: Entry Description (200F_h)

Parameter	Description
Sector Offset	Select the sector offset for configuring hall sensor (degrees) 0. 0 1. 60 2. 120 3. 180 4. 240 5. 300
Direction	Hall sensor direction 0. Standard 1. Inverted
Interpolation	Hall sensor interpolation 0. off 1. on
Identify trigger configuration	Select the method for identifying trigger configuration of hall sensor 0: Standby 1: Direction estimation 2: Sector offset estimation

Table 65: Parameter Description (200F_h)

7.1.6 Object 2014_h: ABN encoder settings

This object relates to the ABN encoder settings. It has parameters which are used to initialize the ABN encoder and run the motor using ABN Encoder as commutation mode.

Object Description			
Index	Name	Object Type	Data Type
2014 _h	ABN encoder settings	Variable	Record

Table 66: Object Description (2014_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Steps	no	0	65535	4096	—	RW
2	Direction	no	0	1	0	—	RW
3	Offset	no	-32768	32767	0	—	RW
4	Clear on null	no	0	1	0	—	WO
5	Clear once	no	0	1	0	—	WO
6	Init mode	no	0	3	0	—	RW
7	Init state	no	0	3	0	—	RO
8	Init delay	no	0	10000	1000	ms	RW
9	Init velocity	no	0	65535	0	rpm	RW
10	Inputs	no	0	7	0	—	RO
11	Encoder value	no	0	65535	0	—	RO
12	Sensor side	no	0	1	0	—	RW
13	Commutation angle	no	-32768	32767	0	—	RO
14	Open loop angle difference	no	-32768	32767	0	—	RO
15	Velocity	no	-2147483648	2147483648	0	—	RO
16	Position	no	-2147483648	2147483648	0	—	RW

Table 67: Entry Description (2014_h)

Parameter	Description
Direction	ABN encoder direction 0. Standard 1. Inverted
Init mode	Select the method for initialization of ABN encoder 0. Estimate offset 1. Estimate offset (Shake) 2. Use offset 3. Use hall
Sensor side	Select where the encoder is connected 0: Disabled 1: Motor side

Table 68: Parameter Description (2014_h)

7.1.7 Object 2015_h: ABN encoder 2 settings

This object relates to the second ABN encoder settings. It has parameters which are used to initialize second ABN encoder and run the motor with second ABN encoder as a backup encoder

Object Description			
Index	Name	Object Type	Data Type
2015 _h	ABN encoder 2 settings	Variable	Record

Table 69: Object Description (2015_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Steps	no	0	65535	4096	—	RW
2	Direction	no	0	1	0	—	RW
3	Inputs	no	0	7	0	—	RO
4	Sensor side	no	0	2	0	—	RW
5	Commutation angle	no	-32768	32767	0	—	RO
6	Velocity	no	-2147483648	2147483648	0	—	RO
7	Position	no	-2147483648	2147483648	0	—	RW

Table 70: Entry Description (2015_h)

Parameter	Description
Direction	ABN 2 encoder direction 0. Standard 1. Inverted
Sensor side	Select where the encoder is connected 0: Disabled 1: Motor side 2: Load side

Table 71: Parameter Description (2015_h)

7.1.8 Object 2019_h: Absolute encoder settings

This object relates to the absolute encoder settings. It has parameters which are used to initialize the absolute encoder and run the motor using absolute encoder as commutation mode.

Object Description			
Index	Name	Object Type	Data Type
2019 _h	Absolute encoder settings	Variable	Record

Table 72: Object Description (2019_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Type	no	0	2	0	—	RW
2	Init mode	no	0	2	0	—	RW
3	Direction	no	0	1	0	—	RW
4	Offset	no	-32768	32767	0	—	RW
5	Sensor side	no	0	2	0	—	RW
6	Commutation angle	no	-32768	32767	0	—	RO
7	Open loop angle difference	no	-32768	32767	0	—	RO
8	Velocity	no	-2147483648	2147483648	0	—	RO
9	Position	no	-2147483648	2147483648	0	—	RW

Table 73: Entry Description (2019_h)

Parameter	Description
Type	Select the ABS encoder connected 0. Disabled 1. AM4096 2. ADA4573
Direction	ABS encoder direction 0. Standard 1. Inverted
Init mode	Select the method for initialization of ABN encoder 0. Estimate offset 1. Estimate offset (shake) 2. Use offset
Sensor side	Select where the encoder is connected 0: Disabled 1: Motor side 2: Load side

Table 74: Parameter Description (2019_h)

7.1.9 Object 201E_h: Torque mode

Object Description			
Index	Name	Object Type	Data Type
201E _h	Torque mode Settings	Variable	Record

Table 75: Object Description (201E_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Target flux	no	-10000	10000	0	mA	RW
2	Actual flux	no	-2147483648	2147483647	0	mA	RO
3	P	no	0	65535	300	—	RW
4	I	no	0	65535	600	—	RW
5	Torque PI error sum	no	-2147483648	2147483647	0	—	RO
6	Flux PI error sum	no	-2147483648	2147483647	0	—	RO
7	Torque PI error	no	-2147483648	2147483647	0	—	RO
8	Flux PI error	no	-2147483648	2147483647	0	—	RO

Table 76: Entry Description (201E_h)

Parameter	Description
P	P parameter for the torque PI controller.
I	I parameter for the torque PI controller.

Table 77: Parameter Description (201E_h)

7.1.10 Object 2023_h: Velocity mode

Object Description			
Index	Name	Object Type	Data Type
2023 _h	Velocity mode	Variable	Record

Table 78: Object Description (2023_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Sensor selection	no	0	4	0	—	RW
2	Unit selection	no	0	1	0	—	RW
3	Halted velocity	no	0	65535	0	rpm	RW
4	P	no	0	65535	600	—	RW
5	I	no	0	65535	300	—	RW
6	PI error sum	no	-2147483648	2147483647	0	—	RO
7	PI error	no	-2147483648	2147483647	0	—	RO

Table 79: Entry Description (2023_h)

Parameter	Description
P	P parameter for the velocity PI controller.
I	I parameter for the velocity PI controller.
Sensor selection	Select a commutation mode that fits best to your motor's sensors. 0: Same as commutation 1: Hall sensor 2: ABN encoder 3: ABN encoder 2 4: ABS encoder
Unit selection	Select mechanical or electrical velocity unit. 0. Mechanical rpm 1. Electrical rpm
Halted velocity	If the actual velocity is below this value the motor halted flag will be set.

Table 80: Parameter Description (2023_h)

7.1.11 Object 2028_h: Position mode

Object Description			
Index	Name	Object Type	Data Type
2028 _h	Position mode	Variable	Record

Table 81: Object Description (2028_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Sensor selection	no	0	4	0	—	RW
2	Position reached distance	no	0	100000	0	—	RW
3	P	no	0	65535	20	—	RW
4	PI error	no	-2147483648	2147483647	0	—	RO
5	Scalar	no	-2147483648	2147483647	0	—	WO

Table 82: Entry Description (202B_h)

Parameter	Description
P	P parameter for the velocity PI controller.
Sensor selection	Select a commutation mode that fits best to your motor's sensors. 0: Same as commutation 1: Hall sensor 2: ABN encoder 3: ABN encoder 2 4: ABS encoder

Table 83: Parameter Description (202B_h)

7.1.12 Object 202B_h: Gearbox

This object deals with the gearbox and is used to include a gear ratio at the output. This can be rotary to rotary or rotary to linear.

Object Description			
Index	Name	Object Type	Data Type
202B _h	Gearbox	Variable	Record

Table 84: Object Description (202B_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Transmission type	no	0	1	0	—	RW
2	Motor displacement	no	1	2147483647	1	—	RW
3	Load displacement	no	1	2147483647	1	—	RW
4	Invert direction	no	0	1	0	—	RW

Table 85: Entry Description (202B_h)

Parameter	Description
Transmission type	Select the transmission type from rotary to linear or rotary 0: rotary to rotary 1: rotary to linear
Invert direction	Use this sub object to invert the direction of gearbox 0. Off 1. On

Table 86: Parameter Description (202B_h)

7.1.13 Object 202D_h: IIT

Object Description			
Index	Name	Object Type	Data Type
202D _h	IIT	Variable	Record

Table 87: Object Description (202D_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Thermal winding time constant 1	no	1000	60000	30000	ms	RW
2	Limit 1	no	0	54000000	10300000	—	RW
3	Sum 1	no	0	4294967295	0	—	RO
4	Thermal winding time constant 2	no	1000	60000	30000	ms	RW
5	Limit 2	no	0	54000000	10300000	—	RW
6	Sum 2	no	0	4294967295	0	—	RO
7	Clear exceeded flags	no	0	1	0	—	WO

Table 88: Entry Description (202D_h)

7.1.14 Object 2032_h: Windows

This object is used to set the position and velocity windows and to clear the windows in case of an error.

Object Description			
Index	Name	Object Type	Data Type
2032 _h	Windows	Variable	Record

Table 89: Object Description (2032_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Velocity window	no	0	65535	500	—	RW
2	Clear velocity window error	no	0	1	0	—	WO
3	Position window	no	0	2147483647	1638400	—	RW
4	Clear position window error	no	0	1	0	—	WO

Table 90: Entry Description (2032_h)

7.1.15 Object 2037_h: Ramp

Object Description			
Index	Name	Object Type	Data Type
2037 _h	Ramp	Variable	Record

Table 91: Object Description (2037_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Enable	no	0	1	0	—	RW
2	Enable velocity feed forward	no	0	1	0	—	RW
3	Acceleration	no	0	200000	0	rpm/s	RO
4	Velocity	no	-200000	200000	0	rpm	RO
5	Position	no	-2147483648	2147483647	0	—	RO

Table 92: Entry Description (2037_h)

7.1.16 Object 203C_h: Homing

This object deals with the homing routine.

Object Description			
Index	Name	Object Type	Data Type
203C _h	Homing	Variable	Record

Table 93: Object Description (203C_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	State	no	0	255	0	—	RO
2	Position offset CW	no	0	4294967295	40000	—	RW
3	Position offset CCW	no	0	4294967295	40000	—	RW
4	Current threshold	no	0	10000	2000	mA	RW
5	Teach position limit	no	0	3	0	—	RW

Table 94: Entry Description (203C_h)

7.1.17 Object 2041_h: Filter target torque

This object is used to configure filter for target torque.

Object Description			
Index	Name	Object Type	Data Type
2041 _h	Filter target torque	Variable	Record

Table 95: Object Description (2041_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Filter type	no	0	2	1	—	RW
2	Average filter size	no	1	8	1	—	RW
3	Biquad coeff a1	no	-2147483648	2147483647	0	—	RW
4	Biquad coeff a2	no	-2147483648	2147483647	0	—	RW
5	Biquad coeff b0	no	-2147483648	2147483647	0	—	RW
6	Biquad coeff b1	no	-2147483648	2147483647	0	—	RW
7	Biquad coeff b2	no	-2147483648	2147483647	0	—	RW

Table 96: Entry Description (2041_h)

Parameter	Description
Filter type	Select the filter type 0: Disabled 1: Average 2. Biquad

Table 97: Parameter Description (2041_h)

7.1.18 Object 2046_h: Filter actual torque

This object is used to configure filter for actual torque.

Object Description			
Index	Name	Object Type	Data Type
2046 _h	Filter actual torque	Variable	Record

Table 98: Object Description (2046_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Filter type	no	0	2	1	—	RW
2	Average filter size	no	1	8	1	—	RW
3	Biquad coeff a1	no	-2147483648	2147483647	0	—	RW
4	Biquad coeff a2	no	-2147483648	2147483647	0	—	RW
5	Biquad coeff b0	no	-2147483648	2147483647	0	—	RW
6	Biquad coeff b1	no	-2147483648	2147483647	0	—	RW
7	Biquad coeff b2	no	-2147483648	2147483647	0	—	RW

Table 99: Entry Description (2046_h)

Parameter	Description
Filter type	Select the filter type 0: Disabled 1: Average 2. Biquad

Table 100: Parameter Description (2046_h)

7.1.19 Object 204B_h: Filter target velocity

This object is used to configure filter for target velocity.

Object Description			
Index	Name	Object Type	Data Type
204B _h	Filter target velocity	Variable	Record

Table 101: Object Description (204B_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Filter type	no	0	2	1	—	RW
2	Average filter size	no	1	8	1	—	RW
3	Biquad coeff a1	no	-2147483648	2147483647	0	—	RW
4	Biquad coeff a2	no	-2147483648	2147483647	0	—	RW
5	Biquad coeff b0	no	-2147483648	2147483647	0	—	RW
6	Biquad coeff b1	no	-2147483648	2147483647	0	—	RW
7	Biquad coeff b2	no	-2147483648	2147483647	0	—	RW

Table 102: Entry Description (204B_h)

Parameter	Description
Filter type	Select the filter type 0: Disabled 1: Average 2: Biquad

Table 103: Parameter Description (204B_h)

7.1.20 Object 2050_h: Filter actual velocity

This object is used to configure filter for actual velocity.

Object Description			
Index	Name	Object Type	Data Type
2050 _h	Filter actual velocity	Variable	Record

Table 104: Object Description (2050_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Filter type	no	0	2	1	—	RW
2	Average filter size	no	1	8	1	—	RW
3	Biquad coeff a1	no	-2147483648	2147483647	0	—	RW
4	Biquad coeff a2	no	-2147483648	2147483647	0	—	RW
5	Biquad coeff b0	no	-2147483648	2147483647	0	—	RW
6	Biquad coeff b1	no	-2147483648	2147483647	0	—	RW
7	Biquad coeff b2	no	-2147483648	2147483647	0	—	RW

Table 105: Entry Description (2050_h)

Parameter	Description
Filter type	Select the filter type 0: Disabled 1: Average 2: Biquad

Table 106: Parameter Description (2050_h)

7.1.21 Object 2055_h: Filter target position

This object is used to configure filter for target position.

Object Description			
Index	Name	Object Type	Data Type
2055 _h	Filter target position	Variable	Record

Table 107: Object Description (2055_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Filter type	no	0	2	1	—	RW
2	Average filter size	no	1	8	1	—	RW
3	Biquad coeff a1	no	-2147483648	2147483647	0	—	RW
4	Biquad coeff a2	no	-2147483648	2147483647	0	—	RW
5	Biquad coeff b0	no	-2147483648	2147483647	0	—	RW
6	Biquad coeff b1	no	-2147483648	2147483647	0	—	RW
7	Biquad coeff b2	no	-2147483648	2147483647	0	—	RW

Table 108: Entry Description (2055_h)

Parameter	Description
Filter type	Select the filter type 0: Disabled 1: Average 2: Biquad

7.1.22 Object 205A_h: Mechanical brake

This object is used to setup and use mechanical brake.

Object Description			
Index	Name	Object Type	Data Type
205A _h	Mechanical brake	Variable	Record

Table 109: Object Description (205A_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Release brake	no	0	1	0	—	RW
2	Releasing duty cycle	no	0	100	75	%	RW
3	Holding duty cycle	no	0	100	11	%	RW
4	Releasing duration	no	0	65535	1000	ms	RW
5	Enable brake output	no	0	1	0	—	RW
6	Invert brake output	no	0	1	0	—	RW
7	Supply voltage	no	0	65535	240	0.1V	RW
8	Resistance	no	0	65535	440	mΩ	RW

Table 110: Entry Description (205A_h)

Parameter	Description
Release brake	Use this sub object to switch the brake PWM functionality 0. brake PWM deactivated 1. brake PWM activated
Enable brake output	Use this sub object to switch the brake functionality 0. brake deactivated 1. brake activated

Table 111: Parameter Description (205A_h)

7.1.23 Object 205F_h: Brake chopper

This object deals with setting up and using brake chopper

Object Description			
Index	Name	Object Type	Data Type
205F _h	Brake chopper	Variable	Record

Table 112: Object Description (205F_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Enable	no	0	1	0	—	RW
2	Voltage limit	no	50	1000	300	0.1V	RW
3	Hysteresis	no	0	50	5	0.1V	RW
4	Type	no	0	1	0	—	RW
5	Active	no	0	1	0	—	RW
6	Supply voltage	no	0	65535	240	0.1V	RW
7	Resistance	no	0	65535	22000	mΩ	RW

Table 113: Entry Description (205F_h)

Parameter	Description
Type	Select the brake chopper type 0. PWM braking 1. Shunt braking

Table 114: Parameter Description (205F_h)

7.1.24 Object 2064_h: Reference switch

Object Description			
Index	Name	Object Type	Data Type
2064 _h	Reference switch	Variable	Record

Table 115: Object Description (2064_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Enable	no	0	3	0	—	RW
2	Polarity	no	0	3	0	—	RW

Table 116: Entry Description (2064_h)

7.1.25 Object 2069_h: Monitoring

Object Description			
Index	Name	Object Type	Data Type
2069 _h	Monitoring	Variable	Record

Table 117: Object Description (2069_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Status flags	no	0	4294967295	0	—	RO
2	Supply voltage	no	0	1000	0	0.1V	RO
3	Supply voltage threshold	no	0	1000	480	0.1V	RW
4	Driver temperature	no	-20	150	0	° C	RO
5	Driver temperature threshold	no	-20	150	0	° C	RW

Table 118: Entry Description (2069_h)

7.1.26 Object 206B_h: Lower velocity PI

Object Description			
Index	Name	Object Type	Data Type
206B _h	Lower velocity PI	Variable	Record

Table 119: Object Description (206B_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Switch over velocity	no	0	32767	0	rpm	RW
2	Lower velocity P	no	0	32767	600	—	RW
3	Lower velocity I	no	0	32767	300	—	RW

Table 120: Entry Description (206B_h)

7.1.27 Object 206C_h: Linear drive settings

Object Description			
Index	Name	Object Type	Data Type
206C _h	Linear drive settings	Variable	Record

Table 121: Object Description (206C_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Maximum speed	no	0	2147483647	2147483647	μm/s	RW
2	Acceleration	no	0	2147483647	2147483647	μm/s ²	RW
3	Target velocity	no	-2147483647	2147483647	0	μm/s	RW
4	Actual velocity	no	-2147483647	2147483647	0	μm/s	RW
5	Target position	no	-2147483647	2147483647	0	μm	RW
6	Actual position	no	-2147483647	2147483647	0	μm	RW
7	Ramp velocity	no	-2147483647	2147483647	0	μm/s	RW
8	Ramp position	no	-2147483647	2147483647	0	μm	RW

Table 122: Entry Description (206C_h)

7.1.28 Object 206D_h: Driver status

This object deals with the error handling for driver

Object Description			
Index	Name	Object Type	Data Type
206D _h	Driver status	Variable	Record

Table 123: Object Description (206D_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Driver status flag	no	0	65535	0	—	RO
2	Clear driver error flag	no	0	1	0	—	RW

Table 124: Entry Description (206D_h)

7.1.29 Object 206E_h: Controller initialize

Object Description			
Index	Name	Object Type	Data Type
206E _h	Controller initialize	Variable	Record

Table 125: Object Description (206E_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Controller initialize	no	0	1	0	–	RW

*Table 126: Entry Description (206E_h)***7.1.30 Object 207D_h: Switch parameters**

Object Description			
Index	Name	Object Type	Data Type
207D _h	Switch parameters	Variable	Record

Table 127: Object Description (207D_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Switch parameters	no	0	4294967295	0	–	RW

*Table 128: Entry Description (207D_h)***7.1.31 Object 2082_h: Home offset display**

Object Description			
Index	Name	Object Type	Data Type
2082 _h	Home offset display	Variable	INTEGER32

Table 129: Object Description (2082_h)

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

Table 130: Entry Description (2082_h)

7.1.32 Object 2701_h: Device digital IO direction

This object sets the direction for the general purpose IOs (GPIO2 to GPIO7) as either input or output.

Object Description			
Index	Name	Object Type	Data Type
2701 _h	Device digital IO direction	Variable	UNSIGNED32

Table 131: Object Description (2701_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Device digital IO direction	no	0	252	0	–	RW

Table 132: Entry Description (2701_h)

7.1.33 Object 2702_h: Device digital inputs

Bit Definitions	
Bit	Description
0	REF_R
1	REF_L
2	GPIO2
3	GPIO3
4	GPIO4
5	GPIO5
6	GPIO6
7	GPIO7

Table 133: Bit Definitions (2702_h)

Object Description			
Index	Name	Object Type	Data Type
2702 _h	Device digital inputs	Variable	UNSIGNED32

Table 134: Object Description (2702_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Device digital inputs	no	0	255	0	–	R

Table 135: Entry Description (2702_h)

7.1.34 Object 2703_h: Device digital outputs

With this object the digital outputs (general purpose outputs) can be set or reset. Bits 2...7 of sub index 1 switch the outputs of the module. Bits 2...7 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
2	GPO2
3	GPO3
4	GPO4
5	GPO5
6	GPO6
7	GPO7

Table 136: Bit Definitions (2703_h)

Object Description			
Index	Name	Object Type	Data Type
2703 _h	Device digital outputs	Variable	ARRAY

Table 137: Object Description (2703_h)

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 138: Entry Description (2703_h)

7.1.35 Object 2704_h: CAN bit rate

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

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

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

Object Description			
Index	Name	Object Type	Data Type
2704 _h	CAN bit rate	Variable	UNSIGNED16

Table 139: Object Description (2704_h)

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

Table 140: Entry Description (2704_h)

7.1.36 Object 2705_h: Node ID

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

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

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

Object Description			
Index	Name	Object Type	Data Type
2705 _h	Node ID	Variable	UNSIGNED8

Table 141: Object Description (2705_h)

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

Table 142: Entry Description (2705_h)

7.1.37 Object 2707_h: CAN bit rate load

This object shows the selected CAN bit rate.

Object Description			
Index	Name	Object Type	Data Type
2707 _h	CAN bit rate load	Variable	UNSIGNED8

Table 143: Object Description (2707_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	CAN bit rate load	no	20	1000	1000		R

Table 144: Entry Description (2707_h)

7.1.38 Object 2708_h: Node ID load

This object shows the selected node ID.

Object Description			
Index	Name	Object Type	Data Type
2708 _h	Node ID load	Variable	UNSIGNED8

Table 145: Object Description (2708_h)

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

Table 146: Entry Description (2708_h)

7.1.39 Object 270B_h: HAL version

Object Description			
Index	Name	Object Type	Data Type
270B _h	HAL version	Variable	Record

Table 147: Object Description (270B_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	HAL version	no	0	4294967295	0	–	R

Table 148: Entry Description (270B_h)

7.1.40 Object 270E_h: Device analog inputs

Object Description			
Index	Name	Object Type	Data Type
270E _h	Device analog inputs	Variable	Record

Table 149: Object Description (270E_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	ADC_u	no	0	4095	0		R
2	ADC_v	no	0	4095	0		R
3	ADC_w	no	0	4095	0		R
4	ADC voltage	no	0	4095	0		R
5	ADC temperature	no	0	4095	0		R
6	ADC ain	no	0	4095	0		R

Table 150: Entry Description (270E_h)

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

8 Profile specific Area

The profile segment contains CiA-402 standard motion control objects. These objects control the motion control functions of the TMC-1690. Since it is not possible to operate the modes in parallel, the user is able to activate the required function by selecting a mode of operation. The control device writes to the modes of operation object in order to select the operation mode. The drive device provides the modes of operation display object to indicate the actual activated operation mode. Controlword, statusword, and set-points are used mode-specific. This implies the responsibility of the control device to avoid inconsistencies and erroneous behavior.

The following operating modes (selectable via object 6060_h, please see 8.1.6) are implemented on the TMC-1690:

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

8.1 Detailed Object Specifications

8.1.1 Object 605A_h: Quick Stop Option Code

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

Value Definition	
Value	Definition
1	Slow down on <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 151: Value Description (605A_h)

Object Description			
Index	Name	Object Type	Data Type
605A _h	Quick stop option code	Variable	SIGNED16

Table 152: Object Description (605A_h)

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

Table 153: Entry Description (605A_h)

8.1.2 Object 605B_h: Shutdown Option Code

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

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

Table 154: Value Description (605B_h)

Object Description			
Index	Name	Object Type	Data Type
605B _h	Shutdown option code	Variable	SIGNED16

Table 155: Object Description (605B_h)

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

Table 156: Entry Description (605B_h)

8.1.3 Object 605C_h: Disable Operation Option Code

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

Value Definition	
Value	Definition
1	Slow down on slow down ramp

Table 157: Value Description (605C_h)

Object Description			
Index	Name	Object Type	Data Type
605C _h	Disable operation option code	Variable	SIGNED16

Table 158: Object Description (605C_h)

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

Table 159: Entry Description (605C_h)

8.1.4 Object 605D_h: Halt Option Code

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

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

Table 160: Value Description (605D_h)

Object Description			
Index	Name	Object Type	Data Type
605D _h	Halt option code	Variable	SIGNED16

Table 161: Object Description (605D_h)

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

Table 162: Entry Description (605D_h)

8.1.5 Object 605E_h: Fault Reaction Option Code

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

Value Definition	
Value	Definition
2	Slow down on quick stop ramp

Table 163: Value Description (605E_h)

Object Description			
Index	Name	Object Type	Data Type
605E _h	Fault reaction option code	Variable	SIGNED16

Table 164: Object Description (605E_h)

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

Table 165: Entry Description (605E_h)

8.1.6 Object 6060_h: Modes of Operation

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

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

Table 166: Value Description (6060_h)

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

Object Description			
Index	Name	Object Type	Data Type
6060 _h	Modes of operation	Variable	SIGNED8

Table 167: Object Description (6060_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	see table 166	0

Table 168: Entry Description (6060_h)

8.1.7 Object 6061_h: Modes of Operation Display

This object shows the operating mode that is currently set.

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

Table 169: Value Description (6061_h)

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

Object Description			
Index	Name	Object Type	Data Type
6061 _h	Modes of operation display	Variable	SIGNED8

Table 170: Object Description (6061_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	see table 169	0

Table 171: Entry Description (6061_h)

8.1.8 Object 60FD_h: Digital Inputs

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

Object Description			
Index	Name	Object Type	Data Type
60FD _h	Digital inputs	Variable	UNSIGNED32

Table 172: Object Description (60FD_h)

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

Table 173: Entry Description (60FD_h)

8.1.9 Object 6502_h: Supported Drive Modes

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

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

Table 174: Value Definition (6502_h)

Object Description			
Index	Name	Object Type	Data Type
6502 _h	Supported drive modes	Variable	UNSIGNED32

Table 175: Object Description (6502_h)

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

Table 176: Entry Description (6502_h)

8.1.10 Object 67FF_h: Single Device Type

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

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

Table 177: Value Definition (67FF_h)

Object Description			
Index	Name	Object Type	Data Type
67ff _h	Single device type	Variable	UNSIGNED32

Table 178: Object Description (67FF_h)

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

Table 179: Entry Description (67FF_h)

9 Profile Position Mode

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

Please refer to object 6060_h (section 8.1.6) for information about how to choose an operation mode. Object 6061_h (section 8.1.7) shows the operation mode that is set.

9.1 Detailed Object Specifications

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

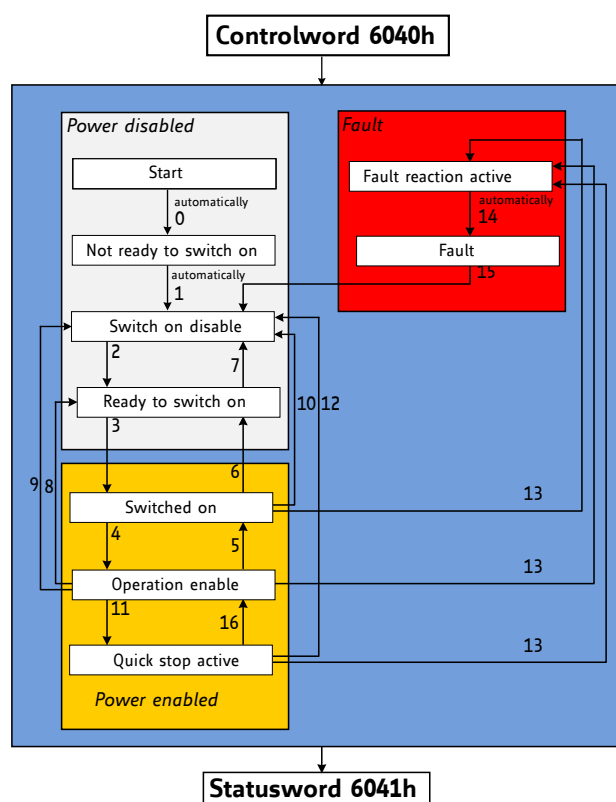


Figure 14: DS402 Finite State Machine

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.

9.1.1 Object 6040_n: Controlword

This object indicates the received command controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to figure 14 for detailed information.

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

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

Table 180: Structure of the Controlword in pp Mode

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

Table 181: Operation Mode specific Bits in pp Mode

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

Table 182: Command Coding

Object Description			
Index	Name	Object Type	Data Type
6040 _h	Controlword	Variable	UNSIGNED16

Table 183: Object Description (6040_h in pp Mode)

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

Table 184: Entry Description (6040_h in pp Mode)

9.1.2 Object 6041_h: Statusword

This object provides the status of the PDS FSA. It reflects the status of the CiA-402 state machine. Please refer to figure 14 for detailed information. The object is structured as defined below.

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

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

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

Table 185: Structure of the Status Word in pp Mode

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

Table 186: 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 187: Operation Mode specific Bits in pp Mode

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

Table 188: State Coding

Object Description			
Index	Name	Object Type	Data Type
6041 _h	Controlword	Variable	UNSIGNED16

Table 189: Object Description (6041_h in pp Mode)

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

Table 190: Entry Description (6041_h in pp Mode)

9.1.3 Object 6062_h: Position Demand Value

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

Object Description			
Index	Name	Object Type	Data Type
6062 _h	Position Demand Value	Variable	SIGNED32

Table 191: Object Description (6062_h)

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

Table 192: Entry Description (6062_h)

9.1.4 Object 6063_h: Position Actual Internal Value

This object provides the demanded position value. The value is given in PositionScaler steps (see ??). It is the same as object 6062_h.

Object Description			
Index	Name	Object Type	Data Type
6063 _h	Position Actual Internal Value	Variable	SIGNED32

Table 193: Object Description (6063_h)

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

Table 194: Entry Description (6063_h)

9.1.5 Object 6064_h: Position Actual Value

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

Object Description			
Index	Name	Object Type	Data Type
6064 _h	Position Actual Value	Variable	SIGNED32

Table 195: Object Description (6064_h)

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

Table 196: Entry Description (6064_h)

9.1.6 Object 6067_h: Position Window

This object indicates the configured symmetrical range of accepted positions relative to the target position. If the actual value of the position encoder is within the position window, this target position is regarded as

having been reached. The value is given in increments. If the value of the position window is FFFFFFFF_h, the position window control is switched off. If this object is set to zero, the target reached event will be signaled when the demand position (6062_h) has reached the target position (6064_h). When the position window is set to a value greater than zero, the target reached event will be signaled when the actual encoder position value (6064_h) is within $(target_position - position_window)$ and $(target_position + position_window)$.

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

Table 197: Object Description (6067_h)

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

Table 198: Entry Description (6067_h)

9.1.7 Object 6068_h: Position Window Time

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

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

Table 199: Object Description (6068_h)

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

Table 200: Entry Description (6068_h)

9.1.8 Object 606C_h: Velocity Actual Value

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

Object Description			
Index	Name	Object Type	Data Type
606C _h	Velocity Actual Value	Variable	SIGNED32

Table 201: Object Description (606C_h)

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

Table 202: Entry Description (606C_h)

9.1.9 Object 607A_h: Target Position

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

Object Description			
Index	Name	Object Type	Data Type
607A _h	Target Position	Variable	SIGNED32

Table 203: Object Description (607A_h in pp Mode)

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

Table 204: Entry Description (607A_h in pp Mode)

9.1.10 Object 607D_h: Software Position Limit

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

$$\text{Corrected_min_position_limit} = \text{min_position_limit} - \text{home_offset}$$

$$\text{Corrected_max_position_limit} = \text{max_position_limit} - \text{home_offset}$$

Object Description			
Index	Name	Object Type	Data Type
607D _h	Software Position Limit	Array	SIGNED32

Table 205: Object Description (607D_h)

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 206: Entry Description (607D_h)

9.1.11 Object 6081_h: Profile Velocity

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

Object Description			
Index	Name	Object Type	Data Type
6081 _h	Profile Velocity	Variable	UNSIGNED32

Table 207: Object Description (6081_h)

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

Table 208: Entry Description (6081_h)

9.1.12 Object 6083_h: Profile Acceleration

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

This value is given using RPM/s units.

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

Object Description			
Index	Name	Object Type	Data Type
6083 _h	Profile Acceleration	Variable	UNSIGNED32

Table 209: Object Description (6083_h)

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

Table 210: Entry Description (6083_h)

9.1.13 Object 6084_h: Profile Deceleration

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

Object Description			
Index	Name	Object Type	Data Type
6084 _h	Profile Deceleration	Variable	UNSIGNED32

Table 211: Object Description (6084_h)

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

Table 212: Entry Description (6084_h)

9.1.14 Object 6085_h: Quick Stop Deceleration

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

Object Description			
Index	Name	Object Type	Data Type
6085 _h	Quick stop deceleration	Variable	UNSIGNED32

Table 213: Object Description (6085_h)

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

Table 214: Entry Description (6085_h)

9.2 How to move a Motor in pp Mode

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

- If you do not have any limit switches connected, first disable the limit switch inputs by writing 3 to object 2005_h.
- Select pp mode by writing 1 to object 6060_h.
- Write 6 to object 6040_h to switch to READY_TO_SWITCH_ON state.
- Write 7 to object 6040_h to switch to SWITCHED_ON state.
- Write 15 to object 6040_h to switch to OPERATION_ENABLED state.
- Write the desired target position (e.g. 500000) to object 607A_h.
- Mark the new target position as active by writing 31 to object 6040_h. The motor starts moving now.
- Reset the activation by writing 15 to object 6040_h (this can be done while the motor is still moving).

10 Profile Velocity Mode

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

The profile velocity mode covers the following sub-functions:

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

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

- Profile velocity
- Profile acceleration
- Motion profile type

10.0.1 Object 6040_n: Controlword

This object indicates the received command controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to figure 14 for detailed information.

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

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

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

Table 215: Structure of the Controlword in pv Mode

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

Table 216: Command Coding

Object Description			
Index	Name	Object Type	Data Type
6040 _h	Controlword	Variable	UNSIGNED16

Table 217: Object Description (6040_h in pv Mode)

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

Table 218: Entry Description (6040_h in pv Mode)

10.0.2 Object 6041_h: Statusword

This object provides the status of the PDS FSA. It reflects the status of the CiA-402 state machine. Please refer to figure 14 for detailed information. The object is structured as defined below.

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

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

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

Table 219: Structure of the Statusword in pv Mode

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

Table 220: 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 221: Operation Mode specific Bits in pv Mode

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

Table 222: State Coding

Object Description			
Index	Name	Object Type	Data Type
6041 _h	Controlword	Variable	UNSIGNED16

Table 223: Object Description (6041_h in pv Mode)

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

Table 224: Entry Description (6041_h in pv Mode)

10.0.3 Object 6062_h: Position Demand Value

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

Object Description			
Index	Name	Object Type	Data Type
6062 _h	Position Demand Value	Variable	SIGNED32

Table 225: Object Description (6062_h)

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

Table 226: Entry Description (6062_h)

10.0.4 Object 6063_h: Position Actual Internal Value

This object provides the demanded position value. The value is given in PositionScaler steps (see ??). It is the same as object 6062_h.

Object Description			
Index	Name	Object Type	Data Type
6063 _h	Position Actual Internal Value	Variable	SIGNED32

Table 227: Object Description (6063_h)

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

Table 228: Entry Description (6063_h)

10.0.5 Object 6064_h: Position Actual Value

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

Object Description			
Index	Name	Object Type	Data Type
6064 _h	Position Actual Value	Variable	SIGNED32

Table 229: Object Description (6064_h)

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

Table 230: Entry Description (6064_h)

10.0.6 Object 606C_h: Velocity Actual Value

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

Object Description			
Index	Name	Object Type	Data Type
606C _h	Velocity Actual Value	Variable	SIGNED32

Table 231: Object Description (606C_h)

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

Table 232: Entry Description (606C_h)

10.0.7 Object 607D_h: Software Position Limit

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

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

$$\text{Corrected_min_position_limit} = \text{min_position_limit} - \text{home_offset}$$

$$\text{Corrected_max_position_limit} = \text{max_position_limit} - \text{home_offset}$$

Object Description			
Index	Name	Object Type	Data Type
607D _h	Software Position Limit	Array	SIGNED32

Table 233: Object Description (607D_h)

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 234: Entry Description (607D_h)

10.0.8 Object 6083_h: Profile Acceleration

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

This value is given using RPM/s units.

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

Object Description			
Index	Name	Object Type	Data Type
6083 _h	Profile Acceleration	Variable	UNSIGNED32

Table 235: Object Description (6083_h)

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

Table 236: Entry Description (6083_h)

10.0.9 Object 6085_h: Quick Stop Deceleration

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

Object Description			
Index	Name	Object Type	Data Type
6085 _h	Quick stop deceleration	Variable	UNSIGNED32

Table 237: Object Description (6085_h)

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

Table 238: Entry Description (6085_h)

10.0.10 Object 60FF_h: Target Velocity

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

Object Description			
Index	Name	Object Type	Data Type
60FF _h	Target Velocity	Variable	SIGNED32

Table 239: Object Description (60FF_h)

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

Table 240: Entry Description (60FF_h)

10.1 How to move a Motor in pv Mode

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

- If you do not have any limit switches connected, first disable the limit switch inputs by writing 3 to object 2005_h.

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

11 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 15 shows the defined input objects as well as the output objects. The user can specify the speeds, acceleration and method of homing. The home offset object 607C_h allows displacing the zero in point the coordinate system for the home position.

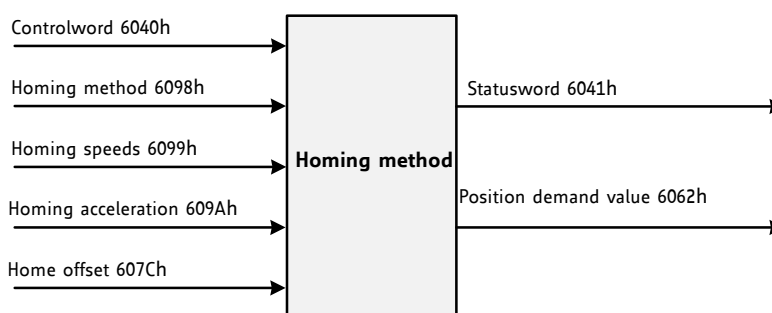


Figure 15: Homing Mode Function

Choosing a homing mode determines the following things:

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

The home position and the zero position are offset by the home offset (see object 607C_h, section 11.2.4).

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.

11.1 Homing Methods

The TMCM-1690 supports a subset of different standard CANopen homing methods. The homing method that is to be used can be chosen via object 6098_h (section 11.2.5).

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

Table 241: Supported CANopen Homing Methods

When using homing methods that need end switch inputs or home switch inputs please take care of their configuration (object 2005_h, section 7.1.3).

11.1.1 Homing Method 17: Homing on negative Limit Switch

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

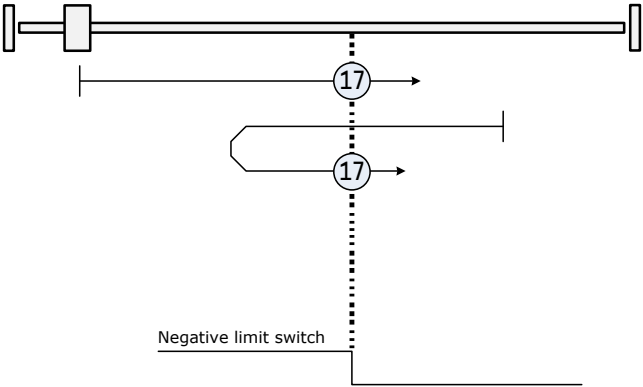


Figure 16: Homing Method 17

11.1.2 Homing Method 18: Homing on positive Limit Switch

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

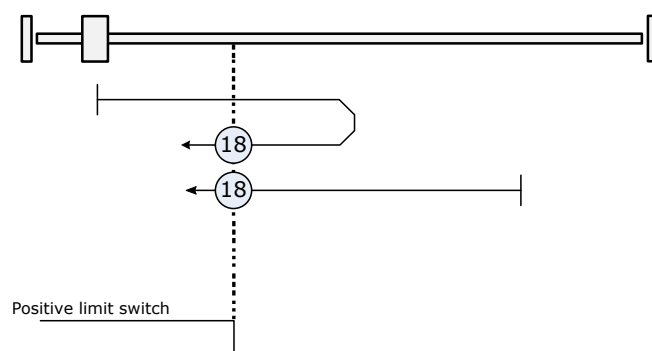


Figure 17: Homing Method 18

11.1.3 Homing Method 19: Homing on positive Home Switch

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

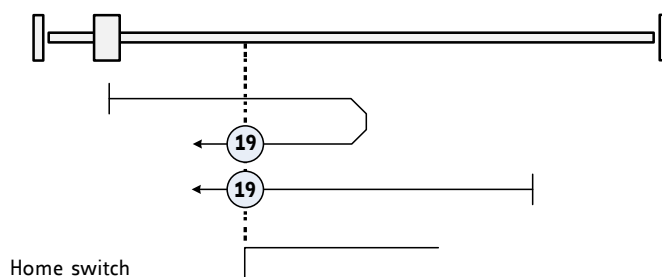


Figure 18: Homing Method 19

11.1.4 Homing Method 21: Homing on negative Home Switch

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

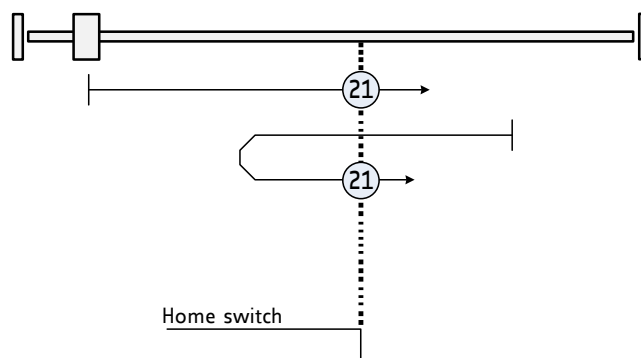


Figure 19: Homing Method 21

11.1.5 Homing Method 35: Current Position as Home Position

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

11.2 Detailed Object Specifications

11.2.1 Object 6040_h: Controlword

This object indicates the received command controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to figure 14 for detailed information.

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

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

Table 242: Structure of the Controlword in hm Mode

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

Table 243: Operation Mode specific Bits in hm Mode

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

Table 244: Command Coding

Object Description			
Index	Name	Object Type	Data Type
6040 _h	Controlword	Variable	UNSIGNED16

Table 245: Object Description (6040_h in hm Mode)

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

Table 246: Entry Description (6040_h in hm Mode)

11.2.2 Object 6041_h: Statusword

This object provides the status of the PDS FSA. It reflects the status of the CiA-402 state machine. Please refer to figure 14 for detailed information. The object is structured as defined below.

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

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

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

Table 247: Structure of the Statusword in hm Mode

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

Table 248: 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 249: Operation Mode specific Bits in hm Mode

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

Table 250: State Coding

Object Description			
Index	Name	Object Type	Data Type
6041 _h	Controlword	Variable	UNSIGNED16

Table 251: Object Description (6041_h in hm Mode)

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

Table 252: Entry Description (6041_h in hm Mode)

11.2.3 Object 606C_h: Velocity Actual Value

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

Object Description			
Index	Name	Object Type	Data Type
606C _h	Velocity Actual Value	Variable	SIGNED32

Table 253: Object Description (606C_h)

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

Table 254: Entry Description (606C_h)

11.2.4 Object 607C_h: Home Offset

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

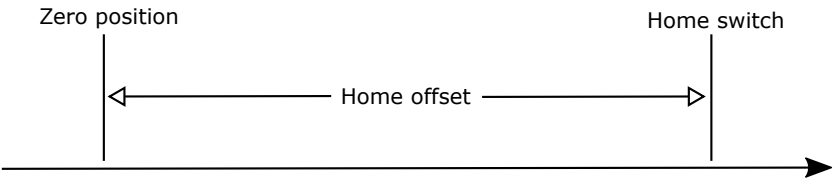


Figure 20: Home Offset

Object Description			
Index	Name	Object Type	Data Type
607C _h	Home offset	Variable	SIGNED32

Table 255: Object Description (607C_h)

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

Table 256: Entry Description (607C_h)

11.2.5 Object 6098_h: Homing Method

The homing method to be used can be selected by writing to this object. Please see table 241 for a list of homing methods supported by the current version of the TMC-1690 CANopen firmware.

Object Description			
Index	Name	Object Type	Data Type
6098 _h	Homing method	Variable	SIGNED8

Table 257: Object Description (6098_h)

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

Table 258: Entry Description (6098_h)

11.2.6 Object 6099_h: Homing Speeds

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

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

Table 259: Object Description (6099_h)

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 260: Entry Description (6099_h)

11.2.7 Object 609A_h: Homing Acceleration

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

Object Description			
Index	Name	Object Type	Data Type
609A _h	Homing acceleration	Variable	UNSIGNED32

Table 261: Object Description (609A_h)

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

Table 262: Entry Description (609A_h)

11.3 How to start a Homing in hm Mode

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

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

12 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 manager), not in the drive device. In cyclic synchronous manner, the control device provides a target position to the drive device, which performs position control, velocity control and torque control.

The main control parameters are the target position (object 607A_h, see section 12.1.7) and the interpolation time period (object 60C2_h, see section 12.1.10). 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.

12.1 Detailed Object Specifications

12.1.1 Object 6040_h: Controlword

This object indicates the received command controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to figure 14 for detailed information. The cyclic synchronous position mode does not use any mode specific bits of the Controlword.

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

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

Table 263: Structure of the Controlword in csp Mode

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

Table 264: Command Coding

Object Description			
Index	Name	Object Type	Data Type
6040 _h	Controlword	Variable	UNSIGNED16

Table 265: Object Description (6040_h in csp Mode)

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

Table 266: Entry Description (6040_h in csp Mode)

12.1.2 Object 6041_h: Statusword

This object provides the status of the PDS FSA. It reflects the status of the CiA-402 state machine. Please refer to figure 14 for detailed information. The object is structured as defined below. For more information about the coding please refer to the CANopen Drives and motion control device profile, part 2.

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

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

Table 267: Structure of the Statusword in csp Mode

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

Table 268: 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 269: Operation Mode specific Bits in csp Mode

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

Table 270: State Coding

Object Description			
Index	Name	Object Type	Data Type
6041 _h	Controlword	Variable	UNSIGNED16

Table 271: Object Description (6041_h in csp Mode)

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

Table 272: Entry Description (6041_h in csp Mode)

12.1.3 Object 6062_h: Position Demand Value

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

Object Description			
Index	Name	Object Type	Data Type
6062 _h	Position Demand Value	Variable	SIGNED32

Table 273: Object Description (6062_h)

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

Table 274: Entry Description (6062_h)

12.1.4 Object 6063_h: Position Actual Internal Value

This object provides the demanded position value. The value is given in PositionScaler steps (see ??). It is the same as object 6062_h.

Object Description			
Index	Name	Object Type	Data Type
6063 _h	Position Actual Internal Value	Variable	SIGNED32

Table 275: Object Description (6063_h)

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

Table 276: Entry Description (6063_h)

12.1.5 Object 6064_h: Position Actual Value

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

Object Description			
Index	Name	Object Type	Data Type
6064 _h	Position Actual Value	Variable	SIGNED32

Table 277: Object Description (6064_h)

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

Table 278: Entry Description (6064_h)

12.1.6 Object 606C_h: Velocity Actual Value

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

Object Description			
Index	Name	Object Type	Data Type
606C _h	Velocity Actual Value	Variable	SIGNED32

Table 279: Object Description (606C_h)

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

Table 280: Entry Description (606C_h)

12.1.7 Object 607A_h: Target Position

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

Object Description			
Index	Name	Object Type	Data Type
607A _h	Target Position	Variable	SIGNED32

Table 281: Object Description (607A_h in csp Mode)

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

Table 282: Entry Description (607A_h in csp Mode)

12.1.8 Object 607D_h: Software Position Limit

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

$$\begin{aligned} \text{Corrected_min_position_limit} &= \text{min_position_limit} - \text{home_offset} \\ \text{Corrected_max_position_limit} &= \text{max_position_limit} - \text{home_offset} \end{aligned}$$

Object Description			
Index	Name	Object Type	Data Type
607D _h	Software Position Limit	Array	SIGNED32

Table 283: Object Description (607D_h)

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 284: Entry Description (607D_h)

12.1.9 Object 60B0_h: Position Offset

This object provides an offset to the target position (object 607A_h, see section 12.1.7)). The value is given in PositionScaler steps (see ??) and will be added to the target position.

Object Description			
Index	Name	Object Type	Data Type
60B0 _h	Position Offset	Variable	SIGNED32

Table 285: Object Description (60B0_h)

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

Table 286: Entry Description (60B0_h)

12.1.10 Object 60C2_h: Interpolation Time Period

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

Object Description			
Index	Name	Object Type	Data Type
60C2 _h	Interpolation Time Period	Record	Interpolation time period record (0080 _h)

Table 287: Object Description (60C2_h)

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 288: Entry Description (60C2_h)

13 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 manager), not in the drive device. In cyclic synchronous manner, the control device provides a target velocity to the drive device, which performs position control, velocity control and torque control.

The main control parameters are the target velocity (object 60FF_h, see section 13.1.4) and the interpolation time period (object 60C2_h, see section 13.1.7). 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.

13.1 Detailed Object Specifications

13.1.1 Object 6040_h: Controlword

This object indicates the received command controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to figure 14 for detailed information. The cyclic synchronous velocity mode does not use any mode specific bits of the Controlword.

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

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

Table 289: Structure of the Controlword in csv Mode

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

Table 290: Command Coding

Object Description			
Index	Name	Object Type	Data Type
6040 _h	Controlword	Variable	UNSIGNED16

Table 291: Object Description (6040_h in csv Mode)

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

Table 292: Entry Description (6040_h in csv Mode)

13.1.2 Object 6041_h: Statusword

This object provides the status of the PDS FSA. It reflects the status of the CiA-402 state machine. Please refer to figure 14 for detailed information. The object is structured as defined below.

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

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

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

Table 293: Structure of the Statusword in csv Mode

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

Table 294: 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 295: Operation Mode specific Bits in csv Mode

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

Table 296: State Coding

Object Description			
Index	Name	Object Type	Data Type
6041 _h	Controlword	Variable	UNSIGNED16

Table 297: Object Description (6041_h in csv Mode)

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

Table 298: Entry Description (6041_h in csv Mode)

13.1.3 Object 606C_h: Velocity Actual Value

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

Object Description			
Index	Name	Object Type	Data Type
606C _h	Velocity Actual Value	Variable	SIGNED32

Table 299: Object Description (606C_h)

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

Table 300: Entry Description (606C_h)

13.1.4 Object 60FF_h: Target Velocity

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

Object Description			
Index	Name	Object Type	Data Type
60FF _h	Target Velocity	Variable	SIGNED32

Table 301: Object Description (60FF_h)

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

Table 302: Entry Description (60FF_h)

13.1.5 Object 607D_h: Software Position Limit

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

$$\begin{aligned} \text{Corrected_min_position_limit} &= \text{min_position_limit} - \text{home_offset} \\ \text{Corrected_max_position_limit} &= \text{max_position_limit} - \text{home_offset} \end{aligned}$$

Object Description			
Index	Name	Object Type	Data Type
607D _h	Software Position Limit	Array	SIGNED32

Table 303: Object Description (607D_h)

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 304: Entry Description (607D_h)

13.1.6 Object 60B1_h: Velocity Offset

This object provides an offset to the target velocity (object 60FF_h, see section 13.1.4)). The value will be added to the target velocity.

Object Description			
Index	Name	Object Type	Data Type
60B1 _h	Velocity Offset	Variable	SIGNED32

Table 305: Object Description (60B1_h)

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

Table 306: Entry Description (60B1_h)

13.1.7 Object 60C2_h: Interpolation Time Period

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

Object Description			
Index	Name	Object Type	Data Type
60C2 _h	Interpolation Time Period	Record	Interpolation time period record (0080 _h)

Table 307: Object Description (60C2_h)

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 308: Entry Description (60C2_h)

14 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.

14.1 Detailed Object Specifications

14.1.1 Object 6040_n: Controlword

This object indicates the received command controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to figure 14 for detailed information. The cyclic synchronous torque mode does not use any mode specific bits of the Controlword.

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

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

Table 309: Structure of the Controlword in cst Mode

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

Table 310: Command Coding

Object Description			
Index	Name	Object Type	Data Type
6040 _h	Controlword	Variable	UNSIGNED16

Table 311: Object Description (6040_h in cst Mode)

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

Table 312: Entry Description (6040_h in cst Mode)

14.1.2 Object 6041_h: Statusword

This object provides the status of the PDS FSA. It reflects the status of the CiA-402 state machine. Please refer to figure 14 for detailed information. The object is structured as defined below.

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

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

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

Table 313: Structure of the Statusword in cst Mode

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

Table 314: 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 315: Operation Mode specific Bits in cst Mode

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

Table 316: State Coding

Object Description			
Index	Name	Object Type	Data Type
6041 _h	Controlword	Variable	UNSIGNED16

Table 317: Object Description (6041_h in cst Mode)

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

Table 318: Entry Description (6041_h in cst Mode)

14.1.3 Object 6062_h: Position Demand Value

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

Object Description			
Index	Name	Object Type	Data Type
6062 _h	Position Demand Value	Variable	SIGNED32

Table 319: Object Description (6062_h)

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

Table 320: Entry Description (6062_h)

14.1.4 Object 6063_h: Position Actual Internal Value

This object provides the demanded position value. The value is given in PositionScaler steps (see ??). It is the same as object 6062_h.

Object Description			
Index	Name	Object Type	Data Type
6063 _h	Position Actual Internal Value	Variable	SIGNED32

Table 321: Object Description (6063_h)

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

Table 322: Entry Description (6063_h)

14.1.5 Object 6064_h: Position Actual Value

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

Object Description			
Index	Name	Object Type	Data Type
6064 _h	Position Actual Value	Variable	SIGNED32

Table 323: Object Description (6064_h)

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

Table 324: Entry Description (6064_h)

14.1.6 Object 6071_h: Target Torque

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

Object Description			
Index	Name	Object Type	Data Type
6071 _h	Target torque	Variable	INTEGER16

Table 325: Object Description (6071_h)

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

Table 326: Entry Description (6071_h)

14.1.7 Object 6077_h: Torque actual Value

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

Object Description			
Index	Name	Object Type	Data Type
6077 _h	Torque actual Value	Variable	INTEGER16

Table 327: Object Description (6077_h)

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

Table 328: Entry Description (6077_h)

14.1.8 Object 607D_h: Software Position Limit

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

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

$$\begin{aligned} \text{Corrected_min_position_limit} &= \text{min_position_limit} - \text{home_offset} \\ \text{Corrected_max_position_limit} &= \text{max_position_limit} - \text{home_offset} \end{aligned}$$

Object Description			
Index	Name	Object Type	Data Type
607D _h	Software Position Limit	Array	SIGNED32

Table 329: Object Description (607D_h)

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 330: Entry Description (607D_h)

14.1.9 Object 60B2_h: Torque Offset

This object provides an offset to the torque value. It will be added to the target torque (object 6071_h, see section 14.1.6).

Object Description			
Index	Name	Object Type	Data Type
60B2 _h	Torque Offset	Variable	SIGNED16

Table 331: Object Description (60B2_h)

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

Table 332: Entry Description (60B2_h)

14.1.10 Object 60C2_h: Interpolation Time Period

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

Object Description			
Index	Name	Object Type	Data Type
60C2 _h	Interpolation Time Period	Record	Interpolation time period record (0080 _h)

Table 333: Object Description (60C2_h)

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 334: Entry Description (60C2_h)

15 Emergency Messages (EMCY)

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

15.1 Format of EMCY Messages

The COB-ID of an EMCY is 80_h+ID of the node. So for example it is 81_h for node #1. The first two bytes contain the error code with LSB first. The third data byte contains the content of the error register (object 1001_h). The other five bytes can contain additional information, depending on the error code.

EMCY							
COB-ID	1	2	3	4	5	6	7 8
0x80 _h +Node ID	Error code		Error register	Additional bytes			
	LSB	MSB	(1001 _h)	#1...#5			

Table 335: Format of EMCY Messages

15.2 Error Codes used by the TMC1690 Module

The following table shows all EMCY error codes used by the TMC1690 module.

Emergency Messages (EMCY) of the TMC1690						
Error code	Additional byte					Description
	1	2	3	4	5	
0000 _h	0	0	0	0	0	Fault reset The fault reset command has been executed.
4310 _h	2	0	0	0	0	Overtemperature error The motor driver has been switched off because the temperature limit has been exceeded.
5441 _h	0	255	0	0	0	Shutdown switch active The enable signal is missing (due to the shutdown switch) and the motor driver has been switched off.
6320 _h	0	255	0	0	0	Parameter error The data in the received PDO is either wrong or cannot be accepted due to the internal state of the drive.
8100 _h	0	255	0	0	0	Communication error General CAN bus communication error.
8110 _h	1	255	0	0	0	CAN controller overflow The receive message buffer of the CAN controller hardware is full and some CAN messages are lost.
8110 _h	2	255	0	0	0	CAN Tx buffer overflow The software CAN transmit buffer is full and thus some CAN messages are lost.

Error code	Additional byte					Description
	1	2	3	4	5	
8110 _h	3	255	0	0	0	CAN Rx buffer overflow The software CAN receive buffer is full and so some CAN messages are lost.
8120 _h	0	255	0	0	0	CAN error passive The CAN controller has detected communication errors and has entered the CAN Error passive state.
8130 _h	0	255	0	0	0	Heartbeat or lifeguard error The module did not receive a heartbeat or lifeguard message in time.
8140 _h	0	255	0	0	0	CAN controller recovered from bus-off state The CAN controller has detected too many errors and has changed into the bus-off state. The drive has been stopped and disabled. This message is sent after the CAN controller has recovered from bus-off state and is bus-on again.
8210 _h	0	255	0	0	0	PDO not processed due to length error A PDO sent to the module could not be processed because too few bytes were supplied.
8220 _h	0	255	0	0	0	PDO length exceeded A PDO sent to the module could not be processed because too many bytes were supplied.
8611 _h	0	0	0	0	0	Following error The deviation between motor position counter and encoder position counter has exceeded the following error window.
ff00 _h	0	0	0	0	0	Undervoltage The supply voltage is too low to drive a motor.
ff01 _h	1	0	0	0	0	Positive software limit The actual position is outside the range defined by object 607d _h .
ff01 _h	2	0	0	0	0	Negative software limit The actual position is outside the range defined by object 607d _h .
ff01 _h	3	0	0	0	0	Positive limit switch The positive limit switch has been touched outside of the homing function.
ff01 _h	4	0	0	0	0	Negative limit switch The negative limit switch has been touched outside of the homing function.

Table 336: Emergency Messages (EMCY)

16 SDO Abort Codes

Trying to access an object via SDO read or SDO write may result in an error. In such a case an SDO abort transfer message containing an abort code will be sent. The following table lists all SDO abort codes defined by the CiA-301 standard. Not all of these are used by the TMCM-1690 module.

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

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

Table 337: SDO Abort Codes

17 Firmware Update

For an update of the TMCM-1690's firmware the TMCL-IDE ¹ is used.

The required update files can be downloaded from the TMCM-1690's product page ².

An update over CAN is only possible via the Trinamic TMCL CAN protocol, because of that, prior to starting the upload, the TMCM-1690 must be put into boot mode by writing the value 12345678_h into object 5FFF_h.

Note Make sure to update only CANopen firmware files.

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

²TMCM-1690 product page: <https://www.trinamic.com/products/modules/details/tmcm-1617/>

18 Figures Index

1	Motor Regulation	7	11	NMT State Machine	21
2	Current Regulation	8	12	Communication Architecture	22
3	Velocity Regulation	9	13	Device Model	23
4	Positioning Regulation	10	14	DS402 Finite State Machine	78
5	Positioning Algorithm	11	15	Homing Mode Function	96
6	Linear shaped ramp (left) S Shaped ramp (Right)	12	16	Homing Method 17	97
7	Filter blocks	14	17	Homing Method 18	98
8	Mechanical brake settings	16	18	Homing Method 19	98
9	Brake chopper example schematic . .	17	19	Homing Method 21	98
10	Ilt Monitor Windows	17	20	Home Offset	103

19 Tables Index

1	Abbreviations used in this Manual . . .	6	53	Object Description (2000 _h)	39
2	Current Regulation Parameters	8	54	Entry Description (2000 _h)	39
3	Velocity Regulation Parameters	9	55	Parameter description (2000 _h)	40
4	Position Regulation Parameters	10	56	Object Description (2003 _h)	40
5	Service Primitives	19	57	Entry Description (2003 _h)	40
6	Service Types	20	58	Parameter Description (2003 _h)	41
7	Object Dictionary	23	59	Object Description (2005 _h)	41
8	Object Description (1000 _h)	25	60	Entry Description (2005 _h)	42
9	Entry Description (1000 _h)	25	61	Object Description (200A _h)	42
10	Object Description (1001 _h)	25	62	Entry Description (200A _h)	42
11	Entry Description (1001 _h)	25	63	Object Description (200F _h)	43
12	Error Register Bits	26	64	Entry Description (200F _h)	43
13	Value Definition (1005 _h)	26	65	Parameter Description (200F _h)	44
14	Object Description (1005 _h)	26	66	Object Description (2014 _h)	44
15	Entry Description (1005 _h)	26	67	Entry Description (2014 _h)	45
16	Object Description (1008 _h)	27	68	Parameter Description (2014 _h)	46
17	Entry Description (1008 _h)	27	69	Object Description (2015 _h)	46
18	Object Description (1009 _h)	27	70	Entry Description (2015 _h)	46
19	Entry Description (1009 _h)	27	71	Parameter Description (2015 _h)	47
20	Object Description (100A _h)	27	72	Object Description (2019 _h)	47
21	Entry Description (100A _h)	28	73	Entry Description (2019 _h)	47
22	Object Description (100C _h)	28	74	Parameter Description (2019 _h)	48
23	Entry Description (100C _h)	28	75	Object Description (201E _h)	48
24	Object Description (100D _h)	28	76	Entry Description (201E _h)	49
25	Entry Description (100D _h)	28	77	Parameter Description (201E _h)	49
26	Save Signature	29	78	Object Description (2023 _h)	49
27	Object Description (1010 _h)	29	79	Entry Description (2023 _h)	50
28	Entry Description (1010 _h)	30	80	Parameter Description (2023 _h)	50
29	Load Signature	30	81	Object Description (2028 _h)	50
30	Object Description (1011 _h)	31	82	Entry Description (2028 _h)	51
31	Entry Description (1011 _h)	31	83	Parameter Description (2028 _h)	51
32	Object Description (1014 _h)	31	84	Object Description (202B _h)	51
33	Entry Description (1014 _h)	31	85	Entry Description (202B _h)	52
34	Object Description (1015 _h)	32	86	Parameter Description (202B _h)	52
35	Entry Description (1015 _h)	32	87	Object Description (202D _h)	52
36	Value Definition (1016 _h)	32	88	Entry Description (202D _h)	53
37	Object Description (1016 _h)	32	89	Object Description (2032 _h)	53
38	Entry Description (1016 _h)	32	90	Entry Description (2032 _h)	53
39	Object Description (1017 _h)	33	91	Object Description (2037 _h)	54
40	Entry Description (1017 _h)	33	92	Entry Description (2037 _h)	54
41	Object Description (1018 _h)	33	93	Object Description (203C _h)	54
42	Entry Description (1018 _h)	33	94	Entry Description (203C _h)	55
43	Object Description (1029 _h)	34	95	Object Description (2041 _h)	55
44	Entry Description (1029 _h)	34	96	Entry Description (2041 _h)	55
45	Object Description (1400 _h)	35	97	Parameter Description (2041 _h)	56
46	Entry Description (1400 _h)	35	98	Object Description (2046 _h)	56
47	Object Description (1600 _h)	35	99	Entry Description (2046 _h)	56
48	Entry Description (1600 _h)	36	100	Parameter Description (2046 _h)	56
49	Object Description (1800 _h)	37	101	Object Description (204B _h)	57
50	Entry Description (1800 _h)	37	102	Entry Description (204B _h)	57
51	Object Description (1A00 _h)	38	103	Parameter Description (204B _h)	57
52	Entry Description (1A00 _h)	38	104	Object Description (2050 _h)	57

105	Entry Description (2050 _h)	58	159	Entry Description (605C _h)	72
106	Parameter Description (2050 _h)	58	160	Value Description (605D _h)	72
107	Object Description (2055 _h)	58	161	Object Description (605D _h)	72
108	Entry Description (2055 _h)	59	162	Entry Description (605D _h)	72
109	Object Description (205A _h)	59	163	Value Description (605E _h)	73
110	Entry Description (205A _h)	60	164	Object Description (605E _h)	73
111	Parameter Description (205A _h)	60	165	Entry Description (605E _h)	73
112	Object Description (205F _h)	60	166	Value Description (6060 _h)	73
113	Entry Description (205F _h)	61	167	Object Description (6060 _h)	74
114	Parameter Description (205F _h)	61	168	Entry Description (6060 _h)	74
115	Object Description (2064 _h)	61	169	Value Description (6061 _h)	74
116	Entry Description (2064 _h)	61	170	Object Description (6061 _h)	74
117	Object Description (2069 _h)	62	171	Entry Description (6061 _h)	75
118	Entry Description (2069 _h)	62	172	Object Description (60FD _h)	75
119	Object Description (206B _h)	62	173	Entry Description (60FD _h)	75
120	Entry Description (206B _h)	62	174	Value Definition (6502 _h)	76
121	Object Description (206C _h)	63	175	Object Description (6502 _h)	76
122	Entry Description (206C _h)	63	176	Entry Description (6502 _h)	76
123	Object Description (206D _h)	63	177	Value Definition (67FF _h)	77
124	Entry Description (206D _h)	63	178	Object Description (67FF _h)	77
125	Object Description (206E _h)	64	179	Entry Description (67FF _h)	77
126	Entry Description (206E _h)	64	180	Structure of the Controlword in pp Mode	79
127	Object Description (207D _h)	64	181	Operation Mode specific Bits in pp Mode	79
128	Entry Description (207D _h)	64	182	Command Coding	79
129	Object Description (2082 _h)	64	183	Object Description (6040 _h in pp Mode)	79
130	Entry Description (2082 _h)	65	184	Entry Description (6040 _h in pp Mode)	80
131	Object Description (2701 _h)	65	185	Structure of the Status Word in pp Mode	80
132	Entry Description (2701 _h)	65	186	Trinamic Specific Bits	80
133	Bit Definitions (2702 _h)	65	187	Operation Mode specific Bits in pp Mode	80
134	Object Description (2702 _h)	66	188	State Coding	81
135	Entry Description (2702 _h)	66	189	Object Description (6041 _h in pp Mode)	81
136	Bit Definitions (2703 _h)	66	190	Entry Description (6041 _h in pp Mode)	81
137	Object Description (2703 _h)	66	191	Object Description (6062 _h)	81
138	Entry Description (2703 _h)	67	192	Entry Description (6062 _h)	82
139	Object Description (2704 _h)	67	193	Object Description (6063 _h)	82
140	Entry Description (2704 _h)	67	194	Entry Description (6063 _h)	82
141	Object Description (2705 _h)	67	195	Object Description (6064 _h)	82
142	Entry Description (2705 _h)	68	196	Entry Description (6064 _h)	82
143	Object Description (2707 _h)	68	197	Object Description (6067 _h)	83
144	Entry Description (2707 _h)	68	198	Entry Description (6067 _h)	83
145	Object Description (2708 _h)	68	199	Object Description (6068 _h)	83
146	Entry Description (2708 _h)	68	200	Entry Description (6068 _h)	83
147	Object Description (270B _h)	69	201	Object Description (606C _h)	84
148	Entry Description (270B _h)	69	202	Entry Description (606C _h)	84
149	Object Description (270E _h)	69	203	Object Description (607A _h in pp Mode)	84
150	Entry Description (270E _h)	69	204	Entry Description (607A _h in pp Mode)	84
151	Value Description (605A _h)	70	205	Object Description (607D _h)	85
152	Object Description (605A _h)	70	206	Entry Description (607D _h)	85
153	Entry Description (605A _h)	71	207	Object Description (6081 _h)	85
154	Value Description (605B _h)	71	208	Entry Description (6081 _h)	85
155	Object Description (605B _h)	71	209	Object Description (6083 _h)	86
156	Entry Description (605B _h)	71			
157	Value Description (605C _h)	71			
158	Object Description (605C _h)	72			

210	Entry Description (6083 _h)	86	260	Entry Description (6099 _h)	104
211	Object Description (6084 _h)	86	261	Object Description (609A _h)	105
212	Entry Description (6084 _h)	86	262	Entry Description (609A _h)	105
213	Object Description (6085 _h)	86	263	Structure of the Controlword in csp Mode	106
214	Entry Description (6085 _h)	87	264	Command Coding	107
215	Structure of the Controlword in pv Mode	88	265	Object Description (6040 _h in csp Mode)	107
216	Command Coding	89	266	Entry Description (6040 _h in csp Mode)	107
217	Object Description (6040 _h in pv Mode)	89	267	Structure of the Statusword in csp Mode	107
218	Entry Description (6040 _h in pv Mode)	89	268	Trinamic Specific Bits	108
219	Structure of the Statusword in pv Mode	90	269	Operation Mode specific Bits in csp Mode	108
220	Trinamic Specific Bits	90	270	State Coding	108
221	Operation Mode specific Bits in pv Mode	90	271	Object Description (6041 _h in csp Mode)	108
222	State Coding	90	272	Entry Description (6041 _h in csp Mode)	109
223	Object Description (6041 _h in pv Mode)	91	273	Object Description (6062 _h)	109
224	Entry Description (6041 _h in pv Mode)	91	274	Entry Description (6062 _h)	109
225	Object Description (6062 _h)	91	275	Object Description (6063 _h)	109
226	Entry Description (6062 _h)	91	276	Entry Description (6063 _h)	109
227	Object Description (6063 _h)	91	277	Object Description (6064 _h)	110
228	Entry Description (6063 _h)	92	278	Entry Description (6064 _h)	110
229	Object Description (6064 _h)	92	279	Object Description (606C _h)	110
230	Entry Description (6064 _h)	92	280	Entry Description (606C _h)	110
231	Object Description (606C _h)	92	281	Object Description (607A _h in csp Mode)	110
232	Entry Description (606C _h)	92	282	Entry Description (607A _h in csp Mode)	111
233	Object Description (607D _h)	93	283	Object Description (607D _h)	111
234	Entry Description (607D _h)	93	284	Entry Description (607D _h)	111
235	Object Description (6083 _h)	93	285	Object Description (60B0 _h)	111
236	Entry Description (6083 _h)	93	286	Entry Description (60B0 _h)	112
237	Object Description (6085 _h)	94	287	Object Description (60C2 _h)	112
238	Entry Description (6085 _h)	94	288	Entry Description (60C2 _h)	112
239	Object Description (60FF _h)	94	289	Structure of the Controlword in csv Mode	113
240	Entry Description (60FF _h)	94	290	Command Coding	114
241	Supported CANopen Homing Methods	97	291	Object Description (6040 _h in csv Mode)	114
242	Structure of the Controlword in hm Mode	100	292	Entry Description (6040 _h in csv Mode)	114
243	Operation Mode specific Bits in hm Mode	100	293	Structure of the Statusword in csv Mode	115
244	Command Coding	100	294	Trinamic Specific Bits	115
245	Object Description (6040 _h in hm Mode)	101	295	Operation Mode specific Bits in csv Mode	115
246	Entry Description (6040 _h in hm Mode)	101	296	State Coding	115
247	Structure of the Statusword in hm Mode	101	297	Object Description (6041 _h in csv Mode)	116
248	Trinamic Specific Bits	101	298	Entry Description (6041 _h in csv Mode)	116
249	Operation Mode specific Bits in hm Mode	102	299	Object Description (606C _h)	116
250	State Coding	102	300	Entry Description (606C _h)	116
251	Object Description (6041 _h in hm Mode)	102	301	Object Description (60FF _h)	116
252	Entry Description (6041 _h in hm Mode)	102	302	Entry Description (60FF _h)	117
253	Object Description (606C _h)	103	303	Object Description (607D _h)	117
254	Entry Description (606C _h)	103	304	Entry Description (607D _h)	117
255	Object Description (607C _h)	103	305	Object Description (60B1 _h)	117
256	Entry Description (607C _h)	103	306	Entry Description (60B1 _h)	118
257	Object Description (6098 _h)	104	307	Object Description (60C2 _h)	118
258	Entry Description (6098 _h)	104	308	Entry Description (60C2 _h)	118
259	Object Description (6099 _h)	104			

309	Structure of the Controlword in cst Mode	119	324	Entry Description (6064 _h)	123
310	Command Coding	119	325	Object Description (6071 _h)	123
311	Object Description (6040 _h in cst Mode)	120	326	Entry Description (6071 _h)	123
312	Entry Description (6040 _h in cst Mode)	120	327	Object Description (6077 _h)	123
313	Structure of the Statusword in cst Mode	120	328	Entry Description (6077 _h)	123
314	Trinamic Specific Bits	120	329	Object Description (607D _h)	124
315	Operation Mode specific Bits in cst Mode	121	330	Entry Description (607D _h)	124
316	State Coding	121	331	Object Description (60B2 _h)	124
317	Object Description (6041 _h in cst Mode)	121	332	Entry Description (60B2 _h)	124
318	Entry Description (6041 _h in cst Mode)	121	333	Object Description (60C2 _h)	125
319	Object Description (6062 _h)	122	334	Entry Description (60C2 _h)	125
320	Entry Description (6062 _h)	122	335	Format of EMCY Messages	126
321	Object Description (6063 _h)	122	336	Emergency Messages (EMCY)	127
322	Entry Description (6063 _h)	122	337	SDO Abort Codes	129
323	Object Description (6064 _h)	122	338	Firmware Revision	135
			339	Document Revision	135

20 Supplemental Directives

20.1 Producer Information

20.2 Copyright

TRINAMIC owns the content of this user manual in its entirety, including but not limited to pictures, logos, trademarks, and resources. © Copyright 2023 TRINAMIC. All rights reserved. Electronically published by TRINAMIC, Germany.

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

20.3 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.

20.4 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.

20.5 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.

20.6 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.

20.7 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.analog.com.

21 Revision History

21.1 Firmware Revision

Version	Date	Description
1.00	06/23	First release
1.01	07/23	New release: • Fixed on the fly change of motion parameters • Fixed issue with the block hall

Table 338: Firmware Revision

21.2 Document Revision

Version	Date	Description
0	07/23	First release

Table 339: Document Revision