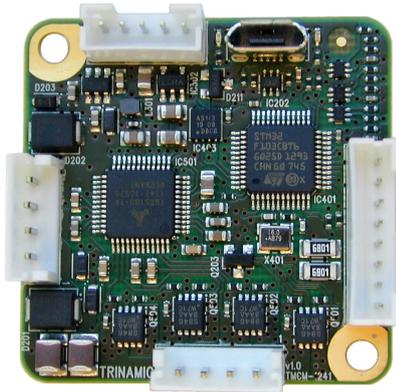# TMCM-1241 CANopen® Firmware Manual

Firmware Version V3.23 | Document Revision V1.02 • 2021-JUL-30

The TMCM-1241 is a single axis controller/driver module for 2-phase bipolar stepper motors. The TMCM-1241 CANopen® firmware allows to control the module using the CANopen® protocol, making use of the Trinamic TMC5160 motion controller and motor driver. Dynamic current control, and quiet, smooth and efficient operation are combined with StealthChop™, DcStep™, StallGuard™ and CoolStep™ features.
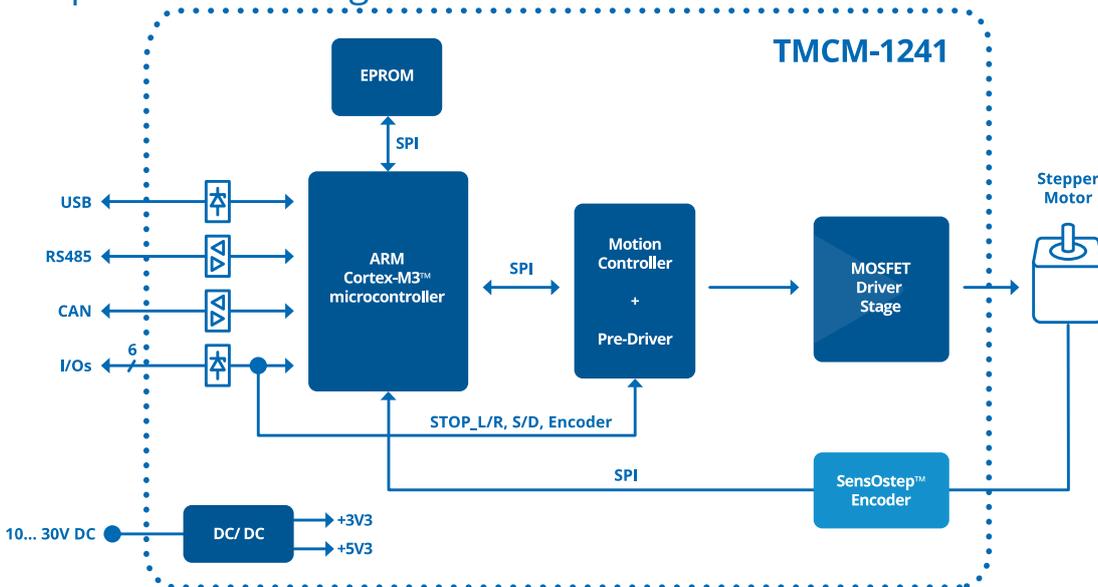
Features

- Single axis Stepper Motor Control
- CANopen® CiA-402 Drive Profile
- SixPoint™ ramp generator
- CoolStep™
- DcStep™
- StallGuard2™
- StealthChop™

Applications

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

Simplified Block Diagram

Read entire documentation.

TRINAMIC
MOTION CONTROL
Now part of Maxim Integrated

Contents

1 Preface

This document specifies objects and modes of operation of the Trinamic TMCM-1241 stepper motor control module with CANopen firmware. The CANopen firmware is designed to fulfill the CANopen DS402 and DS301 standards. This manual assumes that the reader is already familiar with the basics of the CANopen protocol, defined by the DS301 and DS402 standards of the CAN-CiA.

If necessary it is always possible to turn the module into a TMCL module by loading the TMCM-1241 TMCL fimware again through the USB interface, with the help of the firmware update function of the TMCL-IDE 3.0.

1.1 General Features of this CANopen Implementation

Main Characteristics

- Communication according to standard CiA-301 V4.1

- CAN bit rate: 20…1000kBit/s

- CAN ID: 11 bit

- Node ID: 1…127 (use vendor specific objects for changing the node ID)

- NMT services: NMT slave

SDO Communication

- 1 server

- Expedited transfer

- Segmented transfer

- No block transfer

PDO Communication

- Producer

- Consumer

- RPDOs

    – Axis 0: 1, 2, 3, 4
    – 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 USB interface of the module.

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

1.4.1 StallGuard2™

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

Figure 1: stallGuard2 Load Measurement as a Function of Load

1.4.2 CoolStep™

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

Figure 2: Energy Efficiency Example with CoolStep

1.5   SixPoint™ Motion Controller

TRINAMIC's SixPoint™ motion controller is a new type of ramp generator which offers faster machine operation compared to the classical linear acceleration ramps. The SixPoint™ ramp generator allows adapting the acceleration ramps to the torque curves of a stepper motor. It uses two different acceleration settings for the acceleration phase and also tow different deceleration settings for the deceleration phase. Start and stop speeds greater than zero can also be used.

Figure 3: Typical motion profile with TRINAMIC's SixPoint™ motion controller

A six point ramp begins using the start speed V_{START} (which can also be zero). Then, the acceleration value A1 will be used to accelerate the motor to the speed V1. When the speed V1 has been reached, the motor will be further accelerated using the acceleration value A2 until it has reached the speed V_{MAX}. The deceleration phase begins using the deceleration value D2. After reaching the speed V1 again the deceleration value D2 will be used to declerate to the stop speed V_{STOP} (which can also be zero).

The SixPoint™ ramp can be configured using the following objects:

Parameter Name	Object Index
Start velocity (V_{START})	2010_h
Acceleration A1	2011_h
Velocity V1	2012_h
Acceleration A2	6083_h
Maximum positioning velocity (V_{MAX})	6081_h
Deceleration D2	6084_h
Deceleration D1	2013_h
Stop velocity V_{STOP}	6082_h
Wait time WAIT	2015_h

Table 2: SixPoint Ramp Parameters

Setting the velocity V1 (object 2012_h) to zero switches off the SixPoint™ ramp. In this case, a trapezoidal ramp defined by the parameters V_{START}, A2, V_{MAX}, D2 and V_{STOP} will be used.

Note

The SixPoint™ ramp will only be used in profile positioning mode (pp mode). Profile velocity mode (pv mode) will always use a trapezoidal ramp, defined just by the acceleration (object 6083_h), the speed given using object $60FF_h$ and the start and stop speed (objects 2010_h and 6082_h). The deceleration parameters will not be used in pv mode.

---

2 Communication

2.1 Reference Model

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

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

The application and the application layer interact with service primitives.

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

Table 3: 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 4: Service Types

2.2   NMT State Machine

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

Starting and resetting the device is controlled via the state machine. The NMT state machine consists of the states shown in figure 4.

Figure 4: 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 5: Communication Architecture

2.3   Indicator LEDs

As defined by the CiA-303 standard part 3, one green and one red LED are used to indicate the NMT state and error state of the device.

2.3.1   Run LED

The green LED (also called CANopen run LED) shows the NMT state of the device:

- Single flash (200ms on, 1000ms off): The device is in NMT Stopped state.

- Blinking (200ms on, 200ms off): The device is in NMT Pre-operational state.

- On: The device is in NMT Operational state.

Other run LED blink codes are not used by the TMCM-1241 module.

2.3.2   Error LED

The red LED (also called CANopen error LED) shows possible error conditions:

- Off: No error. The device is in working condition.

- Single Flash (200ms on, 1000ms off): Warning limit reached. Too many error frames on the CAN bus. Possible reasons: master not yet initialized or wrong bit rate setting.

- Double Flash (200ms on, 200ms off, 200ms on, 1000ms off): Error control event. A node guard event or a heartbeat event (heartbeat consumer) has occured.

- On: Bus off. The CAN controller of the device is in bus off state.

Other error LED blink codes are not used by the TMCM-1241 module.

2.4 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 6: Device Model

2.5 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 5:

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 5: Object Dictionary

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

The manufacturer segment at indices 2000_h through $5FFF_h$ contains manufacturer specific objects. These objects control the special features of the Trinamic TMCM-1241 motion control device.

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

3 Communication Area

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

3.1 Detailed Object Specifications

3.1.1 Object 1000_h: Device Type

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

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

Table 6: Object Description (1000_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	no	UNSIGNED32	$FFFC0192_h$

Table 7: Entry Description (1000_h)

3.1.2 Object 1001_h: Error Register

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

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

Table 8: Object Description (1001_h)

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

Table 9: 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 10: Error Register Bits

3.1.3   Object 1005$_h$: COB-ID SYNC Message

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

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

Table 11: Value Definition (1005$_h$)

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

Table 12: Object Description (1005$_h$)

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

Table 13: Entry Description (1005$_h$)

3.1.4   Object 1008$_h$: Manufacturer Device Name

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

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

Table 14: Object Description (1008$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	no	—	TMCM-1241

Table 15: Entry Description (1008$_h$)

3.1.5   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 16: 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 17: Entry Description (1009$_h$)

3.1.6   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 18: 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 19: Entry Description (100A$_h$)

3.1.7   Object 100C$_h$: Guard Time

The objects at index 100C$_h$ and 100D$_h$ shall indicate the configured guard time respectively the life time factor. The life time factor multiplied with the guard time gives the life time for the life guarding protocol.

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

Table 20: Object Description (100C$_h$)

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

Table 21: Entry Description (100C$_h$)

3.1.8   Object 100D$_h$: Life Time Factor

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

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

Table 22: Object Description (100D$_h$)

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

Table 23: Entry Description (100D$_h$)

3.1.9   Object 1010$_h$: Store Parameters

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

The TMCM-1241 module supports saving of the following parameter groups:

- Sub-index 1_h: save all parameters.

- Sub-index 2_h: save communication parameters.

- Sub-index 4_h: save motor 0 parameters.

Note   In order to avoid storage of parameters by mistake, storage is only executed when a specific signature is written to the appropriate sub-Index. This signature is "save" (65766173_h, see also table 24).

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

Table 24: 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 25: Object Description (1010_h)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
00_h	Highest supported sub-index	ro	no	UNSIGNED8	4
01_h	Save all parameters	rw	no	UNSIGNED32	—
02_h	Save communication parameters	rw	no	UNSIGNED32	—
04_h	Save motor 0 parameters	rw	no	UNSIGNED32	—

Table 26: Entry Description (1010_h)

3.1.10   Object 1011$_h$: Restore Parameters

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

The TMCM-1241 module supports restoring of the following parameter groups:

- Sub-index 1$_h$: restore all parameters (factory reset).

- Sub-index 2$_h$; restore communication parameters.

- Sub-index 4$_h$: restore motor 0 parameters.

Note   In order to avoid restoring the parameters by mistake, restoring is only executed when a specific signature is written to the appropriate sub-Index. This signature is "load" (64616F6C$_h$, see also table 27).

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

Table 27: 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-1241.

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

Table 28: Object Description (1011$_h$)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
00_h	Highest supported sub-index	ro	no	UNSIGNED8	4
01_h	Restore all parameters	rw	no	UNSIGNED32	—
02_h	Restore communication parameters	rw	no	UNSIGNED32	—
04_h	Restore motor 0 parameters	rw	no	UNSIGNED32	—

Table 29: Entry Description (1011_h)

3.1.11   Object 1014_h: COB-ID Emergency Object

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

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

Table 30: Object Description (1014_h)

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

Table 31: Entry Description (1014_h)

3.1.12   Object 1015_h: Inhibit Time EMCY

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

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

Table 32: Object Description (1015_h)

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

Table 33: Entry Description (1015_h)

3.1.13 Object 1016$_h$: Consumer Heartbeat Time

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

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

Table 34: Value Definition (1016$_h$)

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

Table 35: 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 36: Entry Description (1016$_h$)

3.1.14 Object 1017$_h$: Producer Heartbeat Time

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

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

Table 37: Object Description (1017$_h$)

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

Table 38: Entry Description (1017_h)

3.1.15 Object 1018_h: Identity Object

The object 1018_h contains general information about the device:

- The vendor ID (sub-index 01_h) contains a unique value allocated to each manufacturer. The vendor ID of Trinamic is 286_h.

- The manufacturer specific product code (sub-index 2_h) identifies a specific device version.

- The manufacturer specific revision number (sub-index 3_h) consists of a major revision number and a minor revision number.

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

Table 39: Object Description (1018_h)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
00_h	Number of entries	ro	no	0...3	3
01_h	Vendor ID	ro	no	UNSIGNED32	0286_h
02_h	Product code	ro	no	UNSIGNED32	1241
03_h	Revision number	ro	no	UNSIGNED32	e.g. 20003_h for version 2.3

Table 40: Entry Description (1018_h)

3.1.16 Object 1023_h: OS Command

After switching to TMCL command mode using object 2700_h, this object can be used to send TMCL commands to the TMCM-1241 module. A TMCL command can be written to sub-index 1. After the command has been executed, the status code and the TMCL reply can be read from sub-index 2 and sub-index 3.

Object Description			
Index	Name	Object Type	Data Type
1023_h	OS Command	Record	Command

Table 41: Object Description (1023_h)

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

Table 42: Entry Description (1023$_h$)

3.1.17   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
00$_h$	Number of error classes	ro	no	—	2
01$_h$	Communication error	rw	no	UNSIGNED8	0 (enter stopped state)
02$_h$	Application error	rw	no	UNSIGNED8	1 (remain in current state)

Table 44: Entry Description (1029$_h$)

3.1.18   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 00$_h$ contains the number of valid entries within the communication record. Its value normally is 2, as this object consists of two other entries.

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

Sub-Index 02_h contains the transmission type of the RPDO. This can be FF_h or FE_h for event-driven, or 00_h for synchronous.

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
00_h	Largest sub-index supported	ro	2	2
01_h	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
02_h	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)

3.1.19 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 00_h contains the number of valid entries within the mapping record. This number of entries is also the number of the application variables which shall be received with the corresponding RPDO. The sub-indices from 01_h to the number of entries contain the information about the mapped application variables. These entries describe the PDO contents by their index, sub-index and length.

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
00_h	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
01_h	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
02_h	Mapping entry 2	rw	UNSIGNED32	Index 1600_h: 0 Index 1601_h: 60600008_h Index 1602_h: $607A0020_h$ Index 1603_h: $60FF0020_h$
03_h	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 3.1.18). Then, set its number of mapped PDO entries to zero (sub-index 0 of the appropriate PDO mapping object). Now, the mapppings themself can be changed. After that, set the number of map objects to the desired value, and finally activate the PDO by clearing bit 31 of its COB-ID.

3.1.20   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 00_h contains the number of valid entries within the communication record. Its value normally is 5, as this object consists of five other entries.

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

definition, set this bit to inactivate the PDO.

Sub-index 02_h contains the transmission type of the RPDO. This can be FF_h or FE_h for event-driven, or 00_h or 01_h for synchronous.

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

Sub-index 04_h is not used.

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

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
00_h	Largest sub-index supported	ro	5	5
01_h	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
02_h	Transmission type	rw	UNSIGNED8	Index 1800_h: FF_h Index 1801_h: FF_h Index 1802_h: 01_h Index 1803_h: 01_h
03_h	Inhibit time	rw	UNSIGNED16	0
04_h	Compatibility entry	ro	UNSIGNED8	0
05_h	Event timer	rw	UNSIGNED16	0

Table 50: Entry Description (1800_h)

3.1.21 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 00$_h$ contains the number of valid entries within the mapping record. This number of entries is also the number of the application variables which shall be transmitted with the corresponding TPDO. The sub-indices from 01$_h$ to the number of entries contain the information about the mapped application variables. These entries describe the PDO contents by their index, sub-index and length.

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
00$_h$	Number of mapped aaplication objects in PDO	rw	0…3	Index 1A00$_h$: 1 Index 1A01$_h$: 2 Index 1A02$_h$: 2 Index 1A03$_h$: 2
01$_h$	Mapping entry 1	rw	UNSIGNED32	Index 1A00$_h$: 60410010$_h$ Index 1A01$_h$: 60410010$_h$ Index 1A02$_h$: 60410010$_h$ Index 1A03$_h$: 60410010$_h$
02$_h$	Mapping entry 2	rw	UNSIGNED32	Index 1A00$_h$: 0 Index 1A01$_h$: 60610008$_h$ Index 1A02$_h$: 60640020$_h$ Index 1A03$_h$: 606C0020$_h$
03$_h$	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 3.1.20). Then, set its number of mapped PDO entries to zero (sub-index 0 of the appropriate PDO mapping object). Now, the mapppings themself can be changed. After that, set the number of map objects to the desired value, and finally activate the PDO by clearing bit 31 of its COB-ID.

4   Manufacturer specific Area

The manufacturer segment contains manufacturer specific objects. These objects control the special features of the Trinamic Motion Control device TMCM-1241.

4.1   Objects related to CoolStep™

Figure 7 shows an overview of the CoolStep™ related objects for motor #0. Please bear in mind that the figure only shows one example for a drive. There are objects which concern the configuration of the current. Other objects are for velocity regulation and for time adjustment. The CoolStep™ feature is sometimes also called SmartEnergy.

The following adjustments have to be made:

- Thresholds for current and velocity have to be identified and set.

- The StallGuard2™ feature has to be adjusted and enabled.

- The reduction or increasing of the current in the CoolStep™ area (depending on the load) has to be configured.

Figure 7: CoolStep Adjustment Points and Thresholds

CoolStep Adjustment Objects		
Object	Name	Description
2003_h	Absolute maximum current	The maximum value is 255. This value means 100% of the maximum current of the module. The current adjustment is within the range 0…255 and can be adjusted in 32 steps (0…255 divided by eight; step 0 = 0…7, step 1 = 8…15 and so on). The most important motor setting, since too high values might cause motor damage!
2004_h	Standby current	The current limit two seconds after the motor has stopped.
2098_h	SmartEnergy current minimum	Sets the lower motor current limit for CoolStep operation by scaling the run current (object 2003_h) value. This can be: 0: for 1/2 of the run current 1: for 1/4 of the run current
2099_h	SmartEnergy current down step	Sets the speed of current decrement when the StallGuard2 reading is above the upper threshold. 0: slow decrement 3: fast decrement
$209B_h$	SmartEnergy current up step	Sets the current increment step when the StallGuard2 below the lower threshold. 0: slow increment 3: fast increment / fast reaction to rising load
$209A_h$	SmartEnergy hysteresis	Sets the distance between the lower and the upper threshold for StallGuard2 reading. Above the upper threshold the motor current becomes decreased.
$20A4_h$	Stop on stall	Below this speed the motor will not be stopped. Above this speed the motor will stop in case StallGuard2 load value reaches zero.
$20A5_h$	SmartEnergy threshold speed.	Above this speed CoolStep becomes enabled.
2089_h	Standby delay	Standstill period before the current is changed down to standby current. The standard value is 200 which is 2 seconds.

Table 53: CoolStep related Objects

4.2 Detailed Object Specifications

4.2.1 Object 2000$_h$: Microstep Resolution

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

Object Description			
Index	Name	Object Type	Data Type
2000$_h$	Microstep Resolution	Variable	UNSIGNED8

Table 54: Object Description (2000$_h$)

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

Table 55: Entry Description (2000$_h$)

4.2.2 Object 2001$_h$: Fullstep Resolution

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

Object Description			
Index	Name	Object Type	Data Type
2001$_h$	Motor full step resolution	Variable	UNSIGNED16

Table 56: Object Description (2001$_h$)

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

Table 57: Entry Description (2001$_h$)

4.2.3 Object 2002$_h$: Brake Delay Times

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

Figure 8: Brake Output Timing

Object Description			
Index	Name	Object Type	Data Type
2002$_h$	Brake delay times	Array	UNSIGNED16

Table 58: Object Description (2002$_h$)

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

Table 59: Entry Description (2002$_h$)

4.2.4 Object 2003$_h$: Maximum Current

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

Object Description			
Index	Name	Object Type	Data Type
2003$_h$	Maximum current	Variable	UNSIGNED8

Table 60: Object Description (2003$_h$)

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

Table 61: Entry Description (2003$_h$)

4.2.5 Object 2004$_h$: Standby Current

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

Object Description			
Index	Name	Object Type	Data Type
2004$_h$	Maximum current	Variable	UNSIGNED8

Table 62: Object Description (2004$_h$)

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

Table 63: Entry Description (2004$_h$)

4.2.6 Object 2005$_h$: Limit Switches

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

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

- Bit 2 inverts the left limit switch

- Bit 3 inverts the right limit switch

The polarity of the home switch can be set using bit 5. Furthermore, the functionality of the shutdown input can be controlled using bit 6. If this bit is set then the CiA-402 state machine also reacts on the shutdown input (it will change to FAULT state if a shutdown is detected).

Object Description			
Index	Name	Object Type	Data Type
2005$_h$	Limit switches	Variable	UNSIGNED32

Table 64: Object Description (2005$_h$)

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

Table 65: Entry Description (2005$_h$)

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

Table 66: Bit Definitions (2005$_h$)

4.2.7 Object 200A$_h$: Enable Drive Delay Time

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

Object Description			
Index	Name	Object Type	Data Type
200A$_h$	Enable drive delay time	Variable	UNSIGNED16

Table 67: Object Description (200A$_h$)

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

Table 68: Entry Description (200A)$_h$

4.2.8 Object 200B$_h$: Encoder Parameters

This object defines encoder parameters. These are the direction of rotation for external encoders (set to 1 if the direction is reversed compared to the motor) and if the position is to be intialized with the encoder position. The object is only writable in SWITCHED_ON_DISABLED state.

Object Description			
Index	Name	Object Type	Data Type
200B$_h$	Encoder parameters	Array	UNSIGNED8

Table 69: Object Description (200B$_h$)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
1	Not used	rw	no	—	—
2	Direction of rotation (external encoders only)	rw	no	0/1	0
3	Initialize position	rw	no	0/1	1

Table 70: Entry Description (200B$_h$)

4.2.9   Object 200C$_h$: Brake Current Feed

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

Object Description			
Index	Name	Object Type	Data Type
200C$_h$	Brake current feed	Array	UNSIGNED8

Table 71: Object Description (200C$_h$)

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

Table 72: Entry Description (200C$_h$)

4.2.10   Object 2010$_h$: Profile Start Velocity

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

Object Description			
Index	Name	Object Type	Data Type
2010$_h$	Profile Start Velocity	Variable	UNSIGNED32

Table 73: Object Description (2010$_h$)

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

Table 74: Entry Description (2010$_h$)

4.2.11   Object 2011$_h$: Profile A1

This object contains the acceleration value used for ramping up from the start velocity (object 2011$_h$, see section 4.2.10) to the velocity V1 (object ($_h$2012), see section 4.2.12).

Object Description			
Index	Name	Object Type	Data Type
2011$_h$	Profile A1	Variable	UNSIGNED32

Table 75: Object Description (2011$_h$)

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

Table 76: Entry Description (2011$_h$)

4.2.12   Object 2012$_h$: Profile V1

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

Object Description			
Index	Name	Object Type	Data Type
2012$_h$	Profile V1	Variable	UNSIGNED32

Table 77: Object Description (2012$_h$)

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

Table 78: Entry Description (2012$_h$)

4.2.13   Object 2013$_h$: Profile D1

This object contains the deceleration value used for decelerating from the maximum positioning velocity to the velocity V1 (object 2012$_h$, see section 4.2.12).

Object Description			
Index	Name	Object Type	Data Type
2013$_h$	Profile D1	Variable	UNSIGNED32

Table 79: Object Description (2013$_h$)

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

Table 80: Entry Description (2013$_h$)

4.2.14   Object 2015$_h$: Ramp Wait Time

This object defines the waiting time after ramping down to zero velocity before the next movement or direction inversion can start. Time range is 0 to 2 seconds. This setting avoids excess acceleration e.g. from positive stop velocity to negative start velocity.

Object Description			
Index	Name	Object Type	Data Type
2015$_h$	Ramp Wait Time	Variable	UNSIGNED16

Table 81: Object Description (2015$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0…65535 [0.000032s]	0

Table 82: Entry Description (2015$_h$)

4.2.15   Object 2089$_h$: Setting Delay

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

Unit: 10msec

Object Description			
Index	Name	Object Type	Data Type
2089$_h$	Setting Delay	Variable	UNSIGNED16

Table 83: Object Description (2089$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0…400	0

Table 84: Entry Description (2089$_h$)

4.2.16   Object 208C$_h$: Velocity Dimension Index

With this object different units can be chosen:

  • Writing 0 selects internal units.

  • Writing 181 sets PPS for velocity and PPS/s for acceleration.

This can only be changed in SWITCHED_ON_DISABLED mode.

Object Description			
Index	Name	Object Type	Data Type
208C$_h$	Velocity Dimension Index	Variable	UNSIGNED8

Table 85: Object Description (208C$_h$)

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

Table 86: Entry Description (208C$_h$)

4.2.17   Object 208E$_h$: Acceleration Dimension Index

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

Object Description			
Index	Name	Object Type	Data Type
208E$_h$	Acceleration Dimension Index	Variable	UNSIGNED8

Table 87: Object Description (208E$_h$)

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

Table 88: Entry Description (208E$_h$)

4.2.18   Object 2092$_h$: Chopper Blank Time

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

Object Description			
Index	Name	Object Type	Data Type
2092$_h$	Chopper Blank Time	Variable	UNSIGNED8

Table 89: Object Description (2092$_h$)

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

Table 90: Entry Description (2092$_h$)

4.2.19   Object 2093$_h$: Chopper Mode

Select the chopper mode using this object:

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

Object Description			
Index	Name	Object Type	Data Type
2093$_h$	Chopper Mode	Variable	UNSIGNED8

Table 91: Object Description (2093$_h$)

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

Table 92: Entry Description (2093$_h$)

4.2.20   Object 2094$_h$: Chopper Hysteresis Decrement

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

- 0 – fast decrement

- 3 – very slow decrement

Object Description			
Index	Name	Object Type	Data Type
2094$_h$	Chopper Hysteresis Decrement	Variable	UNSIGNED8

Table 93: Object Description (2094$_h$)

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

Table 94: Entry Description (2094$_h$)

4.2.21   Object 2095$_h$: Chopper Hysteresis End

This object provides the setting of the hysteresis end value after a number of decrements. The decrement interval time is controlled by object 2094$_h$ (section 4.2.20).
Possible values are:

- -3…-1 – negative hysteresis end setting

- 0 – zero hysteresis end setting

- 1…12 – positive hysteresis end setting

Object Description			
Index	Name	Object Type	Data Type
2095$_h$	Chopper Hysteresis End	Variable	SIGNED8

Table 95: Object Description (2095$_h$)

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

Table 96: Entry Description (2095$_h$)

4.2.22   Object 2096$_h$: Chopper Hysteresis Start

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

Object Description			
Index	Name	Object Type	Data Type
2096$_h$	Chopper Hysteresis Start	Variable	UNSIGNED8

Table 97: Object Description (2096$_h$)

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

Table 98: Entry Description (2096$_h$)

4.2.23   Object 2097$_h$: Chopper Off Time

The off time setting controls the minimum chopper frequency. Under normal circumstances, an off time within the range of 5μs to 20μs is used. Off time setting for constant t$_{OFF}$ chopper: $N_{CLK} = 12 + 32 * t_{OFF}$. Minimum is 64 clocks.
Setting this parameter to zero completely disables all driver transistors and so lets the motor free-wheel.

Object Description			
Index	Name	Object Type	Data Type
2097$_h$	Chopper Off Time	Variable	UNSIGNED8

Table 99: Object Description (2097$_h$)

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

Table 100: Entry Description (2097$_h$)

4.2.24   Object 2098$_h$: Smart Energy Current Minimum

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

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

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

Object Description			
Index	Name	Object Type	Data Type
2098$_h$	Smart Energy Current Minimum	Variable	UNSIGNED8

Table 101: Object Description (2098$_h$)

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

Table 102: Entry Description (2098$_h$)

4.2.25   Object 2099$_h$: Smart Energy Current Down Step

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

- 0 – 32 measurements – slowest decrement

- 1 – 8 measurements

- 2 – 2 measurements

- 3 – 1 measurements – fastest decrement

Object Description			
Index	Name	Object Type	Data Type
2099$_h$	Smart Energy Current Down Step	Variable	UNSIGNED8

Table 103: Object Description (2099$_h$)

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

Table 104: Entry Description (2099$_h$)

4.2.26   Object 209A$_h$: Smart Energy Hysteresis

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

Hysteresis: $(\text{SmartEnergy hysteresis value} + 1) * 32$
Upper StallGuard2 threshold: $(\text{SmartEnergy hysteresis start} + \text{SmartEnergy hysteresis} + 1) * 32$

Object Description			
Index	Name	Object Type	Data Type
209A$_h$	Smart Energy Hysteresis	Variable	UNSIGNED8

Table 105: Object Description (209A$_h$)

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

Table 106: Entry Description (209A$_h$)

4.2.27 Object 209B$_h$: Smart Energy Current Up Step

This object sets the current increment step. The current becomes incremented for each measured StallGuard2 value below the lower threshold (see smart energy hysteresis start (object 209C$_h$, section 4.2.28). Possible values are:

- 0 – 1 step – slowest increment
- 1 – 2 steps
- 2 – 4 steps
- 3 – 8 steps – fastest increment

Object Description			
Index	Name	Object Type	Data Type
209B$_h$	Smart Energy Current Up Step	Variable	UNSIGNED8

Table 107: Object Description (209B$_h$)

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

Table 108: Entry Description (209B$_h$)

4.2.28 Object 209C$_h$: Smart Energy Hysteresis Start

This object serves to set the lower threshold for the StallGuard2 value (see smart Energy current up step (section 4.2.27). Setting this to 0 disables the CoolStep™ function.

Object Description			
Index	Name	Object Type	Data Type
209C$_h$	Smart Energy Hysteresis Start	Variable	UNSIGNED8

Table 109: Object Description (209C$_h$)

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

Table 110: Entry Description (209C$_h$)

4.2.29   Object 209D$_h$: Smart Energy Filter Enable

This object is used to set the StallGuard2 filter for more precision of the measurement. It reduces the measurement frequency to one measurement per four fullsteps if set.
In most cases it is expedient to set the filtered mode when using CoolStep™. Use the standard mode for step loss detection.
Possible values are:

- 0 – standard mode

- 1 – filtered mode

Object Description			
Index	Name	Object Type	Data Type
209D$_h$	Smart Energy Filter Enable	Variable	UNSIGNED8

Table 111: Object Description (209D$_h$)

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

Table 112: Entry Description (209D$_h$)

4.2.30   Object 209E$_h$: StallGuard2 Threshold

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

Object Description			
Index	Name	Object Type	Data Type
209E$_h$	StallGuard2 Threshold	Variable	SIGNED8

Table 113: Object Description (209E$_h$)

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

Table 114: Entry Description (209E$_h$)

4.2.31   Object 20A1$_h$: Short Protection Disable

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

- 0 – Short to GND protection enabled

- 1 – Short to GND protection disabled

Object Description			
Index	Name	Object Type	Data Type
20A1$_h$	Short Protection Disable	Variable	UNSIGNED8

Table 115: Object Description (20A1$_h$)

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

Table 116: Entry Description (20A1$_h$)

4.2.32   Object 20A4$_h$: Stop on Stall

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

Object Description			
Index	Name	Object Type	Data Type
20A4$_h$	Stop on Stall	Variable	UNSIGNED32

Table 117: Object Description (20A4$_h$)

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

Table 118: Entry Description (20A4$_h$)

4.2.33   Object 20A5$_h$: Smart Energy Threshold Speed

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

Object Description			
Index	Name	Object Type	Data Type
20A5$_h$	Smart Energy Threshold Speed	Variable	UNSIGNED32

Table 119: Object Description (20A5$_h$)

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

Table 120: Entry Description (20A5$_h$)

4.2.34   Object 20A7$_h$: Stop on Stall Option Code

This object controls what to do when the motor is stopped by the stop on stall function. Possible options are:

• 0: enter FAULT state on stall

• 1: just stop motor on stall, stay in OPERATIONAL state

Object Description			
Index	Name	Object Type	Data Type
20A7$_h$	Stop on Stall Option Code	Variable	UNSIGNED8

Table 121: Object Description (20A7$_h$)

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

Table 122: Entry Description (20A7$_h$)

4.2.35 Object 20B0$_h$: PWM Threshold Speed

The StealthChop feature will be switched on when the value of this object is greater than zero and the actual velocity is lower than the value set by this object.

Object Description			
Index	Name	Object Type	Data Type
20B0$_h$	PWM Threshold Speed	Variable	UNSIGNED32

Table 123: Object Description (20B0$_h$)

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

Table 124: Entry Description (20B0$_h$)

4.2.36 Object 20B1$_h$: PWM Gradient

Velocity dependent gradient for the PWM amplitude (StealthChop). Setting this value to 0 turns off Stealth-Chop.

Object Description			
Index	Name	Object Type	Data Type
20B1$_h$	PWM Gradient	Variable	UNSIGNED8

Table 125: Object Description (20B1$_h$)

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

Table 126: Entry Description (20B1$_h$)

4.2.37 Object 20B2$_h$: PWM Amplitude

Maximum PWM amplitude when switching to StealthChop mode. Do not set too low. Values above 64 are recommended.

Object Description			
Index	Name	Object Type	Data Type
20B2$_h$	PWM Amplitude	Variable	UNSIGNED8

Table 127: Object Description (20B2$_h$)

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

Table 128: Entry Description (20B2$_h$)

4.2.38   Object 20B3$_h$: DcStep Minimum Speed

Minimum speed for switching to DcStep. The motor driver will not switch to DcStep mode when the actual velocity is below this value. Setting this object to zero completely switches off DcStep.

Object Description			
Index	Name	Object Type	Data Type
20B3$_h$	DcStep Minimum Speed	Variable	UNSIGNED32

Table 129: Object Description (20B3$_h$)

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

Table 130: Entry Description (20B3$_h$)

4.2.39   Object 20B4$_h$: DcStep Time

This setting controls the reference pulse width for DcStep load measurement. It must be optimized for robust operation with maximum motor torque. A higher value allows higher torque and higher velocity, a lower value allows operation down to a lower velocity as set by the minimum DcStep speed (object 20B3$_h$, see section 4.2.38).

Object Description			
Index	Name	Object Type	Data Type
20B4$_h$	DcStep Time	Variable	UNSIGNED16

Table 131: Object Description (20B4$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0…1023	0

Table 132: Entry Description (20B4$_h$)

4.2.40   Object 20B5$_h$: DcStep StallGuard

This setting controls stall detection in DcStep mode. Increase this value for higher sensitivity.

Object Description			
Index	Name	Object Type	Data Type
20B5$_h$	DcStep StallGuard	Variable	UNSIGNED8

Table 133: Object Description (20B5$_h$)

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

Table 134: Entry Description (20B5$_h$)

4.2.41   Object 20B6$_h$: Fullstep Threshold Speed

Depending on the settings of objects 20B7$_h$ and 20B8$_h$ (see sections 4.2.42 and 4.2.43) the driver switches to fullstep mode and/or to a different chopper mode when the speed set by this object is exceeded.

Object Description			
Index	Name	Object Type	Data Type
20B6$_h$	Fullstep Threshold Speed	Variable	UNSIGNED32

Table 135: Object Description (20B6$_h$)

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

Table 136: Entry Description (20B6$_h$)

4.2.42   Object 20B7$_h$: High Speed Chopper Mode

The motor driver will switch to a different chopper mode when this object is set to 1 and the measured speed is greater than the threshold speed set by object 20B6$_h$ (see section 4.2.41).

Object Description			
Index	Name	Object Type	Data Type
20B7$_h$	High Speed Chopper Mode	Variable	UNSIGNED8

Table 137: Object Description (20B7$_h$)

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

Table 138: Entry Description (20B7$_h$)

4.2.43 Object 20B8$_h$: High Speed Fullstep Mode

The motor driver will switch to fullstep mode when this object is set to 1 and the measured speed is greater than the threshold speed set by object 20B6$_h$ (see section 4.2.41).

Object Description			
Index	Name	Object Type	Data Type
20B8$_h$	High Speed Chopper Mode	Variable	UNSIGNED8

Table 139: Object Description (20B8$_h$)

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

Table 140: Entry Description (20B8$_h$)

4.2.44 Object 20B9$_h$: Power Down Ramp

The value set by this object controls the number of clock cycles for motor power down after a motion as soon as the motor has stopped and the setting time (set by object 2089$_h$, please see section 4.2.15) has expired. The smooth transition avoids a motor jerk upon power down.

- 0=instant power down.

- 15=longest possible power down ramp.

Object Description			
Index	Name	Object Type	Data Type
20B9$_h$	Power Down Ramp	Variable	UNSIGNED8

Table 141: Object Description (20B9$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0…15	7

Table 142: Entry Description (20B9$_h$)

4.2.45 Object 2100$_h$: Home Offset Display

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

Object Description			
Index	Name	Object Type	Data Type
2100$_h$	Home Offset Display	Variable	SIGNED32

Table 143: Object Description (2100$_h$)

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

Table 144: Entry Description (2100$_h$)

4.2.46 Object 2101$_h$: Actual Load Value

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

Object Description			
Index	Name	Object Type	Data Type
2101$_h$	Actual Load Value	Variable	UNSIGNED16

Table 145: Object Description (2101$_h$)

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

Table 146: Entry Description (2101$_h$)

4.2.47 Object 2102$_h$: Driver Error Flags

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

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

Table 147: Driver Error Flags (2102_h)

Object Description			
Index	Name	Object Type	Data Type
2102_h	Driver Error Flags	Variable	UNSIGNED8

Table 148: Object Description (2102_h)

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

Table 149: Entry Description (2102_h)

4.2.48 Object 2107_h: Microstep Resolution Display

This object shows the microstep resolution, set by object 2000_h (please see section 4.2.1).

Object Description			
Index	Name	Object Type	Data Type
2107_h	Microstep resolution display	Variable	UNSIGNED8

Table 150: Object Description (2107_h)

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

Table 151: Entry Description (2107$_h$)

4.2.49 Object 2108$_h$: Absolute Resolver Value

This object shows the value of the (single turn) absolute position resolver.

Object Description			
Index	Name	Object Type	Data Type
2108$_h$	Microstep resolution display	Variable	UNSIGNED32

Table 152: Object Description (2108$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	0…4095	—

Table 153: Entry Description (2108$_h$)

4.2.50 Object 210B$_h$: Step Counter

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

Object Description			
Index	Name	Object Type	Data Type
210B$_h$	Step Counter	Variable	RECORD

Table 154: Object Description (210B$_h$)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
1	Lower 32 Bits	ro	no	0…ffffffff$_h$	0
2	Higher 32 Bits	ro	no	0…ffffffff$_h$	0
3	64 Bit Value	ro	no	0…ffffffffffffffff$_h$	0

Table 155: Entry Description (210B$_h$)

4.2.51 Object 2121$_h$: PWM Scale Value

Actual PWM scale value used when the motor driver is operaing in StealthChop mode.

Object Description			
Index	Name	Object Type	Data Type
2121$_h$	PWM Scale Value	Variable	UNSIGNED8

Table 156: Object Description (2121$_h$)

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

Table 157: Entry Description (2121$_h$)

4.2.52 Object 2122$_h$: Measured Velocity

This object contains the velocity measured by the motor driver. This value is important only when the motor driver is operating in DcStep mode.

Object Description			
Index	Name	Object Type	Data Type
2122$_h$	Measured Velocity	Variable	SIGNED32

Table 158: Object Description (2122$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	no	-16777215…16777215	—

Table 159: Entry Description (2122$_h$)

4.2.53 Object 2700$_h$: TMCL Direct Communication

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

Object Description			
Index	Name	Object Type	Data Type
2700$_h$	TMCL Direct Communication	Variable	UNSIGNED32

Table 160: Object Description (2700$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	656b616d$_h$	0

Table 161: Entry Description (2700)$_h$

4.2.54 Object 2701$_h$: Manufacturer Specific Mode

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

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

read:	0	=	manufacturer specific mode is inactive
	1	=	manufacturer specific mode is active
write:	656b616d$_h$	=	make signature
	6c6c696b$_h$	=	kill signature

Object Description			
Index	Name	Object Type	Data Type
2701$_h$	Manufacturer Specific Mode	Variable	UNSIGNED32

Table 162: Object Description (2701$_h$)

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

Table 163: Entry Description (2701)$_h$

4.2.55 Object 2702$_h$: Device Digital Inputs

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

Bit Definitions	
Bit	Description
16	IN0
17	IN1
18	IN2

Table 164: Bit Definitions (2702_h)

Object Description			
Index	Name	Object Type	Data Type
2702_h	Device Digital Inputs	Variable	UNSIGNED32

Table 165: Object Description (2702_h)

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

Table 166: Entry Description ($2702)_h$

4.2.56   Object 2703_h: Device Digital Outputs

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

Bit Definitions	
Bit	Description
16	OUT0

Table 167: Bit Definitions (2703_h)

Object Description			
Index	Name	Object Type	Data Type
2703_h	Device Digital Outputs	Variable	ARRAY

Table 168: 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 169: Entry Description (2703$_h$)

Note   Output #0 can also be used for automatically controlling a brake that can be connected to the module via this output. In order to be able to control output #0 via this object the autmatic brake control function has to be disabled. Do this by writing 0 to sub-index 1 and sub-index 2 of object 200C$_h$.

4.2.57 Object 2704$_h$: CAN Bit Rate

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

Object Description			
Index	Name	Object Type	Data Type
2704$_h$	CAN Bit Rate	Variable	UNSIGNED16

Table 170: Object Description (2704$_h$)

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

Table 171: Entry Description (2704)$_h$

How to change the bit rate of a module:

- Write new bit rate in object 2704$_h$.

- Write the save signature 65766173$_h$ to sub-index 1 of object 2706$_h$.

- Reset module.

4.2.58 Object 2705$_h$: Node ID

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

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

Table 172: Object Description (2705$_h$)

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

Table 173: Entry Description (2705)$_h$

How to change the node ID of a module:

- Write new node ID to object 2705$_h$.

- Write save signature 65766173$_h$ to sub-index 1 of object 2706$_h$.

- Reset the module.

4.2.59   Object 2706$_h$: Store

Writing the save signature to this object permanently saves changes made to objects 2704$_h$ and 2705$_h$. The save signature is 65766173$_h$.

Object Description			
Index	Name	Object Type	Data Type
2706$_h$	Store	Variable	UNSIGNED32

Table 174: Object Description (2706$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	save signature: 65766173$_h$	1

Table 175: Entry Description (2706)$_h$

4.2.60   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 176: Object Description (2707_h)

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

Table 177: Entry Description (2707)$_h$

4.2.61 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 178: Object Description (2708_h)

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

Table 179: Entry Description (2708_h)

4.2.62 Object $270E_h$: Device Analog Inputs

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

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

Table 180: Object Description ($270E_h$)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
1	Analog input 0	ro	yes	0…4095	–

Table 181: Entry Description (270E$_h$)

4.2.63   Object 2710$_h$: Pull-up Resistors

This object enables or disables the built-in pull-up resistors for the general purpose inputs. Bit 0 is used to switch the pull-up resistor for the IN0/HOME input, and bit 1 is used to switch the pull-up resistor for the inputs IN1 and IN2.

Value Definition	
Value	Description
0	All pull-up resistors off.
1	Pull-up resistor on for IN0/HOME. Pull-up resistors off for IN1/IN2.
2	Pull-up resistor off for IN0/HOME. Pull-up resistors on for IN1/IN2.
3	All pull-up resistors on.

Table 182: Value Description (2710$_h$)

Object Description			
Index	Name	Object Type	Data Type
2710$_h$	Pull-up resistors	Variable	UNSIGNED8

Table 183: Object Description (2710$_h$)

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

Table 184: Entry Description (2710$_h$)

5   Profile specific Area

The profile segment contains CiA-402 standard motion control objects. These objects control the motion control functions of the TMCM-1241. Since it is not possible to operate the modes in parallel, the user is able to activate the required function by selecting a mode of operation. The control device writes to the modes of operation object in order to select the operation mode. The drive device provides the modes of operation display object to indicate the actual activated operation mode. Controlword, statusword, and set-points are used mode-specific. This implies the responsibility of the control device to avoid inconsistencies and erroneous behavior.

The following operating modes (selectable via object 6060_h, please see 5.1.6) are implemented on the TMCM-1241:

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

5.1   Detailed Object Specifications

5.1.1   Object $605A_h$: Quick Stop Option Code

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

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

Table 185: Value Description ($605A_h$)

Object Description			
Index	Name	Object Type	Data Type
$605A_h$	Quick stop option code	Variable	SIGNED16

Table 186: Object Description ($605A_h$)

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

Table 187: Entry Description ($605A_h$)

5.1.2   Object 605B$_h$: Shutdown Option Code

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

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

Table 188: Value Description (605B$_h$)

Object Description			
Index	Name	Object Type	Data Type
605B$_h$	Shutdown option code	Variable	UNSIGNED16

Table 189: Object Description (605B$_h$)

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

Table 190: Entry Description (605B$_h$)

5.1.3   Object 605C$_h$: Disable Operation Option Code

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

Value Definition	
Value	Definition
1	Slow down on slow down ramp

Table 191: Value Description (605C$_h$)

Object Description			
Index	Name	Object Type	Data Type
605C$_h$	Disable operation option code	Variable	UNSIGNED16

Table 192: Object Description (605C$_h$)

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

Table 193: Entry Description (605C$_h$)

5.1.4 Object 605D$_h$: Halt Option Code

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

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

Table 194: Value Description (605D$_h$)

Object Description			
Index	Name	Object Type	Data Type
605D$_h$	Halt option code	Variable	UNSIGNED16

Table 195: Object Description (605D$_h$)

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

Table 196: Entry Description (605D$_h$)

5.1.5 Object 605E$_h$: Fault Reaction Option Code

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

Value Definition	
Value	Definition
2	Slow down on quick stop ramp

Table 197: Value Description (605E$_h$)

Object Description			
Index	Name	Object Type	Data Type
605E$_h$	Fault reaction option code	Variable	UNSIGNED16

Table 198: Object Description (605E$_h$)

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

Table 199: Entry Description (605E$_h$)

5.1.6 Object 6060$_h$: Modes of Operation

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

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

Table 200: 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 201: Object Description (6060$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	refer to CiA-402	see table 200	0

Table 202: Entry Description (6060$_h$)

5.1.7   Object 6061$_h$: Modes of Operation Display

This object shows the operating mode that is currently set.

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

Table 203: 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 204: Object Description (6061$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	refer to CiA-402	see table 203	0

Table 205: Entry Description (6061$_h$)

5.1.8   Object 606A$_h$: Sensor Selection Code

This object provides the source of the position sensor actual value. It selects if an encoder is to be used for position monitoring and if yes, which encoder will be used. This can either be no encoder, the internal magnetic encoder, or an externally connected incremental encoder.

Value Definition	
Value	Mode
0	Internal magnetic encoder
-1	No encoder
-2	External incremental encoder

Table 206: Value Description (606A$_h$)

Object Description			
Index	Name	Object Type	Data Type
606A$_h$	Sensor selection code	Variable	SIGNED16

Table 207: Object Description (606A$_h$)

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

Table 208: Entry Description (606A$_h$)

5.1.9   Object 608F$_h$: Position Encoder Resolution

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

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

All values are dimensionless.

This object is only valid for an externally connected inremental encoder. The resolution of the internal encoder is fixed.

Object Description			
Index	Name	Object Type	Data Type
608F$_h$	Position Encoder Resolution	Array	UNSIGNED32

Table 209: Object Description (608F$_h$)

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

Table 210: Entry Description (608F$_h$)

5.1.10   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 211: Object Description (60FD$_h$)

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

Table 212: Entry Description (60FD$_h$)

5.1.11   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 213: Value Definition (6502$_h$)

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

Table 214: Object Description (6502$_h$)

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

Table 215: Entry Description (6502_h)

6 Profile Position Mode

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

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

6.1 Detailed Object Specifications

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

Figure 9: DS402 Finite State Machine

Notes on state transitions:

- Commands directing a change in state are processed completely and the new state achieved before additional state change commands are processed.

- Transitions 0 and 1 occur automatically at drive power-on or reset. Transition 14 occurs automatically, too. All other state changes must be directed by the host.

- Drive function disabled indicates that no current is being supplied to the motor.

- Drive function enabled indicates that current is available for the motor and profile position and profile velocity reference values may be processed.

6.1.1 Object 6040$_h$: Control Word

This object indicates the received command controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to figure 9 for detailed information.

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

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

Table 216: Structure of the Control Word in pp Mode

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

Table 217: Operation Mode specific Bits in pp Mode

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

Table 218: Command Coding

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

Table 219: Object Description (6040_h in pp Mode)

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

Table 220: Entry Description (6040_h in pp Mode)

6.1.2   Object 6041_h: Status Word

This object provides the status of the PDS FSA. It reflects the status of the CiA-402 state machine. Please refer to figure 9 for detailed information. The object is structured as defined below.
For more information about the coding please refer to the CANopen Drives and motion control device profile, part 2.

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

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

Table 221: Structure of the Staus Word in pp Mode

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

Table 222: 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 223: Operation Mode specific Bits in pp Mode

State Coding	
Status word	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 224: State Coding

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

Table 225: Object Description (6041$_h$ in pp Mode)

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

Table 226: Entry Description (6041$_h$ in pp Mode)

6.1.3 Object 6062$_h$: Position Demand Value

This object provides the demanded position value. The value is given in microsteps. 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
6062h	Position Demand Value	Variable	SIGNED32

Table 227: Object Description (6062h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	Refer to CiA402-3	SIGNED32	no

Table 228: Entry Description (6062h)

6.1.4 Object 6063h: Position Actual Internal Value

This object provides the actual value of the encoder or the motor. Please use the sensor selection object 606Ah (see section 5.1.8) for selecting the motor or the encoder first. Object 6063h indicates the actual position of the encoder or the motor, re-scaled to the microstep resolution. The value is given in microsteps.

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

Table 229: Object Description (6063h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	Refer to CiA402-3	SIGNED32	no

Table 230: Entry Description (6063h)

6.1.5 Object 6064h: Position Actual Value

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

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

Table 231: Object Description (6064h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	Refer to CiA402-3	SIGNED32	no

Table 232: Entry Description (6064$_h$)

6.1.6 Object 6065$_h$: Following Error Window

This object indicates the configured range of tolerated position values symmetrically to the position demand value. If the position actual value is out of the following error window, a following error occurs. A following error may occur when a drive is blocked, unreachable profile velocity occurs, or at wrong closed-loop coefficients. The value shall be given in microsteps.

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

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

Object Description			
Index	Name	Object Type	Data Type
6065$_h$	Following Error Window	Variable	UNSIGNED32

Table 233: Object Description (6065$_h$)

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

Table 234: Entry Description (6065$_h$)

6.1.7 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 235: Object Description (6067$_h$)

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

Table 236: Entry Description (6067$_h$)

6.1.8 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 237: Object Description (6068$_h$)

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

Table 238: Entry Description (6068$_h$)

6.1.9 Object 606C$_h$: Velocity Actual Value

This object shows the actual velocity value of the motor. The value is given in internal or user-defined velocity units (depending on object 208C$_h$, described in section 4.2.16).

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

Table 239: Object Description (606C$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	Refer to CiA402-3	SIGNED32	no

Table 240: Entry Description (606C$_h$)

6.1.10 Object 607A$_h$: Target Position

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

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

Table 241: Object Description (607A$_h$ in pp Mode)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	Refer to CiA402-3	SIGNED32	0

Table 242: Entry Description (607A$_h$ in pp Mode)

6.1.11 Object 607D$_h$: Software Position Limit

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

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

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

Table 243: Object Description (607D$_h$)

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

Table 244: Entry Description (607D$_h$)

6.1.12 Object 6081$_h$: Profile Velocity

This object indicates the configured velocity normally attained at the end of the acceleration ramp during a profiled motion and is valid for both directions of motion. The profile velocity is the maximum velocity used when driving to a new position. It is given in internal or user specific units (depending on object 208C$_h$, section 4.2.16).

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

Table 245: Object Description (6081$_h$)

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

Table 246: Entry Description (6081$_h$)

6.1.13 Object 6082$_h$: End Velocity

This object indicates the configured velocity normally attained at the end of the deceleration ramp during a profiled motion and is valid for both directions of motion. The end velocity is the velocity used when reaching the new position. It is given in internal or user specific units (depending on object 208C$_h$, section 4.2.16).

Object Description			
Index	Name	Object Type	Data Type
6082$_h$	End Velocity	Variable	UNSIGNED32

Table 247: Object Description (6082$_h$)

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

Table 248: Entry Description (6082$_h$)

6.1.14 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.
The units for object 6083$_h$ can be choosen with object 208E$_h$, described in section 4.2.17.
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 249: Object Description (6083$_h$)

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

Table 250: Entry Description (6083$_h$)

6.1.15 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.
The units for object 6084$_h$ can be choosen with object 208E$_h$, described in section 4.2.17.

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

Table 251: Object Description (6084$_h$)

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

Table 252: Entry Description (6084$_h$)

6.1.16 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 253: Object Description (6085$_h$)

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

Table 254: Entry Description (6085$_h$)

6.1.17 Object 60F2$_h$: Positioning Option Code

This object indicates the positioning behaviour in profile position mode. Only bits 0 and 1 (relative option) are supported.

Bit Definitions		
Bit 1	Bit 0	Definition
0	0	Positioning moves shall be performed relative to the preceding (internal absolute) target position.
0	1	Positioning moves shall be performed relative to the actual position demand value (object 6063$_h$).
1	0	Positioning moves shall be performed relative to the position actual value (object 6064$_h$).
1	1	reserved

Table 255: Bit Definitions of Object 60F2$_h$

Object Description			
Index	Name	Object Type	Data Type
60F2$_h$	Positioning option code	Variable	UNSIGNED16

Table 256: Object Description (60F2$_h$)

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

Table 257: Entry Description (60F2$_h$)

6.2   How to move a Motor in pp Mode

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

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

- Select pp mode by writing 1 to object 6060_h.

- Write 6 to object 6040_h to switch to READY_TO_SWITCH_ON state.

- Write 7 to object 6040_h to switch to SWITCHED_ON state.

- Write 15 to object 6040_h to switch to OPERATION_ENABLED state.

- Write the desired target position (e.g. 500000) to object $607A_h$.

- Mark the new target position as active by writing 31 to object 6040_h. The motor starts moving now.

- Reset the activation by writing 15 to object 6040_h (this can be done while the motor is still moving).

7 Profile Velocity Mode

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

The profile velocity mode covers the following sub-functions:

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

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

- Profile velocity
- Profile acceleration
- Profile deceleration
- Emergency stop
- Motion profile type

7.1 Detailed Object Specifications

7.1.1 Object 6040$_h$: Control Word

This object indicates the received command controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to figure 9 for detailed information.

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

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

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

Table 258: Structure of the Control Word in pv Mode

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

Table 259: Command Coding

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

Table 260: Object Description (6040_h in pv Mode)

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

Table 261: Entry Description (6040_h in pv Mode)

7.1.2   Object 6041_h: Status Word

This object provides the status of the PDS FSA. It reflects the status of the CiA-402 state machine. Please refer to figure 9 for detailed information. The object is structured as defined below.
For more information about the coding please refer to the CANopen Drives and motion control device profile, part 2.

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

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

Table 262: Structure of the Status Word in pv Mode

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

Table 263: 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 264: Operation Mode specific Bits in pv Mode

State Coding	
Status word	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 265: State Coding

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

Table 266: Object Description (6041$_h$ in pv Mode)

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

Table 267: Entry Description (6041_h in pv Mode)

7.1.3 Object 6062_h: Position Demand Value

This object provides the demanded position value. The value is given in microsteps. 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 268: Object Description (6062_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	Refer to CiA402-3	SIGNED32	no

Table 269: Entry Description (6062_h)

7.1.4 Object 6063_h: Position Actual Internal Value

This object provides the actual value of the encoder or the motor. Please use the sensor selection object $606A_h$ (see section 5.1.8) for selecting the motor or the encoder first. Object 6063_h indicates the actual position of the encoder or the motor, re-scaled to the microstep resolution. The value is given in microsteps.

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

Table 270: Object Description (6063_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	Refer to CiA402-3	SIGNED32	no

Table 271: Entry Description (6063_h)

7.1.5 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 272: Object Description (6064$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	Refer to CiA402-3	SIGNED32	no

Table 273: Entry Description (6064$_h$)

7.1.6 Object 6065$_h$: Following Error Window

This object indicates the configured range of tolerated position values symmetrically to the position demand value. If the position actual value is out of the following error window, a following error occurs. A following error may occur when a drive is blocked, unreachable profile velocity occurs, or at wrong closed-loop coefficients. The value shall be given in microsteps.

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

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

Object Description			
Index	Name	Object Type	Data Type
6065$_h$	Following Error Window	Variable	UNSIGNED32

Table 274: Object Description (6065$_h$)

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

Table 275: Entry Description (6065$_h$)

7.1.7 Object 606C$_h$: Velocity Actual Value

This object shows the actual velocity value of the motor. The value is given in internal or user-defined velocity units (depending on object 208C$_h$, described in section 4.2.16).

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

Table 276: Object Description (606C$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	Refer to CiA402-3	SIGNED32	no

Table 277: Entry Description (606C$_h$)

7.1.8 Object 607D$_h$: Software Position Limit

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

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

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

Table 278: Object Description (607D$_h$)

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

Table 279: Entry Description (607D$_h$)

7.1.9   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.
The units for object 6083$_h$ can be choosen with object 208E$_h$, described in section 4.2.17.
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 280: Object Description (6083$_h$)

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

Table 281: Entry Description (6083$_h$)

7.1.10   Object 6085$_h$: Quick Stop Deceleration

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

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

Table 282: Object Description (6085$_h$)

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

Table 283: Entry Description (6085$_h$)

7.1.11   Object 60FF$_h$: Target Velocity

This object indicates the configured target velocity and is used as input for the trajectory generator. Object 60FF$_h$ sets the target velocity when using profile velocity mode. The drive then accelerates or decelerates to that velocity using the acceleration and deceleration set by objects 6083$_h$ and 6084$_h$. The values are given in units which can be selected with object 208C$_h$, described in section 4.2.16.

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

Table 284: Object Description (60FF$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	see CiA402-3	SIGNED32	0

Table 285: Entry Description (60FF$_h$)

7.2 How to move a Motor in pv Mode

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

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

- Select pv mode by writing 3 to object 6060$_h$.

- Write 6 to object 6040$_h$ to switch to READY_TO_SWITCH_ON state.

- Write 7 to object 6040$_h$ to switch to SWITCHED_ON state.

- Write 15 to object 6040$_h$ to switch to OPERATION_ENABLED state.

- Write the desired target speed (e.g. 100000) to object 60FF$_h$. The motor now accelerates to that speed.

- Stop the motor by writing 0 to object 60FF$_h$.

8 Homing Mode

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

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

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

Figure 10 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 10: Homing Mode Function

Choosing a homing mode determines the following things:

- The homing signal (positive limit switch, negative limit switch, and home switch).

- The direction of actuation where appropriate.

- The position of the index pulse.

The home position and the zero position are offset by the home offset (see object $607C_h$, section 8.2.4).

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

- Negative and positive limit switches.

- Home switch.

- Index pulse of an encoder.

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

8.1 Homing Methods

The TMCM-1241 supports a subset of different standard CANopen homing methods. The homing method that is to be used can be choosen via object 6098_h (section 8.2.5).

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

Table 286: Supported CANopen Homing Methods

When using homing methods that need end switch inputs or home switch inputs please take care of their configuration (object 2005_h, section 4.2.6).

8.1.1 Homing Method 1: Homing on negative Limit Switch and Index Pulse

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

Figure 11: Homing Method 1

8.1.2   Homing Method 2: Homing on positive Limit Switch and Index Pulse

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

Figure 12: Homing Method 2

8.1.3   Homing Method 3: Homing on positive Home Switch and Index Pulse

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

Figure 13: Homing Method 3

8.1.4   Homing Method 5: Homing on negative Home Switch and Index Pulse

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

Figure 14: Homing Method 5

8.1.5 Homing Method 17: Homing on negative Limit Switch

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

Figure 15: Homing Method 17

8.1.6 Homing Method 18: Homing on positive Limit Switch

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

Figure 16: Homing Method 18

8.1.7   Homing Method 19: Homing on positive Home Switch

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

Figure 17: Homing Method 19

8.1.8   Homing Method 21: Homing on negative Home Switch

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

Figure 18: Homing Method 21

8.1.9   Homing Method 33 and 34: Homing on next Index Pulse

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

Figure 19: Homing Methods 33 and 34

8.1.10   Homing Method 35: Current Position as Home Position

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

8.2   Detailed Object Specifications

8.2.1   Object 6040$_h$: Control Word

This object indicates the received command controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to figure 9 for detailed information.

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

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

Table 287: Structure of the Control Word in hm Mode

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

Table 288: Operation Mode specific Bits in hm Mode

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

Table 289: Command Coding

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

Table 290: Object Description (6040_h in hm Mode)

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

Table 291: Entry Description (6040_h in hm Mode)

8.2.2 Object 6041_h: Status Word

This object provides the status of the PDS FSA. It reflects the status of the CiA-402 state machine. Please refer to figure 9 for detailed information. The object is structured as defined below.
For more information about the coding please refer to the CANopen Drives and motion control device profile, part 2.

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

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

Table 292: Structure of the Status Word in hm Mode

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

Table 293: 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 294: Operation Mode specific Bits in hm Mode

State Coding	
Status word	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 295: State Coding

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

Table 296: Object Description (6041$_h$ in hm Mode)

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

Table 297: Entry Description (6041$_h$ in hm Mode)

8.2.3 Object 606C$_h$: Velocity Actual Value

This object shows the actual velocity value of the motor. The value is given in internal or user-defined velocity units (depending on object 208C$_h$, described in section 4.2.16).

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

Table 298: Object Description (606C$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	Refer to CiA402-3	SIGNED32	no

Table 299: Entry Description (606C$_h$)

8.2.4 Object 607C$_h$: Home Offset

This object indicates the configured difference between the zero position for the application and the machine home position/home switch (found during homing). While homing, the machine home position is found and once the homing is completed, the zero position is offset from the home position by adding the home offset to the home position. The effect of setting the home position to a non-zero value depends on the selected homing method. The value of this object is given in microsteps. 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 300: Object Description (607C$_h$)

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

Table 301: Entry Description (607C$_h$)

8.2.5   Object 6098$_h$: Homing Method

The homing method to be used can be selected by writing to this object. Please see table 286 for a list of homing methods supported by the current version of the TMCM-1241 CANopen firmware.

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

Table 302: Object Description (6098$_h$)

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

Table 303: Entry Description (6098$_h$)

8.2.6   Object 6099$_h$: Homing Speeds

This object indicates the configured speeds used during homing procedure. The values are given in pps units or internal units selectable with object 208C$_h$ (section 4.2.16). 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 304: 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 305: Entry Description (6099$_h$)

8.2.7   Object 609A$_h$: Homing Acceleration

This object indicates the configured acceleration and deceleration to be used during homing operation. The value is given in units selected by object 208E$_h$ (section 4.2.17).

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

Table 306: Object Description (609A$_h$)

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

Table 307: Entry Description (609A$_h$)

8.3   How to start a Homing in hm Mode

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

- Select hm mode by writing 6 to object 6060$_h$.

- Write 6 to object 6040$_h$ to switch to READY_TO_SWITCH_ON state.

- Write 7 to object 6040$_h$ to switch to SWITCHED_ON state.

- Write 15 to object 6040$_h$ to switch to OPERATION_ENABLED state.

- Select homing method 19 by writing 19 to object 6098$_h$.

- Set the homing speeds by writing e.g. 50000 to object 6099$_h$ sub index 1 and e.g. 10000 to object 6099$_h$ sub index 2.

- Write 31 to object 6040$_h$ to start the homing process.

- Press and release the home switch.

- When homing has finished, write 15 to object 6040$_h$ again.

9 Cyclic synchronous Position Mode

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

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

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

- Position demand value input directly via an object.

- Monitoring of the position.

- Limiting the position using the software limits or the hardware limit switches.

9.1 Detailed Object Specifications

9.1.1 Object 6040_h: Control Word

This object indicates the received command controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to figure 9 for detailed information. The cyclic synchronous position mode does not use any mode specific bits of the control word.

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

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

Table 308: Structure of the Control Word in csp Mode

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

Table 309: Command Coding

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

Table 310: Object Description (6040_h in csp Mode)

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

Table 311: Entry Description (6040_h in csp Mode)

9.1.2   Object 6041_h: Status Word

This object provides the status of the PDS FSA. It reflects the status of the CiA-402 state machine. Please refer to figure 9 for detailed information. The object is structured as defined below.
For more information about the coding please refer to the CANopen Drives and motion control device profile, part 2.

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

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

Table 312: Structure of the Status Word in csp Mode

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

Table 313: 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 314: Operation Mode specific Bits in csp Mode

State Coding	
Status word	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 315: State Coding

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

Table 316: Object Description (6041$_h$ in csp Mode)

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

Table 317: Entry Description (6041$_h$ in csp Mode)

9.1.3 Object 6062$_h$: Position Demand Value

This object provides the demanded position value. The value is given in microsteps. 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 318: Object Description (6062$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	Refer to CiA402-3	SIGNED32	no

Table 319: Entry Description (6062$_h$)

9.1.4 Object 6063$_h$: Position Actual Internal Value

This object provides the actual value of the encoder or the motor. Please use the sensor selection object 606A$_h$ (see section 5.1.8) for selecting the motor or the encoder first. Object 6063$_h$ indicates the actual position of the encoder or the motor, re-scaled to the microstep resolution. The value is given in microsteps.

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

Table 320: Object Description (6063$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	Refer to CiA402-3	SIGNED32	no

Table 321: 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 322: Object Description (6064$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	Refer to CiA402-3	SIGNED32	no

Table 323: Entry Description (6064$_h$)

9.1.6 Object 606C$_h$: Velocity Actual Value

This object shows the actual velocity value of the motor. The value is given in internal or user-defined velocity units (depending on object 208C$_h$, described in section 4.2.16).

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

Table 324: Object Description (606C$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	Refer to CiA402-3	SIGNED32	no

Table 325: Entry Description (606C$_h$)

9.1.7 Object 607A$_h$: Target Position

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

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

Table 326: Object Description (607A$_h$ in csp Mode)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	Refer to CiA402-3	SIGNED32	0

Table 327: Entry Description (607A$_h$ in csp Mode)

9.1.8 Object 607D$_h$: Software Position Limit

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

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

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

Table 328: Object Description (607D$_h$)

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

Table 329: Entry Description (607D$_h$)

9.1.9 Object 60B0$_h$: Position Offset

This object provides an offset to the target position (object 607A$_h$, see section 9.1.7)). The value is given in microsteps and will be added to the target position.

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

Table 330: Object Description (60B0$_h$)

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

Table 331: Entry Description (60B0$_h$)

9.1.10   Object 60C2$_h$: Interpolation Time Period

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

Object Description			
Index	Name	Object Type	Data Type
60C2$_h$	Offset Torque	Vecord	Interpolation time period record (0080$_h$)

Table 332: 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 333: Entry Description (60C2$_h$)

10 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.

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.
1000_h	1	0	0	0	0	**Generic error: open load bridge A** The motor driver indicates open load on bridge A. It is possible that the motor cable is broken or that there is an error in the power amplifier itself.
1000_h	2	0	0	0	0	**Generic error: open load bridge B** The motor driver indicates open load on bridge B. It is possible that the motor cable is broken or that there is an error in the power amplifier itself.
2310_h	0	0	0	0	0	**Overcurrent high side** The motor driver indicates an overcurrent on the high side. This can be caused by a short circuit in the driver stage.
2311_h	0	0	0	0	0	**Overcurrent bridge B** The motor driver indicates that there is overcurrent on bridge B. This can be caused by a short circuit in the motor itself or in the motor driver stage.
2312_h	0	0	0	0	0	**Overcurrent bridge A** The motor driver indicates that there is overcurrent on bridge A. This can be caused by a short circuit in the motor itself or in the motor driver stage.
3230_h	0	0	0	0	0	**stallGuard2 error** The actual load value exceeds the stallGuard2 limit.
4310_h	1	0	0	0	0	**Overtemperature pre-warning** The temperature in the motor driver exceeds the pre-warning limit.
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.

Error code	Additional byte					Description
	1	2	3	4	5	
8110_h	1	255	0	0	0	**CAN controller overflow** The receive message buffer of the CAN controller hardware is full and some CAN messages are lost.
8110_h	2	255	0	0	0	**CAN Tx buffer overflow** The software CAN transmit buffer is full and thus some CAN messages are lost.
8110_h	3	255	0	0	0	**CAN Rx buffer overflow** The software CAN receive buffer is full and so some CAN messages are lost.
8120_h	0	255	0	0	0	**CAN error passive** The CAN controller has detected communication errors and has entered the CAN Error passive state.
8130_h	0	255	0	0	0	**Heartbeat or lifeguard error** The module did not receive a heartbeat or lifeguard message in time.
8140_h	0	255	0	0	0	**CAN controller recovered from bus-off state** The CAN controller has detected too many errors and has changed into the bus-off state. The drive has been stopped and disabled. This message is sent after the CAN controller has recovered from bus-off state and is bus-on again.
8210_h	0	255	0	0	0	**PDO not processed due to length error** A PDO sent to the module could not be processed because too few bytes were supplied.
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.

Error code	Additional byte					Description
	1	2	3	4	5	

Table 334: Emergency Messages (EMCY)

11 Figures Index

12 Tables Index

13   Supplemental Directives

13.1   Producer Information

13.2   Copyright

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

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

13.3   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.

13.4   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.

13.5   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.

13.6   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.

13.7   Collateral Documents & Tools

This product documentation is related and/or associated with additional tool kits, firmware and other items, as provided on the product page at: www.trinamic.com.

14 Revision History

14.1 Firmware Revision

Version	Date	Author	Description
V3.21	2018-MAY-03	OK	First release.
V3.22		OK	Not released.
V3.23	2020-OCT-07	OK	Parameter store/restore functions extended. Halt bit supported in pp, pv and hm mode. Same operation mode can also be written twice.

Table 335: Firmware Revision

14.2 Document Revision

Version	Date	Author	Description
V1.00	2018-MAY-03	OK	First release.
V1.01	2020-OCT-07	OK	Firmware V3.23 included.
V1.02	2021-JUL-30	OK	Missing object $20A7_h$ added.

Table 336: Document Revision