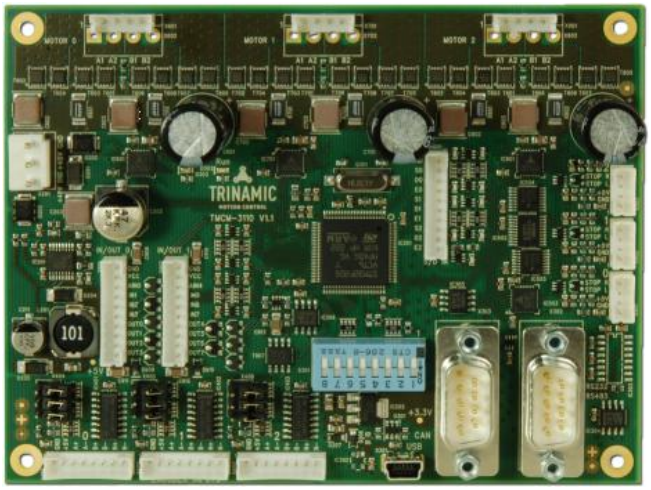
CANopen Firmware Version V3.16

CANopen MANUAL



TMCM-3110

3-Axis Stepper
Controller / Driver
2.8 A / 48 V
USB, RS485, and CAN
Step/Dir Interface
Encoder Interface

CANopen

UNIQUE FEATURES

coolStep™

stallGuard2™

TRINAMIC Motion Control GmbH & Co. KG
Hamburg, Germany

www.trinamic.com

TRINAMIC
MOTION CONTROL

Table of Contents

Table of Figures

1 Preface

This document specifies objects and modes of operation for the TMCM-3110 with CANopen firmware. As this product is a stepper motor controller and driver module, the use of the CiA DSP402 protocol (described in the *CiA CANopen drives and motion control device profile, Part 2*) is fundamental. The CANopen firmware is designed to fulfill the DS301 version 4.02 and DS402 version 3.0 standards. The CiA conformance has also been tested. This manual assumes that the reader is already familiar with the basics of the CANopen protocol (especially DS301 and DS402). On the following pages you will find an overview and afterwards the information will be more in detail.

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

1.1 General Features of the CANopen Implementation

MAIN CHARACTERISTICS

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

SDO communication

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

PDO communication

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

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

Further Characteristics

SYNC:	consumer (TPDO3 and TPDO6 are synchronous PDOs)
Emergency:	producer
RTR:	supported only for node guarding/life guarding

1.2 Abbreviations

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 1.1 Abbreviations

1.3 Firmware Update

The software running on the microprocessor of each module 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. Therefore, use the USB interface.

TRINAMICs Unique Features – Easy to Use with CANopen

stallGuard2™ stallGuard2 is a high-precision sensorless load measurement using the back EMF on 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 goes to zero or near to zero. This is the most energy-efficient point of operation for the motor.

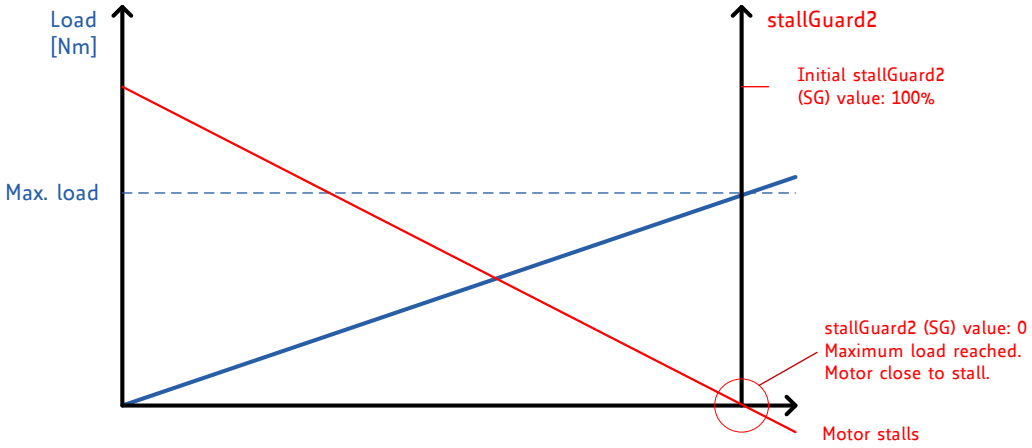


Figure 1.1 stallGuard2 load measurement SG as a function of load

coolStep™ coolStep is a load-adaptive automatic current scaling based on the load measurement via stallGuard2 adapting the required current to the load. Energy consumption can be reduced by as much as 75%. coolStep allows substantial energy savings, especially for motors which see varying loads or operate at a high duty cycle. Because a stepper motor application needs to work with a torque reserve of 30% to 50%, even a constant-load application allows significant energy savings because coolStep automatically enables torque reserve when required. Reducing power consumption keeps the system cooler, increases motor life, and allows reducing cost.

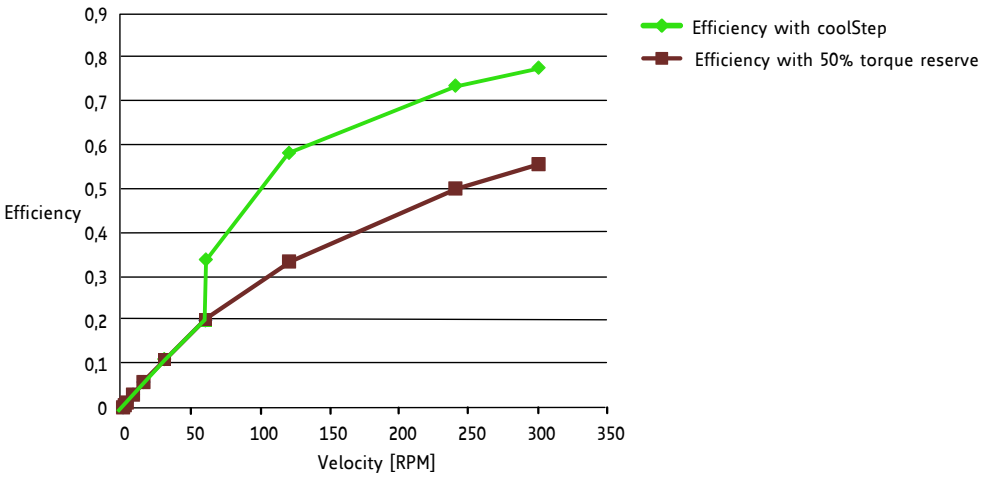


Figure 1.2 Energy efficiency example with coolStep

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 the application layer offers to an application is logically divided over different *service data objects* (SDO) in the application layer. A service object offers a specific functionality and all the related services.

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

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

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

Table 2.1 Service primitives

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

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

Table 2.2 Service types

2.2 NMT State Machine

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

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

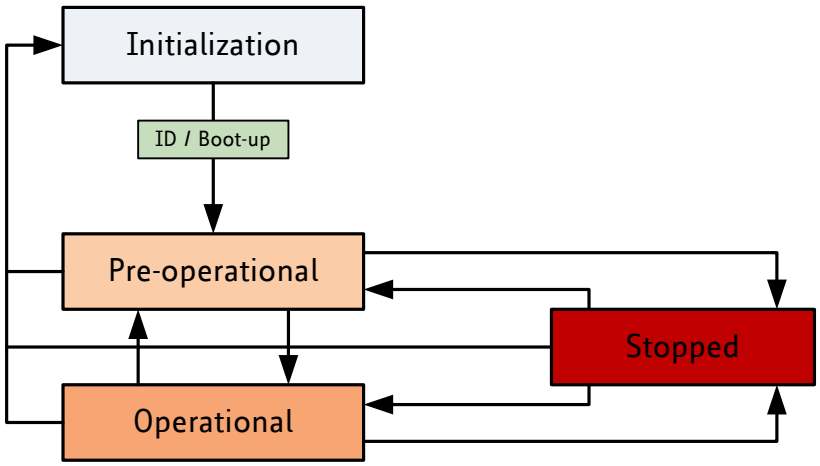


Figure 2.1 Overview CANopen NMT state machine

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

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

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

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

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

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

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

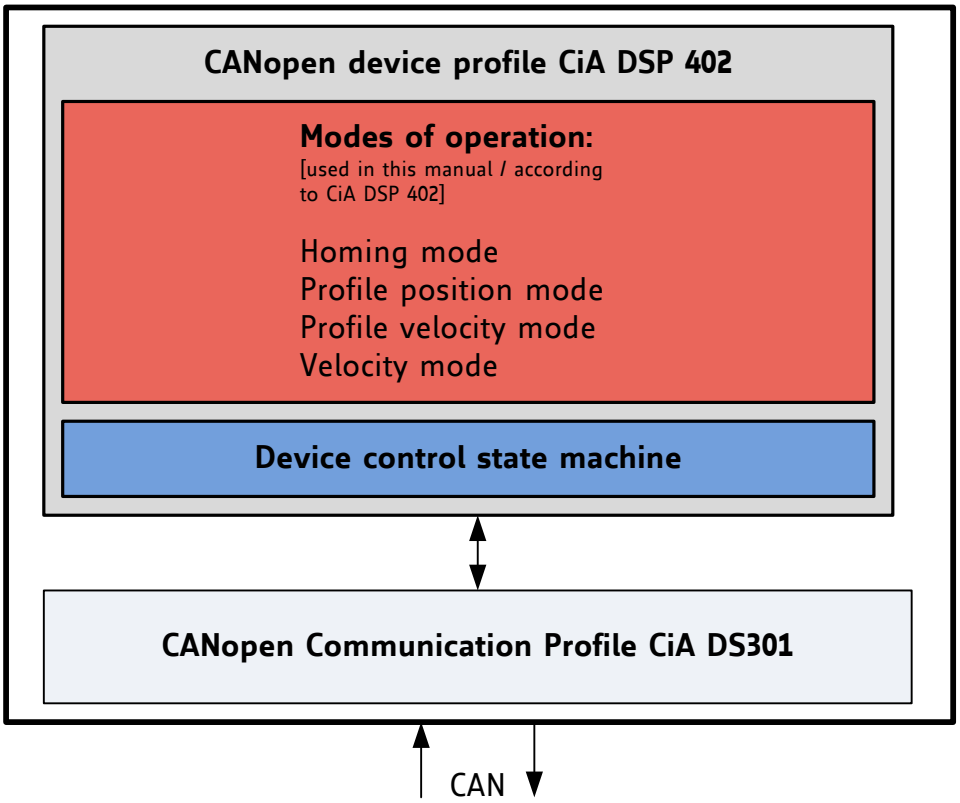


Figure 2.2 Communication architecture

2.3 Device Model

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

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

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

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

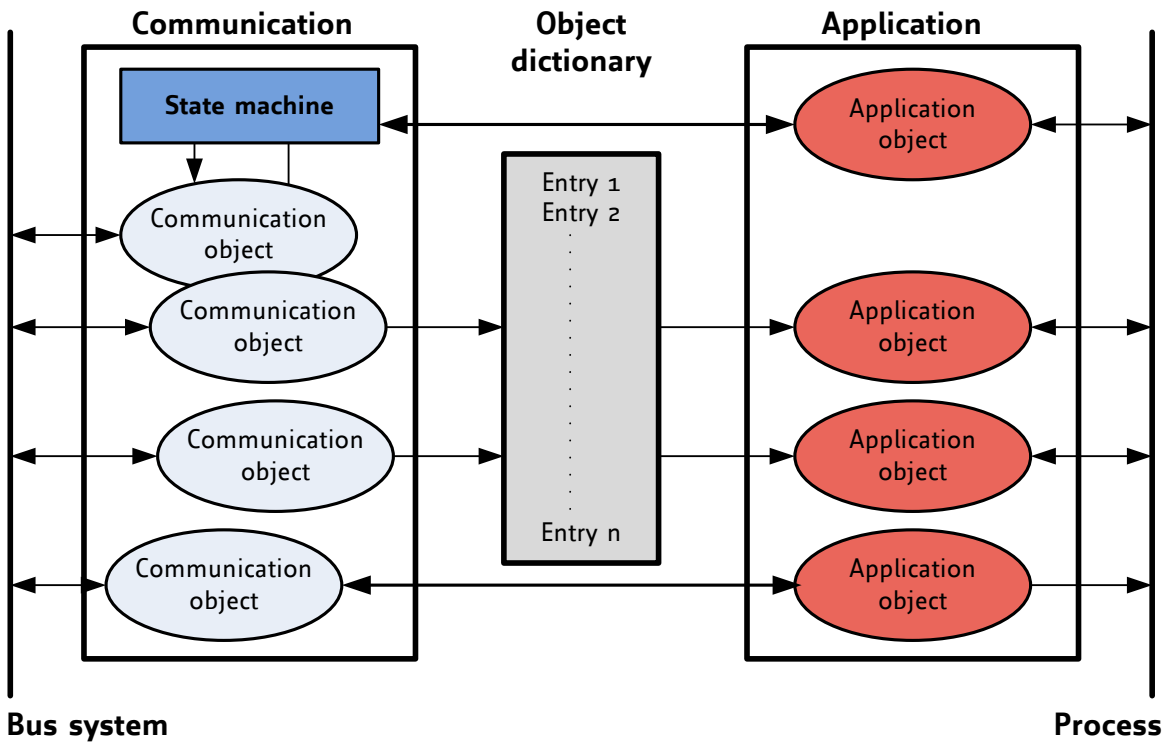


Figure 2.3 Device model

2.4 Object Dictionary

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

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

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

Table 2.3 Object Dictionary

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

*[2] The manufacturer area at indices 2000_h through $5FFF_h$ contains manufacturer specific objects. These objects control the special features of the TRINAMIC motion control device TMCM-3110.

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

3 Communication Objects

3.1 Detailed Object Specifications

3.1.1 Object 1000_h: Device Type

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

3.1.2 Object 1001_h: Error Register

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

STRUCTURE OF THE ERROR REGISTER

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

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

3.1.3 Object 1005_h: COB-ID SYNC Message

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

3.1.4 Object 1008_h: Manufacturer Device Name

This object contains the manufacturer device name.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
1008_h	Manufacturer device name	Variable	Visible string

ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00_h	const.	no	no	TMCM-3110

3.1.5 Object 1009_h: Manufacturer Hardware Version

This object contains the hardware version description.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
1009_h	Manufacturer hardware version	Variable	Visible string

ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00_h	const.	no	no	*depends on module*

3.1.6 Object $100A_h$: Manufacturer Software Version

This object contains the software version description.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$100A_h$	Manufacturer software version	Variable	Visible string

ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00_h	Const.	no	no	*according to software version*

3.1.7 Object $100C_h$: Guard Time

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$100C_h$	Guard time	Variable	UNSIGNED16

ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00_h	rw *ro, if life guarding is not supported*	no	UNSIGNED16	0000_h

3.1.8 Object $100D_h$: Life Time Factor

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$100D_h$	Life time factor	Variable	UNSIGNED8

ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00_h	rw *ro, if life guarding is not supported*	no	UNSIGNED8	00_h

3.1.9 Object 1010_h: Store Parameters

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

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

SEVERAL PARAMETER GROUPS ARE DISTINGUISHED:

- Sub-index 0_h: contains the largest sub-index that is supported.
- Sub-index 1_h: saves all parameters.
- Sub-index 2_h: saves communication parameters $100C_h$, $100D_h$, 1015_h, 1017_h, and 1029_h.
- Sub-index 4_h: saves motor axis 1 parameters
- Sub-index 5_h: saves motor axis 2 parameters
- Sub-index 6_h: saves motor axis 3 parameters
- Sub-index 7_h: saves device parameters / TRINAMIC specific parameters (2000_h... $270E_h$).

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

Storage write access structure

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

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

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

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

Storage read access structure

	UNSIGNED 32 MSB		LSB
bits	31-2	1	0
	Reserved	1/0	1/0

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

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

3.1.10 Object 1011$_h$: Restore Default Parameters

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

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

SEVERAL PARAMETER GROUPS ARE DISTINGUISHED:

- Sub-index 0$_h$: contains the largest sub-index that is supported.
- Sub-index 1$_h$: restores all parameters.
- Sub-index 2$_h$: restores communication parameters 100C$_h$, 100D$_h$, 1015$_h$, 1017$_h$, and 1029$_h$.
- Sub-index 4$_h$: restores motor axis 1 parameters
- Sub-index 5$_h$: restores motor axis 2 parameters
- Sub-index 6$_h$: restores motor axis 3 parameters
- Sub-index 7$_h$: restores device parameters / TRINAMIC specific parameters (2000$_h$... 270E$_h$).

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

Signature	MSB			LSB
ASCII	d	a	o	l
hex	64_h	61_h	$6F_h$	$6C_h$

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

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

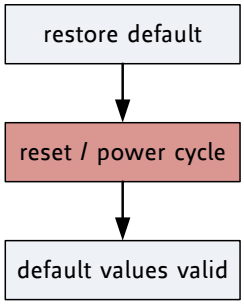


Figure 3.1 Restore procedure

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

Structure of restore read access

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

3.1.11 Object 1014_h: COB-ID Emergency Object

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

3.1.12 Object 1015_h: Inhibit Time EMCY

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
1015_h	Inhibit time EMCY	Variable	UNSIGNED16

ENTRY DESCRIPTION

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

3.1.13 Object 1016_h: Consumer Heartbeat Time

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

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

Table 3.1 Structure of consumer heartbeat time entry

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

3.1.14 Object 1017_h: Producer Heartbeat Time

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

3.1.15 Object 1018_h: Identity Object

The object 1018_h contains general information about the device.

- The Vendor ID (sub-index 01_h) contains a unique value allocated to each manufacturer. The vendor ID of TRINAMIC is 0286_h.
- The manufacturer-specific Product code (sub-index 2_h) identifies a specific device version.
- The Manufacturer-specific Revision number (sub-index 3_h) consists of a major revision number and a minor revision number.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type	Category
1018_h	Identity object	RECORD	Identity	Optional

ENTRY DESCRIPTION

Sub-Index	Description	Access	PDO Mapping	Value Range	Default Value
00_h	Number of entries	ro	no	0... 3	number of entries:3
01_h	Vendor ID (TRINAMIC)	ro	no	UNSIGNED32	reads 0286_h
02_h	Product code	ro	no	UNSIGNED32	1180, 1160, or 1140
03_h	Revision number	ro	no	UNSIGNED32	firmware revision number; reads e.g. 3016_h for version 3.16
04_h	Serial number	ro	no	UNSIGNED32	serial number

3.1.16 Object 1023_h: OS Command

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

3.1.17 Object 1029_h: Error Behavior

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

- Communication error
- Application error

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

Sub-Index	Description	Access	PDO Mapping	Value Range	Default Value
00_h	Largest sub-index supported	ro	no	Number of entries: 2	2
01_h	COB-ID used by PDO	rw	no	UNSIGNED32	Index 1400_h: 200_h + Node-ID Index 1401_h: 300_h + Node-ID Index 1402_h: 400_h + Node-ID Index 1403_h: 500_h + Node-ID Index 1405_h: 0
02_h	Transmission type	rw	no	UNSIGNED8	Index 1400_h: Ff_h Index 1401_h: Ff_h Index 1402_h: Ff_h Index 1403_h: Ff_h Index 1405_h: Ff_h

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

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

3.1.21 Objects 1600_h-1605_h: Receive PDO Mapping

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

Sub-Index	Description	Access	PDO Mapping	Value Range	Default Value
00_h	Number of mapped application objects in PDO	rw	no	1... 3 0 deactivated	Index 1600_h: 1 Index 1601_h: 2 Index 1602_h: 2 Index 1603_h: 2 Index 1605_h: 2
01_h	Mapping entry 1	rw	no	UNSIGNED32	Index 1600_h: 60400010_h Index 1601_h: 60400010_h Index 1602_h: 60400010_h Index 1603_h: 60400010_h Index 1605_h: 60400010_h
02_h	Mapping entry 2	rw	no	UNSIGNED32	Index 1600_h: 0 Index 1601_h: 60600008_h Index 1602_h: $607A0020_h$ Index 1603_h: $60FF0020_h$ Index 1605_h: 60420010_h
03_h	Mapping entry 3	rw	no	UNSIGNED32	Index 1600_h: 0 Index 1601_h: 0 Index 1602_h: 0 Index 1603_h: 0 Index 1605_h: 0

3.1.22 Objects 1640_h-1645_h: Receive PDO Mapping

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

Sub-Index	Description	Access	PDO Mapping	Value Range	Default Value
00_h	Number of mapped application objects in PDO	rw	no	1... 3 0 deactivated	Index 1640_h: 1 Index 1641_h: 2 Index 1642_h: 2 Index 1643_h: 2 Index 1645_h: 2
01_h	Mapping entry 1	rw	no	UNSIGNED32	Index 1640_h: 68400010_h Index 1641_h: 68400010_h Index 1642_h: 68400010_h Index 1643_h: 68400010_h Index 1645_h: 68400010_h
02_h	Mapping entry 2	rw	no	UNSIGNED32	Index 1640_h: 0 Index 1641_h: 68600008_h Index 1642_h: $687A0020_h$ Index 1643_h: $68FF0020_h$ Index 1645_h: 68420010_h
03_h	Mapping entry 3	rw	no	UNSIGNED32	Index 1640_h: 0 Index 1641_h: 0 Index 1642_h: 0 Index 1643_h: 0 Index 1645_h: 0

3.1.23 Objects 1680_h-1685_h: Receive PDO Mapping

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

Sub-Index	Description	Access	PDO Mapping	Value Range	Default Value
00_h	Number of mapped application objects in PDO	rw	no	1... 3 0 deactivated	Index 1680_h: 1 Index 1681_h: 2 Index 1682_h: 2 Index 1683_h: 2 Index 1685_h: 2
01_h	Mapping entry 1	rw	no	UNSIGNED32	Index 1680_h: 70400010_h Index 1681_h: 70400010_h Index 1682_h: 70400010_h Index 1683_h: 70400010_h Index 1685_h: 70400010_h
02_h	Mapping entry 2	rw	no	UNSIGNED32	Index 1680_h: 0 Index 1681_h: 70600008_h Index 1682_h: $707A0020_h$ Index 1683_h: $70FF0020_h$ Index 1685_h: 70420010_h
03_h	Mapping entry 3	rw	no	UNSIGNED32	Index 1680_h: 0 Index 1681_h: 0 Index 1682_h: 0 Index 1683_h: 0 Index 1685_h: 0

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

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
1800_h-1805_h	Transmit PDO communication parameter	RECORD	PDO CommPar
1800_h	TPDO 1 transmit communication parameter	RECORD	PDO CommPar
1801_h	TPDO 2 transmit0 communication parameter	RECORD	PDO CommPar
1802_h	TPDO 3 transmit communication parameter	RECORD	PDO CommPar
1803_h	TPDO 4 transmit communication parameter	RECORD	PDO CommPar
1805_h	TPDO 6 transmit communication parameter	RECORD	PDO CommPar

ENTRY DESCRIPTION

Sub-Index	Description	Access	PDO Mapping	Value Range	Default Value
00_h	Largest sub-index supported	ro	no	2… 5 Number of entries: 5	5
01_h	TPDO 1-4, 6: COB-ID	rw	no	UNSIGNED32	Index 1800_h: 180_h + Node-ID Index 1801_h: 280_h + Node-ID Index 1802_h: 380_h + Node-ID Index 1803_h: 480_h + Node-ID Index 1805_h: 0
02_h	Transmission type	rw	no	UNSIGNED8	Index 1800_h: ff_h Index 1801_h: ff_h Index 1802_h: 1 Index 1803_h: 1 Index 1805_h: 1
03_h	Inhibit time	rw	no	UNSIGNED16	0
04_h	Compatibility entry	ro	no	UNSIGNED8	0
05_h	Event timer	rw	no	0 not used UNSIGNED16	0

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

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
1840_h-1845_h	Transmit PDO communication parameter	RECORD	PDO CommPar
1840_h	TPDO 65 transmit communication parameter	RECORD	PDO CommPar
1841_h	TPDO 66 transmit0 communication parameter	RECORD	PDO CommPar
1842_h	TPDO 67 transmit communication parameter	RECORD	PDO CommPar
1843_h	TPDO 68 transmit communication parameter	RECORD	PDO CommPar
1845_h	TPDO 70 transmit communication parameter	RECORD	PDO CommPar

ENTRY DESCRIPTION

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

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

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
1880_h-1885_h	Transmit PDO communication parameter	RECORD	PDO CommPar
1880_h	TPDO 129 transmit communication parameter	RECORD	PDO CommPar
1881_h	TPDO 130 transmit0 communication parameter	RECORD	PDO CommPar
1882_h	TPDO 131 transmit communication parameter	RECORD	PDO CommPar
1883_h	TPDO 132 transmit communication parameter	RECORD	PDO CommPar
1885_h	TPDO 134 transmit communication parameter	RECORD	PDO CommPar

ENTRY DESCRIPTION

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

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

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$1A00_h$-$1A05_h$	Transmit PDO mapping parameter	RECORD	PDO mapping
$1A00_h$	TPDO 1: transmit PDO mapping parameter	RECORD	PDO mapping
$1A01_h$	TPDO 2: transmit PDO mapping parameter	RECORD	PDO mapping
$1A02_h$	TPDO 3: transmit PDO mapping parameter	RECORD	PDO mapping
$1A03_h$	TPDO 4: transmit PDO mapping parameter	RECORD	PDO mapping
$1A05_h$	TPDO 6: transmit PDO mapping parameter	RECORD	PDO mapping

ENTRY DESCRIPTION

Sub-Index	Description	Access	PDO Mapping	Value Range	Default Value
00_h	Number of mapped application objects in PDO	rw	no	0 deactivated Number of entries: 1… 3	Index $1A00_h$: 1 Index $1A01_h$: 2 Index $1A02_h$: 2 Index $1A03_h$: 2 Index $1A05_h$: 2
01_h	Mapping entry 1	rw	no	UNSIGNED32	Index $1A00_h$: 60410010_h Index $1A01_h$: 60410010_h Index $1A02_h$: 60410010_h Index $1A03_h$: 60410010_h Index $1A05_h$: 60410010_h
02_h	Mapping entry 2	rw	no	UNSIGNED32	Index $1A00_h$: 0 Index $1A01_h$: 60610008_h Index $1A02_h$: 60640020_h Index $1A03_h$: $606c0020_h$ Index $1A05_h$: 60440010_h
03_h	Mapping entry 3	rw	no	UNSIGNED32	Index $1A00_h$: 0 Index $1A01_h$: 0 Index $1A02_h$: 0 Index $1A03_h$: 0 Index $1A05_h$: 0

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

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$1A40_h$-$1A45_h$	Transmit PDO mapping parameter	RECORD	PDO mapping
$1A40_h$	TPDO 65: transmit PDO mapping parameter	RECORD	PDO mapping
$1A41_h$	TPDO 66: transmit PDO mapping parameter	RECORD	PDO mapping
$1A42_h$	TPDO 67: transmit PDO mapping parameter	RECORD	PDO mapping
$1A43_h$	TPDO 68: transmit PDO mapping parameter	RECORD	PDO mapping
$1A45_h$	TPDO 70: transmit PDO mapping parameter	RECORD	PDO mapping

ENTRY DESCRIPTION

Sub-Index	Description	Access	PDO Mapping	Value Range	Default Value
00_h	Number of mapped application objects in PDO	rw	no	0 deactivated Number of entries: 1… 3	Index $1A40_h$: 1 Index $1A41_h$: 2 Index $1A42_h$: 2 Index $1A43_h$: 2 Index $1A45_h$: 2
01_h	Mapping entry 1	rw	no	UNSIGNED32	Index $1A40_h$: 68410010_h Index $1A41_h$: 68410010_h Index $1A42_h$: 68410010_h Index $1A43_h$: 68410010_h Index $1A45_h$: 68410010_h
02_h	Mapping entry 2	rw	no	UNSIGNED32	Index $1A40_h$: 0 Index $1A41_h$: 68610008_h Index $1A42_h$: 68640020_h Index $1A43_h$: $686c0020_h$ Index $1A45_h$: 68440010_h
03_h	Mapping entry 3	rw	no	UNSIGNED32	Index $1A40_h$: 0 Index $1A41_h$: 0 Index $1A42_h$: 0 Index $1A43_h$: 0 Index $1A45_h$: 0

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

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$1A80_h$-$1A85_h$	Transmit PDO mapping parameter	RECORD	PDO mapping
$1A80_h$	TPDO 129: transmit PDO mapping parameter	RECORD	PDO mapping
$1A81_h$	TPDO 130: transmit PDO mapping parameter	RECORD	PDO mapping
$1A82_h$	TPDO 131: transmit PDO mapping parameter	RECORD	PDO mapping
$1A83_h$	TPDO 132: transmit PDO mapping parameter	RECORD	PDO mapping
$1A85_h$	TPDO 134: transmit PDO mapping parameter	RECORD	PDO mapping

ENTRY DESCRIPTION

Sub-Index	Description	Access	PDO Mapping	Value Range	Default Value
00_h	Number of mapped application objects in PDO	rw	no	0 deactivated Number of entries: 1… 3	Index $1A80_h$: 1 Index $1A81_h$: 2 Index $1A82_h$: 2 Index $1A83_h$: 2 Index $1A85_h$: 2
01_h	Mapping entry 1	rw	no	UNSIGNED32	Index $1A80_h$: 70410010_h Index $1A81_h$: 70410010_h Index $1A82_h$: 70410010_h Index $1A83_h$: 70410010_h Index $1A85_h$: 70410010_h
02_h	Mapping entry 2	rw	no	UNSIGNED32	Index $1A80_h$: 0 Index $1A81_h$: 70610008_h Index $1A82_h$: 70640020_h Index $1A83_h$: $706c0020_h$ Index $1A85_h$: 70440010_h
03_h	Mapping entry 3	rw	no	UNSIGNED32	Index $1A80_h$: 0 Index $1A81_h$: 0 Index $1A82_h$: 0 Index $1A83_h$: 0 Index $1A85_h$: 0

4 Device Profile Objects (CiA402) and Modes of Operation

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

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

THE FOLLOWING OPERATION MODES CAN BE CHOSEN:

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

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

4.1 Detailed Object Specifications

4.1.1 Object $605A_h$, $685A_h$, and $705A_h$: Quick Stop Option Code

These objects indicate what action is performed when the quick stop function is executed. The slow down ramp is the deceleration value of the used mode of operations. The following quick stop option codes are supported in the current version of the CANopen firmware.

VALUE DEFINITION

Value	Definition
1	Slow down on *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*

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$605A_h$ (motor 0) $685A_h$ (motor 1) $705A_h$ (motor 2)	Quick stop option code	Variable	SIGNED16

ENTRY DESCRIPTION

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

4.1.2 Object $605B_h$, $685B_h$, and $705B_h$: Shutdown Option Code

These objects indicate 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 drive power stage)

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$605B_h$ (motor 0) $685B_h$ (motor 1) $705B_h$ (motor 2)	Shutdown option code	Variable	UNSIGNED16

ENTRY DESCRIPTION

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

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

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

VALUE DEFINITION

Value	Definition
1	Slow down with *slow down ramp*; disable of the *drive function*

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$605C_h$ (motor 0) $685C_h$ (motor 1) $705C_h$ (motor 2)	Disable operation option code	Variable	UNSIGNED16

ENTRY DESCRIPTION

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

4.1.4 Object $605D_h$, $685D_h$, and $705D_h$: Halt Option Code

These objects indicate what action is performed when the halt function is executed. The slow down ramp is the deceleration value of the used mode of operation. The halt option code always has the value 1 as only this is supported.

VALUE DEFINITION

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$605D_h$ (motor 0) $685D_h$ (motor 1) $705D_h$ (motor 2)	Halt option code	Variable	UNSIGNED16

ENTRY DESCRIPTION

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

4.1.5 Object $605E_h$, $685E_h$, and $705E_h$: Fault Reaction Option Code

These objects indicate 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*

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$605E_h$ (motor 0) $685E_h$ (motor 1) $705E_h$ (motor 2)	Fault reaction option code	Variable	UNSIGNED16

ENTRY DESCRIPTION

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

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

These objects indicate the requested operation mode.

Supported operating modes are:

VALUE DEFINITION

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

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

These objects show the operating mode that is set.

VALUE DEFINITION

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

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

These objects provide the source of the velocity sensor actual value. Use these objects to select whether encoders are to be used or not.

VALUE DEFINITION

Value	Definition
0	Encoder used
-1	No encoder

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$606A_h$ (motor 0) $686A_h$ (motor 1) $706A_h$ (motor 2)	Sensor selection code	Variable	SIGNED16

ENTRY DESCRIPTION

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

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

These objects indicate the configured deceleration used to stop a motor when the *quick stop function* is activated and the *quick stop code object* ($605A_h$/$685A_h$/$705A_h$) is set to 2 [or 6]. The value is given in the same unit as *profile acceleration object* 6083_h. The units can be chosen with objects $208E_h$/$228E_h$/$248E_h$, described in paragraph 5.2.17. The objects $208E_h$/$228E_h$/$248E_h$ read 0 when internal units are selected or 179 when user units (PPS/s) are selected.

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

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

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

All values are dimensionless.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$608F_h$ (motor 0) $688F_h$ (motor 1) $708F_h$ (motor 2)	Encoder resolution	Array	UNSIGNED32

ENTRY DESCRIPTION

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

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

These objects indicate the configured maximum acceleration. They are used to limit the acceleration to an acceptable value in order to prevent the specific motor and the moved mechanics from being destroyed. The value is given in internal or user specific units (depending on object $208C_h$/$228C_h$/$248C_h$, paragraph 5.2.16, object 2087_h and object 2088_h, paragraph 5.2.13 and paragraph 5.2.14)

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$60C5_h$ (motor 0) $68C5_h$ (motor 1) $70C5_h$ (motor 2)	Max acceleration	Variable	UNSIGNED32

ENTRY DESCRIPTION

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

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

These objects indicate the configured maximal deceleration. They are used to limit the deceleration to an acceptable value in order to prevent the specific motor and the moved mechanics from being destroyed. The max deceleration is given in internal or user specific units (depending on objects $208C_h$/$228C_h$/$248C_h$, paragraph 5.2.16, object 2087_h and object 2088_h, paragraph 5.2.13 and paragraph 5.2.14). Object $60C6_h$ has the same unit as the max acceleration object $60C5_h$.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$60C6_h$ (motor 0) $68C6_h$ (motor 1) $70C6_h$ (motor 2)	Max deceleration	Variable	UNSIGNED32

ENTRY DESCRIPTION

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

4.1.13 Object $60FD_h$, $68FD_h$, and $70FD_h$: Digital Inputs

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

VALUE DEFINITION

Value	Definition
0_b	Switched off
1_b	Switched on

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$60FD_h$ (motor 0) $68FD_h$ (motor 1) $70FD_h$ (motor 2)	Digital inputs	Variable	UNSIGNED32

ENTRY DESCRIPTION

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

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

These objects provide information on the supported drive modes and contain always the *value* 00000025_h which means that the following modes are provided by the drive:

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6502_h (motor 0) $6D02_h$ (motor 1) 7502_h (motor 2)	Supported drive modes	Variable	UNSIGNED32

ENTRY DESCRIPTION

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

4.2 Profile Position Mode

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

Please refer to objects 6060_h/6860_h/7060_h (section 4.1.6) for information about how to choose an operation mode. Object 6061_h/6861_h/7061_h (section 0) show the operation mode for each motor that is set.

4.2.1 Detailed Object Specifications

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

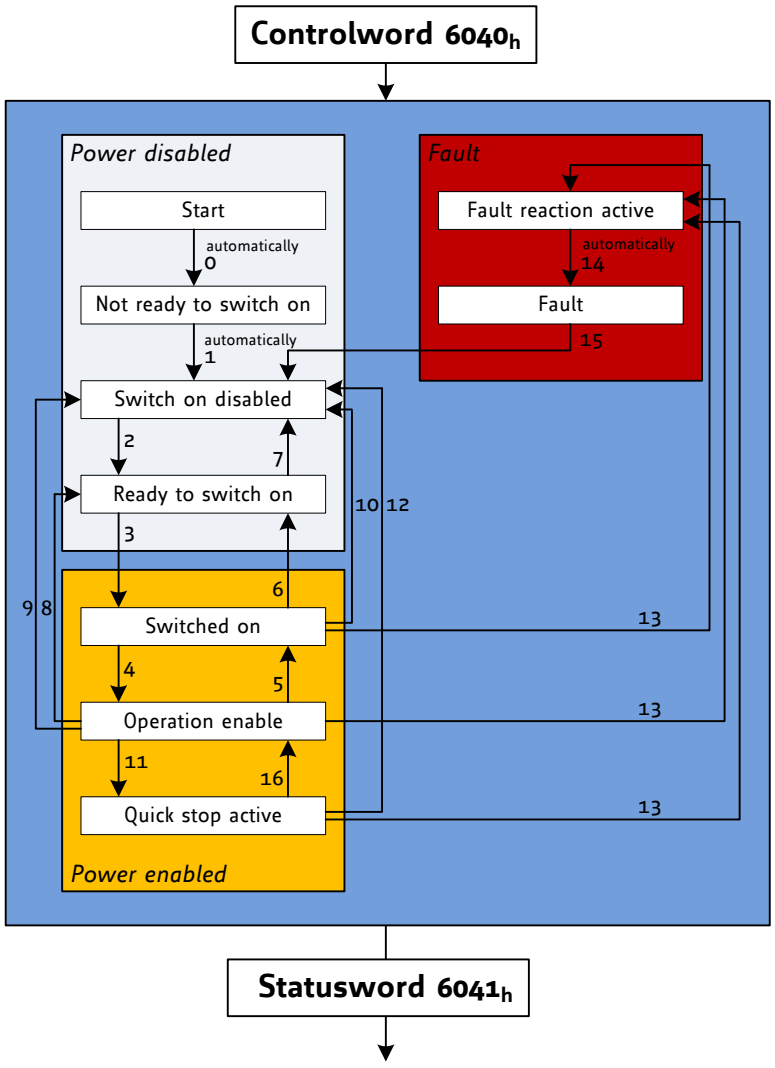


Figure 4.1 Finite state machine

NOTES ON STATE TRANSITIONS

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

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

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

STRUCTURE OF CONTROLWORD / VALUE DEFINITION

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

OPERATION MODE SPECIFIC BITS IN PP MODE

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

COMMAND CODING

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

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

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

STRUCTURE OF STATUSWORD / VALUE DEFINITION

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

TRINAMIC SPECIFIC BITS

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

OPERATION MODE SPECIFIC BITS IN PP MODE

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

STATE CODING

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

These objects indicate the configured range of tolerated position values symmetrically to the *position demand value* (objects 6062_h/6862_h/7062_h). If the *position actual value* (objects 6064_h/6864_h/7064_h) 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 is in microsteps. If the value of the following error window is 0 or FFFF $FFFF_h$, the following control will be switched off.

When the difference between *motor position* (4.2.1.3) and *encoder position* (4.2.1.4 and 4.2.1.5) is greater than the value set here, the motor will be stopped and an emergency message will be sent. Setting this object to zero will turn off this feature completely.

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

These objects indicate the configured symmetrical range of accepted positions relative to the target position. If the actual value of the *position encoder* (4.2.1.4 and 4.2.1.5) 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 FFFF $FFFF_{h,}$ the position window control is switched off.

If these objects are set to zero, the target reached event will be signaled when the *demand position* (4.2.1.3) has reached the *target position* (4.2.1.10).

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 (4.2.1.5) is within the *target position – position window* and *target position + position window*.

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

4.2.1.8 Object 6068_h, 6868_h, and 7068_h: Position Window Time

These objects indicate the configured time, during which the actual position within the *position window* is measured. The value is given in ms. If these objects are set to a value greater than zero and also the *position window* (4.2.1.7) is set to a value greater than zero the target reached event will not be signaled until the *actual position* (4.2.1.5) is at least as many milliseconds within the position window as defined by this object.

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$606C_h$ $686C_h$ $706C_h$	Velocity actual value	Variable	INTEGER32

ENTRY DESCRIPTION

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

4.2.1.10 Object $607A_h$, $687A_h$, and $707A_h$: Target Position

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$607A_h$ (motor 0) $687A_h$ (motor 1) $707A_h$ (motor 2)	Target position	Variable	SIGNED32

ENTRY DESCRIPTION

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

4.2.1.11 Object $607D_h$, $687D_h$, and $707D_h$: Software Position Limit

These objects indicate the configured maximal and minimal software position limits. They define the absolute position limits for the *position demand value* (4.2.1.3) and the *position actual value* (4.2.1.5). 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* (4.2.1.10), they are corrected internally by the *home offset* (4.3.3.4) as follows:

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

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$607D_h$ (motor 0) $687D_h$ (motor 1) $707D_h$ (motor 2)	Software position limit	Array	SIGNED32

ENTRY DESCRIPTION

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

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

These objects indicate the configured *maximum velocity* in either direction during a profiled motion. It is the maximum velocity that can be used for positioning. The value is given in internal or user specific units (depending on objects $208C_h$/$228C_h$/$248C_h$, paragraph 5.2.16).

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$607F_h$ (motor 0) $687F_h$ (motor 1) $707F_h$ (motor 2)	Maximum profile velocity	Variable	UNSIGNED32

ENTRY DESCRIPTION

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

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

These objects indicate 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 objects $208C_h$/$228C_h$/$248C_h$, paragraph 5.2.16)

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

These objects indicate the configured acceleration for each axis. The maximum acceleration to be used in profile positioning mode can be set using these objects.

Please choose the units with objects $208E_h$/$228E_h$/$248E_h$, described in paragraph 5.2.16. The objects $208E_h$/$228E_h$/$248E_h$ read 0 when internal units are selected or 179 when user units (PPS/s) are selected.

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

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

Please choose the units for object 6084_h with objects $208E_h$/$228E_h$/$248E_h$, described in paragraph 5.2.16. The objects $208E_h$/$228E_h$/$248E_h$ read 0 when internal units are selected or 179 when user units (PPS/s) are selected.

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

4.2.1.16 Object 6086_h, 6886_h, and 7086_h: Motion profile type

These objects indicate the configured type of motion profile used to perform a profiled motion. Set these objects to 0 for trapezoid ramps and to 1 for S-shaped ramps.

VALUE DEFINITION

Value	Definition
0	Linear ramp, trapezoidal profile

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

4.2.2 How to Move a Motor in *pp* Mode

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

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

4.3 Homing Mode

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

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

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

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

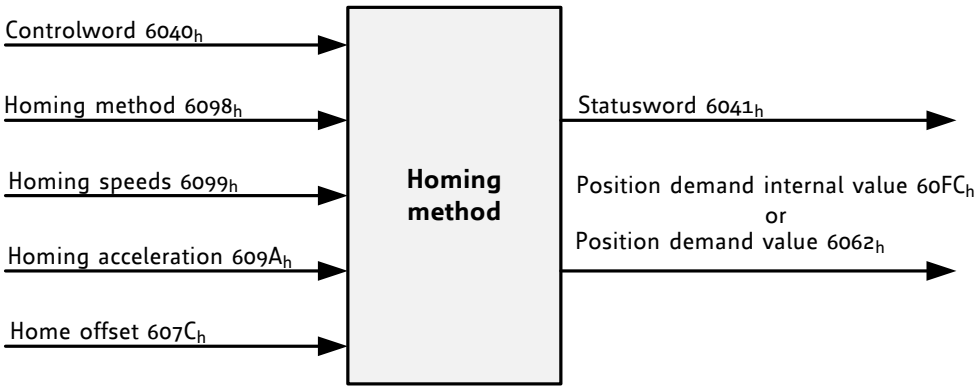


Figure 4.2 Homing mode function (motor 0)

BY CHOOSING A HOMING METHOD, THE FOLLOWING IS DETERMINED:

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

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

THERE ARE FOUR SOURCES OF HOMING SIGNAL AVAILABLE:

- negative and positive homing 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. There are several application specific methods. The homing method is used for selection.

Please refer to objects 6060_h/6860_h/7060_h (section 4.1.6) for information about how to choose an operation mode. Objects 6061_h/6861_h/7061_h (section 0) show the operation modes that are set for each motor.

4.3.1 Connecting Home Switch, Left Switch, and Right Switch

Three 2mm pitch 4 pin JST B4B-PH-K connectors are used for connecting left and right limit switches for each motor to the unit. Further, it is possible to connect home switches using digital inputs of the I/O connectors. If desired, it is possible to connect a brake. Therefore, use the outputs. *For detailed information refer to the hardware manual of your module!*

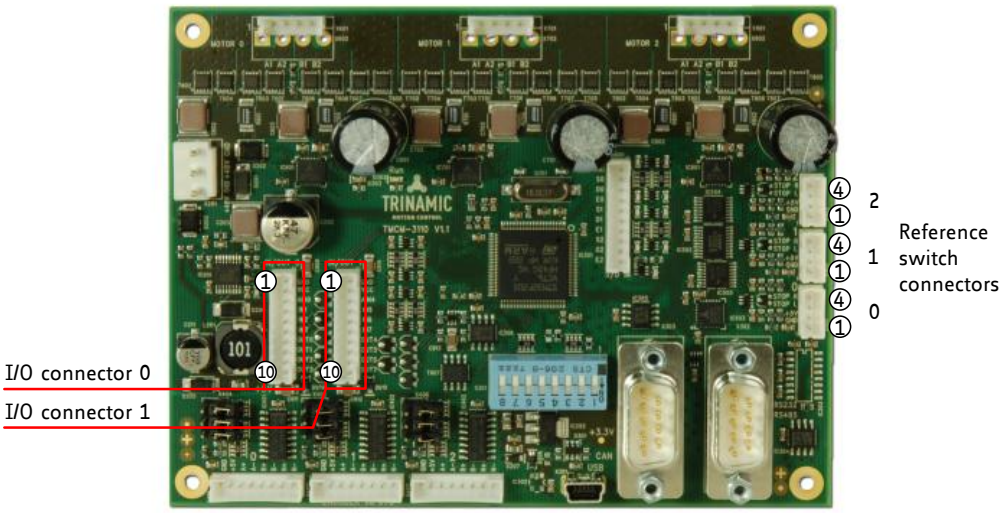


Figure 4.3 TMCM-3110 connectors

REFERENCE SWITCH CONNECTORS

For each stepper motor axis a separate reference/limit switch input connector is available.

Pin	Label	Direction	Description
1	GND	Power (GND)	Signal and system ground
2	+5V	Power (Supply output)	+5V output for external circuit
3	REF_L	Input	Input for reference / limit switch left, integrated pull-up to +5V
4	REF_R	Input	Input for reference / limit switch right, integrated pull-up to +5V

I/O CONNECTOR 0

Pin	Label	Direction	Description
1	GND	Power (GND)	GND
2	$V_{DIGITAL}$	Power (supply output)	Connected to $V_{DIGITAL}$ of Power connector
3	AIN_0	Input	Dedicated analog input, input voltage range: 0… +10V, resolution: 12bit (0… 4095)
4	IN_1	Input	Digital input (+24V compatible)
5	IN_2	Input	Digital input (+24V compatible)
6	IN_3	Input	Digital input (+24V compatible)
7	OUT_0	Output	Open-drain output (max. 100mA) Integrated freewheeling diode
8	OUT_1	Output	Open-drain output (max. 100mA) Integrated freewheeling diode to $V_{digital}$
9	OUT_2	Output	Open-drain output (max. 100mA) Integrated freewheeling diode to $V_{digital}$
10	OUT_3	Output	Open-drain output (max. 1A) Integrated freewheeling diode to $V_{digital}$

I/O CONNECTOR 1

Pin	Label	Direction	Description
1	GND	Power (GND)	GND
2	$V_{DIGITAL}$	Power (supply output)	Connected to $V_{DIGITAL}$ of Power connector
3	AIN_4	Input	Dedicated analog input, input voltage range: 0… +10V, resolution: 12bit (0… 4095)
4	IN_5	Input	Digital input (+24V compatible)
5	IN_6	Input	Digital input (+24V compatible)
6	IN_7	Input	Digital input (+24V compatible)
7	OUT_4	Output	Open-drain output (max. 100mA) Integrated freewheeling diode
8	OUT_5	Output	Open-drain output (max. 100mA) Integrated freewheeling diode to $V_{digital}$
9	OUT_6	Output	Open-drain output (max. 100mA) Integrated freewheeling diode to $V_{digital}$
10	OUT_7	Output	Open-drain output (max. 1A) Integrated freewheeling diode to $V_{digital}$

NOTE

All inputs have resistor based voltage dividers with protection diodes. These resistors also ensure a valid GND level when left unconnected.

4.3.2 Homing Methods

There are several different methods of homing. For choosing your homing method, refer to objects 6098_h/6898_h/7098_h (4.3.3.5).

OVERVIEW: HOMING METHODS

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

METHOD 1: HOMING ON NEGATIVE LIMIT SWITCH AND INDEX PULSE

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

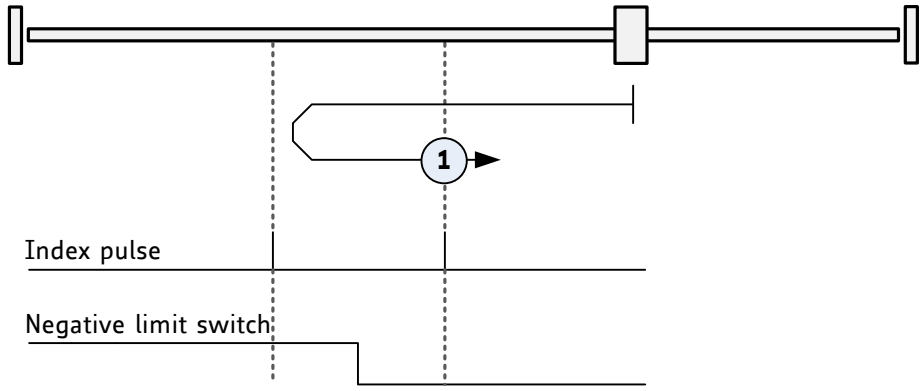


Figure 4.4 Homing on negative switch and index pulse

METHOD 2: HOMING ON POSITIVE LIMIT SWITCH AND INDEX PULSE

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

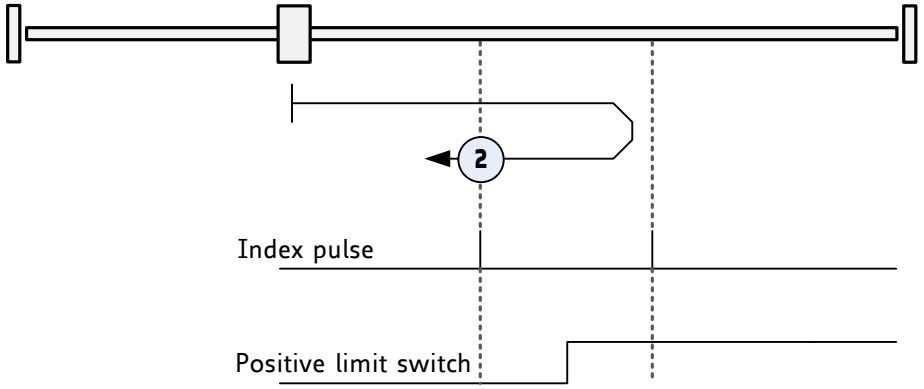


Figure 4.5 Homing on positive limit switch and index pulse

METHOD 3: HOMING ON POSITIVE HOME SWITCH AND INDEX PULSE

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

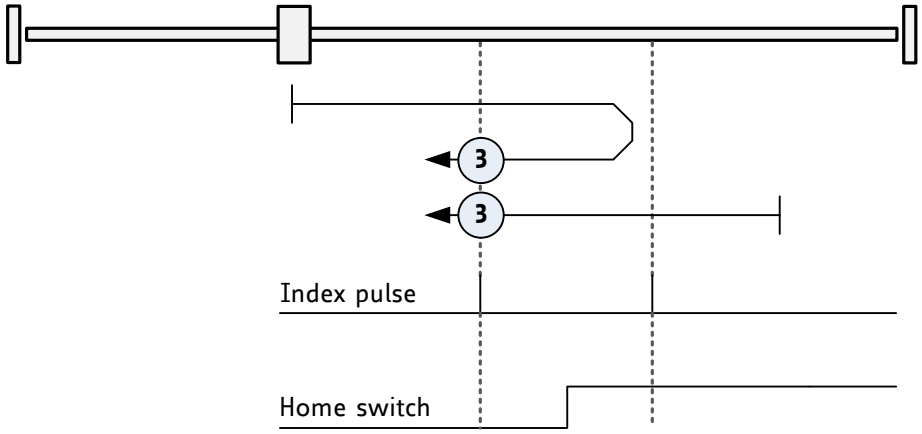


Figure 4.6 Homing on positive home switch and index pulse

METHOD 5: HOMING ON NEGATIVE HOME SWITCH AND INDEX PULSE

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

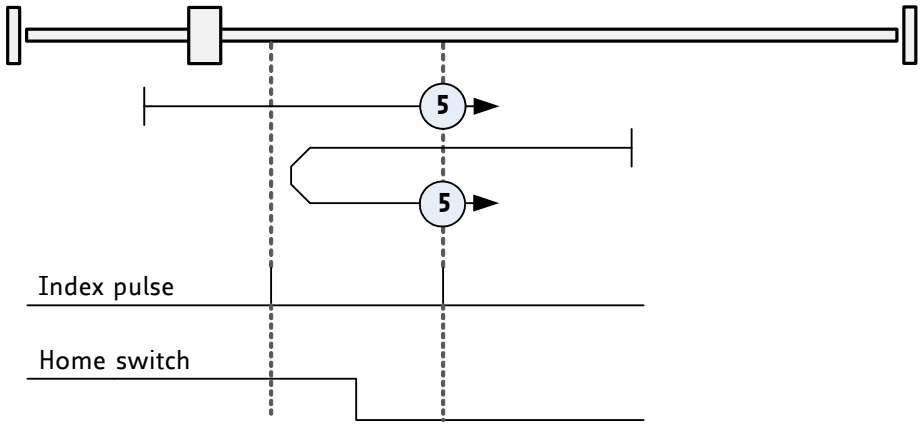


Figure 4.7 Homing on negative home switch and index pulse

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

These methods are similar to methods 1 to 5 except that the home position is not dependent on the index pulse but only dependent on the relevant home or limit switch transitions. Method 19 (similar to method 3) is shown in Figure 4.8.

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

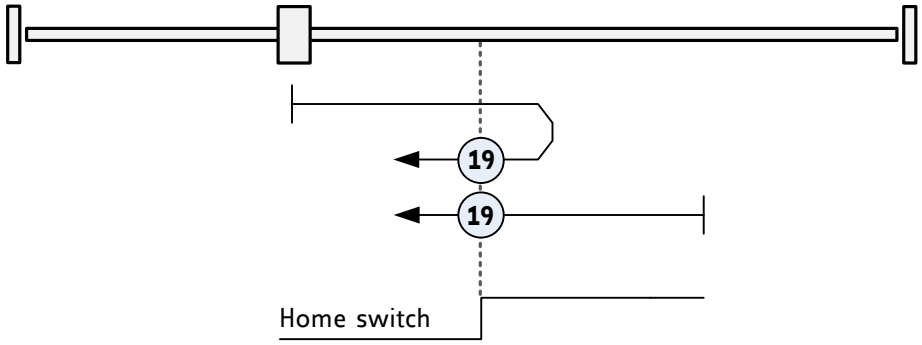


Figure 4.8 Homing without index pulse

METHOD 33 AND 34: HOMING ON NEXT INDEX PULSE

Using these methods, the direction of homing is negative or positive respectively. The home position shall be at the index pulse found in the selected direction as shown in Figure 4.9.

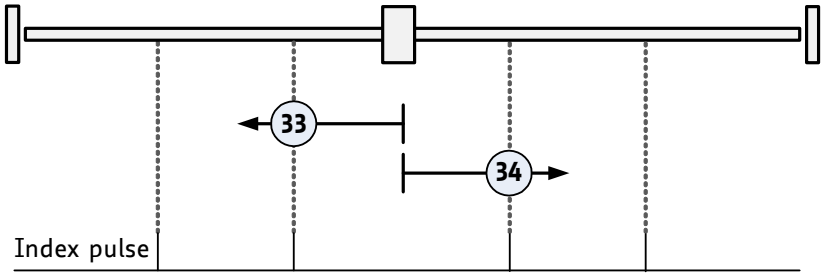


Figure 4.9 Homing on index pulse

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

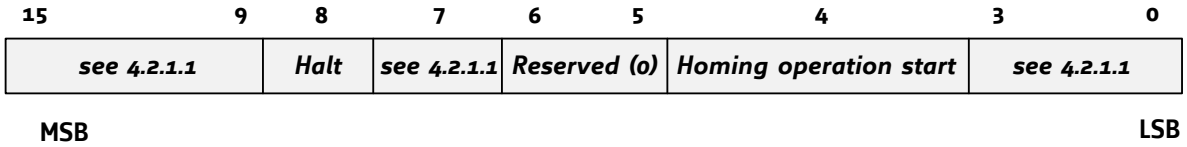
In this method, the current position shall be taken to be the home position. This method does not require the drive device to be in *operational enabled* state.

4.3.3 Detailed Object Definitions

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

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

STRUCTURE OF CONTROLWORD / VALUE DEFINITION



OPERATION MODE SPECIFIC BITS IN HM MODE

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

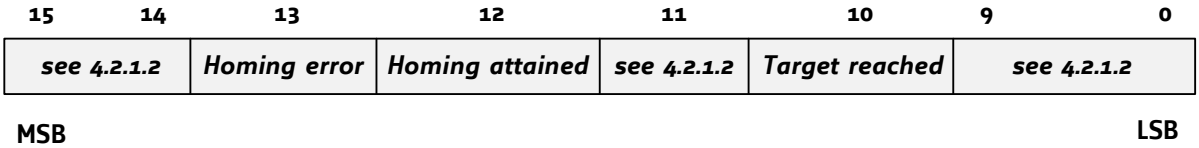
Sub-Index	Access	PDO Mapping	Value Range	Default Value
00_h	rw	Refer to CiA402-3	See command coding above	Device and operation mode specific

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

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

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

STRUCTURE OF STATUSWORD / VALUE DEFINITION



OPERATION MODE SPECIFIC BITS IN HM MODE

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

DEFINITION OF BIT 10, BIT 12, AND BIT 13

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

4.3.3.3 Object $606C_h$, $686C_h$, and $706C_h$: Velocity Actual Value

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$606C_h$ $686C_h$ $706C_h$	Velocity actual value	Variable	INTEGER32

ENTRY DESCRIPTION

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

4.3.3.4 Object $607C_h$, $687C_h$, and $707C_h$: Home Offset

These objects indicate 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* (4.3.3.5). The value of these objects is given in microsteps. Negative values indicate the opposite direction.

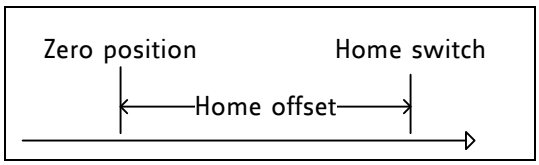


Figure 4.10 Home offset definition

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$607C_h$ (motor 0) $687C_h$ (motor 1) $707C_h$ (motor 2)	Home offset	Variable	SIGNED32

ENTRY DESCRIPTION

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

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

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

VALUE DEFINITION

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

These objects indicate the configured speeds used during homing procedure. The values are given in units, which can be selected with objects $208C_h$/$228C_h$/$248C_h$, described in paragraph 5.2.16.

Using the *homing speeds* objects, a fast and a slow homing speed can be set. In most homing modes, the home switch is searched with the fast speed first. When the home switch has been found, the motor will be decelerated to the slow speed (using the *homing acceleration*, described in paragraph 4.3.3.7) and then stopped at the exact switch point.

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

4.3.3.7 Object $609A_h$, $689A_h$, and $709A_h$: Homing Acceleration

These objects indicate the configured acceleration and deceleration to be used during homing operation. The values for accelerating to the fast homing speed and for decelerating to the slow homing speed can be set here. The values are given in units, which can be selected with objects $208E_h$/$228E_h$/$248E_h$, described in paragraph 5.2.17.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$609A_h$ (motor 0) $689A_h$ (motor 1) $709A_h$ (motor 2)	Homing acceleration	Variable	UNSIGNED32

ENTRY DESCRIPTION

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

4.3.3.8 Object 2100_h: Home Offset Display

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

4.3.4 How to Start a Homing in *hm* Mode

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

- Select *hm* mode by writing 6 to object 6060_h.
- Write 6 to object 6040_h to switch to READY_TO_SWITCH_ON state.
- Write 7 to object 6040_h to switch to SWITCHED_ON state.
- Write 15 to object 6040_h to switch to OPERATION_ENABLED state.
- Select homing method 19 by writing 19 to object 6098_h.
- Set the homing speeds by writing e.g. 1500 to object 6099_h sub index 1 and e.g. 1000 to object 6099_h sub index 2.
- Write 31 to object 6040_h to start the homing process.
- Press and release the home switch.
- When homing has finished, write 15 to object 6040_h again.

4.4 Velocity Mode

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

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

- Velocity
- Acceleration
- Deceleration
- Emergency stop

Please refer to object 6060_h (section 4.1.6) for information about how to choose an operation mode. Object 6061_h (section 0) shows the operation mode that is set.

4.4.1 Detailed Object Definitions

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

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

STRUCTURE OF CONTROLWORD / VALUE DEFINITION

15 9	8	7	6	5	4	3 0
see 4.2.1.1	*Halt*	*see 4.2.1.1*	*reference ramp*	*unlock ramp*	*enable ramp*	*see 4.2.1.1*

MSB **LSB**

There are no mode specific bits supported in vl mode.

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

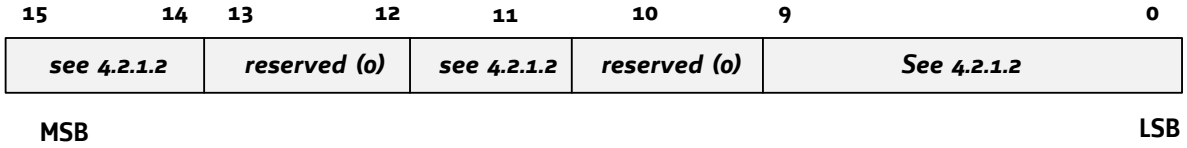
Sub-Index	Access	PDO Mapping	Value Range	Default Value
00_h	rw	Refer to CiA402-3	See command coding above	Operation mode specific

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

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

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

STRUCTURE OF STATUSWORD / VALUE DEFINITION



There are no mode specific bits supported in vl mode.

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00_h	r0	Refer to CiA402-3	See state coding above	No

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

These objects indicate the required velocity for each motor. Use them to control the velocity of the motors in velocity mode. When these objects are changed the motor will be accelerated or decelerated to the new velocity.

The range depends on the chosen units, which can be internal or user-specific. Please refer to objects $208C_h$/$228C_h$/$248C_h$ (paragraph 5.2.16) for more information about that.

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

4.4.1.4 Object 6043_h, 6843_h, and 7043_h: *vl* Velocity Demand

These objects provide the instantaneous velocities generated by the ramp function. They show the last target velocities that have been set using objects 6042_h, 6842_h, and 7042_h.

The value is given in the same unit as the *target velocity* (4.4.1.3). Please refer to objects $208C_h$/$228C_h$/$248C_h$ (5.2.16) for more information about that.

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

These objects show the actual velocities of the motors when the motors are in velocity mode. The value is given in the same unit as *vl target velocity* (4.4.1.3) and *vl velocity demand* (4.4.1.4). Please refer to objects $208C_h$/$228C_h$/$248C_h$ (paragraph 5.2.16) for more information about that.

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

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

OBJECT DESCRIPTION

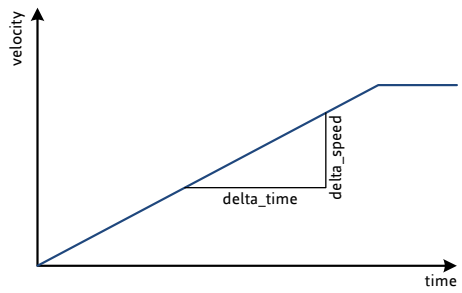
Index	Name	Object Code	Data Type
6046_h (motor 0) 6846_h (motor 1) 7046_h (motor 2)	*vl* velocity min max amount	ARRAY	UNSIGNED32

ENTRY DESCRIPTION

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

4.4.1.7 Object 6048_h, 6848_h, and 7048_h: *vl* Velocity Acceleration

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



$$vl\ velocity\ deceleration = \frac{delta\ speed}{delta\ time}$$

Figure 4.11 Transfer characteristic of the velocity acceleration

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6048_h (motor 0) 6848_h (motor 1) 7048_h (motor 2)	*vl* velocity acceleration	Record	vl acceleration deceleration

ENTRY DESCRIPTION

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

4.4.1.8 Object 6049_h, 6849_h, and 7049_h: *vl* Velocity Deceleration

These objects contain the same values as set in the objects 6048_h, 6848_h, and 7048_h. A separate declaration value is not supported.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6049_h (motor 0) 6849_h (motor 1) 7049_h (motor 2)	*vl* velocity deceleration	Record	vl acceleration deceleration

ENTRY DESCRIPTION

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

4.4.2 How to Move a Motor in *vl* Mode

Here is a little example that shows how to get a motor running in vl mode. In this little example we assume that the module has been reset (and then switched to start) 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 *vl* mode by writing 2 to object 6060_h.
- Write 6 to object 6040_h to switch to READY_TO_SWITCH_ON state.
- Write 7 to object 6040_h to switch to SWITCHED_ON state.
- Write 15 to object 6040_h to switch to OPERATION_ENABLED state.
- Write the desired target speed (e.g. 500) to object 6042_h. The motor now accelerates to that speed.
- Stop the motor by writing 0 to object 6042_h.

4.5 Profile Velocity Mode

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

THE PROFILE VELOCITY MODE COVERS THE FOLLOWING SUB-FUNCTIONS:

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

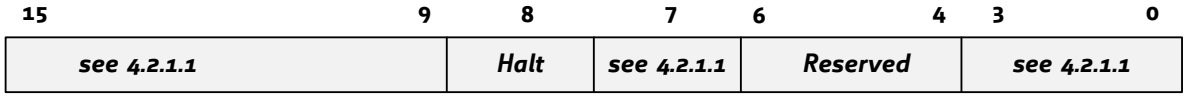
THE OPERATION OF THE REFERENCE VALUE GENERATOR AND ITS INPUT PARAMETERS INCLUDE:

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

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

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

STRUCTURE OF CONTROLWORD / VALUE DEFINITION



MSB **LSB**

There are no mode specific bits supported in this mode.

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

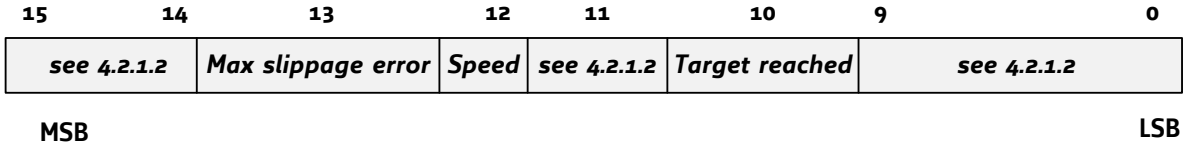
Sub-Index	Access	PDO Mapping	Value Range	Default Value
00_h	rw	Refer to CiA402-3	See command coding above	Device and operation mode specific

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

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

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

STRUCTURE OF STATUSWORD / VALUE DEFINITION



OPERATION MODE SPECIFIC BITS IN PV MODE

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

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

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

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$606C_h$ (motor 0) $686C_h$ (motor 1) $706C_h$ (motor 2)	Velocity actual value	Variable	INTEGER32

ENTRY DESCRIPTION

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

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

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

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

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

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$607D_h$ (motor 0) $687D_h$ (motor 1) $707D_h$ (motor 2)	Software position limit	Array	SIGNED32

ENTRY DESCRIPTION

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

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

These objects indicate the configured acceleration. The three *profile acceleration* objects set the maximum acceleration for each axis to be used in profile velocity mode.
Please choose the units for these objects with objects $208E_h$/$228E_h$/$248E_h$, described in paragraph 5.2.17. The objects $208E_h$/$228E_h$/$248E_h$ read 0 when internal units are selected or 179 when user units (PPS/s) are selected.

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

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

Please choose the units for these objects with objects $208E_h$/$228E_h$/$248E_h$, described in paragraph 5.2.17. The objects $208E_h$/$228E_h$/$248E_h$ read 0 when internal units are selected or 179 when user units (PPS/s) are selected.

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

The target velocity objects set the target velocity for the related motors when using profile velocity mode. They are used as input for the trajectory generator. The drive then accelerates or decelerates to that velocity using the acceleration and deceleration set by objects 6083_h/6883_h/7083_h (4.2.1.14) and 6084_h/6884_h/7084_h (4.2.1.15). The values are given in units, which can be selected with objects $208C_h$/$228C_h$/$248C_h$, described in paragraph 5.2.15.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$60FF_h$ $68FF_h$ $70FF_h$	Target velocity	Variable	SIGNED32

ENTRY DESCRIPTION

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

4.5.2 How to Move a Motor in *pv* Mode

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

- If you do not have any limit switches connected, first disable the limit switch inputs by writing 3 to object 2005_h.
- Select *pv* mode by writing 3 to object 6060_h.
- Write 6 to object 6040_h to switch to READY_TO_SWITCH_ON state.
- Write 7 to object 6040_h to switch to SWITCHED_ON state.
- Write 15 to object 6040_h to switch to OPERATION_ENABLED state.
- Write the desired target speed (e.g. 1500) to object $60FF_h$. The motor now accelerates to that speed.
- Stop the motor by writing 0 to object $60FF_h$.

5 Manufacturer Specific Area

The manufacturer segment contains manufacturer specific objects. These objects control special features of the TRINAMIC motion control device TMCM-3110.

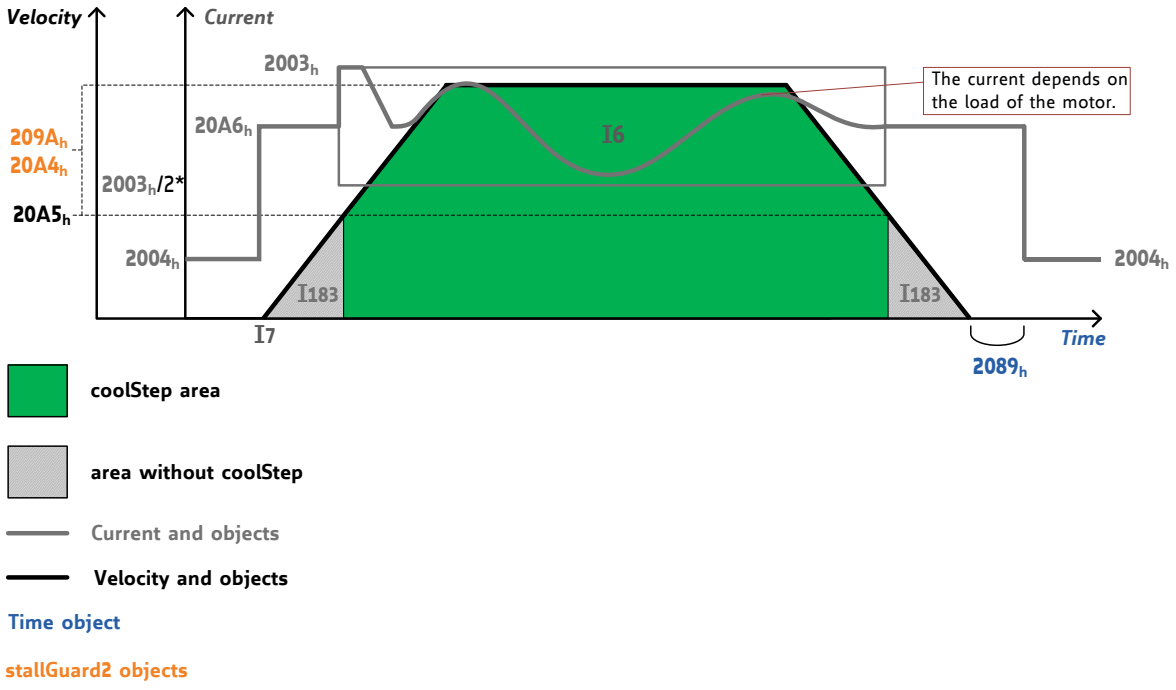
5.1 coolStep™ related Objects (Overview)

The figure below gives an overview of the coolStep related objects for motor 0. Please have in mind that the figure shows only 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 FOLLOWING ADJUSTMENTS HAVE TO BE MADE:

- Thresholds for current and velocity have to be identified and set.
- The stallGuard2 feature has to be adjusted and enabled.
- The reduction or increasing of the current in the coolStep area (depending on the load) has to be configured.

coolStep™ adjustment points and thresholds



**** The lower threshold of the coolStep current can be adjusted up to $2003_h/4$. Refer to parameter object 2098_h.***

COOLSTEP RELATED OBJECTS FOR MOTOR 0 *Note: smartEnergy is an earlier name for coolStep*		
Object	**Name**	**Description**
2003_h	Absolute max. current (CS / Current Scale)	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; e.g. 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.

COOLSTEP RELATED OBJECTS FOR MOTOR 0 *Note: smartEnergy is an earlier name for coolStep*		
Object	**Name**	**Description**
2098_h	smartEnergy current minimum (SEIMIN)	Sets the lower motor current limit for coolStep operation by scaling the CS (current Scale, see object 2003_h) value. Minimum motor current: 0 – 1/2 of CS 1 – 1/4 of CS
2099_h	smartEnergy current down step	Sets the number of stallGuard2 readings above the upper threshold necessary for each current decrement of the motor current. Number of stallGuard2 measurements per decrement: Scaling: 0… 3: 32, 8, 2, 1 0: slow decrement 3: fast decrement
$209B_h$	smartEnergy current up step	Sets the current increment step. The current becomes incremented for each measured stallGuard2 value below the lower threshold (see smartEnergy hysteresis start). current increment step size: Scaling: 0… 3: 1, 2, 4, 8 0: slow increment 3: fast increment / fast reaction to rising load
$20A6_h$	smartEnergy slow run current	Sets the motor current which is used below the threshold speed. Please adjust the threshold speed with axis parameter 182.
$209A_h$	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 (value equates 2000msec).

5.2 Detailed Object Specifications

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

These objects set the microstep resolution for each axis. A value of 6 means 64 microsteps (2^6). They are only writeable in the SWITCHED_ON_DISABLED state, but always readable.

VALUE DESCRIPTION

Value	Description
0	1 step
1	2 microsteps
2	4 microsteps
3	8 microsteps
4	16 microsteps
5	32 microsteps
6	64 microsteps
7	128 microsteps
8	256 microsteps

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

These objects show the fullstep resolutions of each axis.

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

With these objects the delay times for applying and releasing an (optional) brake for each motor can be defined. Please see also objects $200A_h$, $220A_h$, and $240A_h$ (5.2.7) for additional delay between enabling the power stage and releasing the brake. Both times are given in ms.

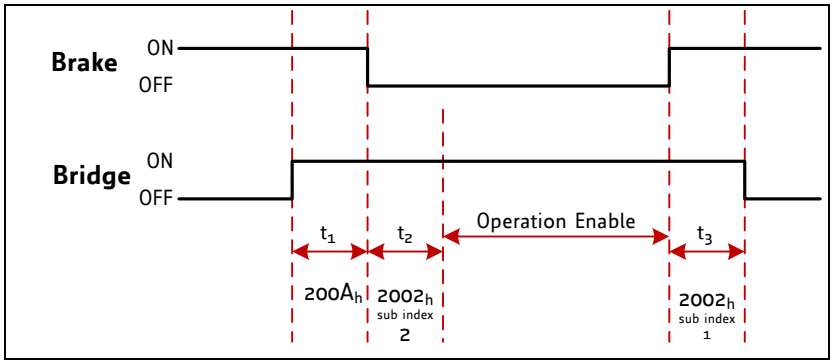


Figure 5.1 Example: brake output timing for motor 0

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

These objects define the current used when the related motor is standing (two seconds after the last move). A value of 255 means 100% of the maximum current of a drive.

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

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

The limit switches can also be inverted using bit 2 and bit 3:
- Bit 2 inverts the left limit switch
- Bit 3 inverts the right limit switch

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

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

This is an additional delay time (in milliseconds) between enabling the power stage and releasing the brake. It can be used to prevent the brake from being released too early (before the hold current in the motor has been reached). Please see also objects 2002_h, 2202_h, and 2402_h (paragraph 5.2.3).

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$200A_h$ (motor 0) $220A_h$ (motor 1) $240A_h$ (motor 2)	Enable drive delay time	Variable	UNSIGNED16

ENTRY DESCRIPTION

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

5.2.8 Object $200B_h$, $220B_h$, and $240B_h$: Encoder Parameters

These objects define the polarity of the encoder null channel for each motor. The objects are only writable in SWITCHED_ON_DISABLED state.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$200B_h$ (motor 0) $220B_h$ (motor 1) $240B_h$ (motor 2)	Encoder parameters	ARRAY	UNSIGNED8

ENTRY DESCRIPTION

Sub-Index	Description	Access	PDO Mapping	Value Range	Default Value
01_h	Null channel polarity	rw	no	UNSIGNED8	7
02_h	Direction of rotation	rw	no	0/1	0
03_h	Initialize position*)	rw	no	0/1	1

*) Write 0 to sub-index 03_h to stop automatic transfer of actual encoder position to all related registers. Per default, the encoder position is transferred automatically.

5.2.9 Object $200C_h$, $220C_h$, and $240C_h$: Brake Current Feed

These objects configure how much current has to be fed into a brake to apply and to release it. 0 means 0%, 255 means 100% of the maximum current. In most cases it is needed to feed current into the brake to release it. The default configuration is made for this case.

Setting both values to 0 disables the automatic brake control. These objects are only writable in SWITCHED_ON_DISABLED state.

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

5.2.10 Object $204E_h$, $224E_h$, and $244E_h$: Boost Current

These objects are used to set the current used for acceleration and deceleration phases for each motor. If set to 0 the current set with objects 2003_h, 2203_h, and 2403_h (5.2.4) will be used. The value 1 can be chosen for a minimum boost current setting. A value of 255 means 100% of the maximum possible current of the module.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$204E_h$ (motor 0) $224E_h$ (motor 1) $244E_h$ (motor 2)	Boost current	Variable	UNSIGNED8

ENTRY DESCRIPTION

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

5.2.11 Object 2085_h, 2285_h, and 2485_h: Ramp Divisor

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00_h	rw *not writable if user* units are selected	no	0... 13	7

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

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

The functionality of these objects depends on the unit selection for each axis (refer to objects $208C_h$/$228C_h$/$248C_h$ in chapter 5.2.16):

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

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

The functionality of these objects also depends on the unit selection (refer to objects $208E_h$/$228E_h$/$248E_h$ in chapter 5.2.17):

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

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

5.2.15 Object 2089_h, 2289_h, and 2489_h: Standby Delay

These objects have to be used for setting a standstill period before the current is changed down to standby current.

UNIT: 10msec

Index	Name	Object Code	Data Type
2089_h (motor 0) 2289_h (motor 1) 2489_h (motor 2)	Standby delay	Variable	INTEGER16

ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00_h	rw	no	1… 65535	200

5.2.16 Object $208C_h$, $228C_h$, and $248C_h$: Velocity Dimension Index

With these objects different units can be chosen:

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

This can only be changed in SWITCHED_ON_DISABLED mode.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$208C_h$ (motor 0) $228C_h$ (motor 1) $248C_h$ (motor 2)	Velocity dimension index	Variable	UNSIGNED8

ENTRY DESCRIPTION

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

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

5.2.17 Object $208E_h$, $228E_h$, and $248E_h$: Acceleration Dimension Index

With these objects, the unit for acceleration can be read out for each axis. The units can be set using objects $208C_h$/$228C_h$/$248C_h$ (5.2.16).

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$208E_h$ (motor 0) $228E_h$ (motor 1) $248E_h$ (motor 2)	Acceleration dimension index	Variable	UNSIGNED8

ENTRY DESCRIPTION

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

5.2.18 Object 2092_h, 2292_h, and 2492_h: Chopper Blank Time

These objects serve 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 drives, 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 Code	Data Type
2092_h (motor 0) 2292_h (motor 1) 2492_h (motor 2)	Chopper blank time	Variable	UNSIGNED8

ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00_h	rw	no	0... 3	2

5.2.19 Object 2093_h, 2293_h, and 2493_h: Chopper Mode

Select a chopper mode for each axis with these objects:

0 – spreadCycle™ chopper
1 – classic const. off time chopper

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2093_h (motor 0) 2293_h (motor 1) 2493_h (motor 2)	Chopper mode	Variable	UNSIGNED8

ENTRY DESCRIPTION

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

5.2.20 Object 2094_h, 2294_h, and 2494_h: Chopper Hysteresis Decrement

These objects serve for the hysteresis decrement setting of each axis. The settings determine 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 Code	Data Type
2094_h (motor 0) 2294_h (motor 1) 2494_h (motor 2)	Chopper hysteresis decrement	Variable	UNSIGNED8

ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00_h	rw	no	0... 3	0

5.2.21 Object 2095_h, 2295_h, and 2495_h: Chopper Hysteresis End

These objects provide the setting of the hysteresis end value after a number of decrements for all three motors. The decrement interval time is controlled by objects 2094_h/2294_h/2494_h (5.2.20).

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2095_h (motor 0) 2295_h (motor 1) 2495_h (motor 2)	Chopper hysteresis end	Variable	SIGNED8

ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00_h	rw	no	-3... 12	2

5.2.22 Object 2096_h, 2296_h, and 2496_h: Chopper Hysteresis Start

These objects provide the hysteresis start setting for each motor. Note that the *chopper hysteresis start* value is an offset to the *hysteresis end value* (5.2.21).

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2096_h (motor 0) 2296_h (motor 1) 2496_h (motor 2)	Chopper hysteresis start	Variable	UNSIGNED8

ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00_h	rw	no	0... 8	3

5.2.23 Object 2097_h, 2297_h, and 2497_h: Chopper Off Time

The off time setting controls the minimum chopper frequency. An off time within the range of 5µs to 20µs will fit.

Off time setting for constant t_{OFF} chopper: N_{CLK}= 12 + 32*t_{OFF} (Minimum is 64 clocks)

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2097_h (motor 0) 2297_h (motor 1) 2497_h (motor 2)	Chopper off time	Variable	UNSIGNED8

ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00_h	rw	no	0 / 2... 15	5

5.2.24 Object 2098_h, 2298_h, and 2498_h: Smart Energy Current Minimum

These objects provide the setting of the lower motor current limit for coolStep operation by scaling the CS value for each axis.

Minimum motor current:
0 – 1/2 of CS
1 – 1/4 of CS

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2098_h (motor 0) 2298_h (motor 1) 2498_h (motor 2)	Smart energy current minimum	Variable	UNSIGNED8

ENTRY DESCRIPTION

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

5.2.25 Object 2099_h, 2299_h, and 2499_h: Smart Energy Current down Step

These objects provide the setting of the number of stallGuard2 readings above the upper threshold necessary for each current decrement of the motor current.

NUMBER OF STALLGUARD2 MEASUREMENTS PER DECREMENT:

Scaling: 0… 3 – 32, 8, 2, 1
0 – slow decrement
3 – fast decrement

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2099_h (motor 0) 2299_h (motor 1) 2499_h (motor 2)	Smart energy current down step	Variable	UNSIGNED8

ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00_h	rw	no	0… 3	0

5.2.26 Object $209A_h$, $229A_h$, and $249A_h$: Smart Energy Hysteresis

These objects set the distance between the lower and the upper threshold for stallGuard2 reading for each axis. Above the upper threshold the motor current becomes decreased.

Hysteresis
(smartEnergy hysteresis value + 1) * 32

Upper stallGuard2 threshold:
(smartEnergy hysteresis start + smartEnergy hysteresis + 1) * 32

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$209A_h$ (motor 0) $229A_h$ (motor 1) $249A_h$ (motor 2)	Smart energy hysteresis	Variable	UNSIGNED8

ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00_h	rw	no	0… 15	0

5.2.27 Object $209B_h$, $229B_h$, and $249B_h$: Smart Energy Current up Step

These objects set the current increment step. The current becomes incremented for each measured stallGuard2 value below the lower threshold (see *smart energy hysteresis start* $209C_h$/$229C_h$/$249C_h$ chapter 5.2.28).

Current increment step size:

Scaling: 0... 3 – 1, 2, 4, 8
0 – slow increment
3 – fast increment / fast reaction to rising load

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$209B_h$ (motor 0) $229B_h$ (motor 1) $249B_h$ (motor 2)	Smart energy current up step	Variable	UNSIGNED8

ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00_h	rw	no	0... 3	0

5.2.28 Object $209C_h$, $229C_h$, and $249C_h$: Smart Energy Hysteresis Start

These objects serve to set the lower threshold for the stallGuard2 value (see *smart Energy current up step* $209B_h$/$229B_h$/$249B_h$ chapter 5.2.27).

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$209C_h$ (motor 0) $229C_h$ (motor 1) $249C_h$ (motor 2)	Smart energy hysteresis start	Variable	UNSIGNED8

ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00_h	rw	no	0... 15	0

5.2.29 Object $209D_h$, $229D_h$, and $249D_h$: stallGuard2 Filter Enable

These objects are used to set the stallGuard2 filter for more precision of the measurement. If set, reduces the measurement frequency to one measurement per four fullsteps.

In most cases it is expedient to set the filtered mode before using coolStep. Choose the standard mode for step loss detection.

0 – standard mode
1 – filtered mode

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$209D_h$ (motor 0) $229D_h$ (motor 1) $249D_h$ (motor 2)	stallGuard2 filter enable	Variable	UNSIGNED8

ENTRY DESCRIPTION

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

5.2.30 Object $209E_h$, $229E_h$, and $249E_h$: stallGuard2 Threshold

These signed values control the stallGuard2 threshold level for stall output and set 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.

Value	Description
0	Indifferent value
1… 63	less sensitivity
-1… -64	higher sensitivity

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$209E_h$ (motor 0) $229E_h$ (motor 1) $249E_h$ (motor 2)	stallGuard2 threshold	Variable	SIGNED8

ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00_h	rw	no	-64… 63	5

5.2.31 Object $209F_h$, $229F_h$, and $249F_h$: Slope Control High Side

These objects determine the slope of the motor driver outputs.

0 – lowest slope
3 – fastest slope

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$209F_h$ (motor 0) $229F_h$ (motor 1) $249F_h$ (motor 2)	Slope control high side	Variable	UNSIGNED8

ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00_h	rw	no	0… 3	3

5.2.32 Object $20A0_h$, $22A0_h$, and $24A0_h$: Slope Control Low Side

These objects determine the slope of the motor driver outputs. Set it identical to *slope control high side* objects $209F_h$/$229F_h$/$249F_h$ (5.2.31).

0 – lowest slope
3 – fastest slope

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$20A0_h$ (motor 0) $22A0_h$ (motor 1) $24A0_h$ (motor 2)	Slope control low side	Variable	UNSIGNED8

ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00_h	rw	no	0… 3	3

5.2.33 Object $20A1_h$, $22A1_h$, and $24A1_h$: Short Protection Disable

These objects are used to enable or to disable the short to ground protection for each axis. *Use default value!*

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$20A1_h$ (motor 0) $22A1_h$ (motor 1) $24A1_h$ (motor 2)	Short protection disable	Variable	UNSIGNED8

ENTRY DESCRIPTION

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

5.2.34 Object $20A2_h$, $22A2_h$, and $24A2_h$: Short Detection Timer

These objects provide the timing of the short detection for each axis. *Use default value!*

Value	Timing
0	3.2µs
1	1.6µs
2	1.2µs
3	0.8µs

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$20A2_h$ (motor 0) $22A2_h$ (motor 1) $24A2_h$ (motor 2)	Short detection timer	Variable	UNSIGNED8

ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00_h	rw	no	0… 3	0

5.2.35 Object $20A3_h$, $22A3_h$, and $24A3_h$: Vsense

These objects are used for setting the sense resistor voltage based current scaling for each axis. *Use the default value and change only when recommended by TRINAMIC!*

0 – Full scale sense resistor voltage is 1/18 VDD
1 – Full scale sense resistor voltage is 1/36 VDD
(This refers to a current setting of 31 and DAC value 255.)

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$20A3_h$ (motor 0) $22A3_h$ (motor 1) $24A3_h$ (motor 2)	Vsense	Variable	UNSIGNED8

ENTRY DESCRIPTION

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

5.2.36 Object $20A4_h$, $22A4_h$, and $24A4_h$: Stop on Stall

Below this speed the related motor will not be stopped. Above this speed the related motor will stop in case the stallGuard2 load value reaches zero.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$20A4_h$ (motor 0) $22A4_h$ (motor 1) $24A4_h$ (motor 2)	Stop on stall	Variable	UNSIGNED8

ENTRY DESCRIPTION

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

5.2.37 Object $20A5_h$, $22A5_h$, and $24A5_h$: Smart Energy Threshold Speed

Above this speed coolStep becomes enabled.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$20A5_h$ (motor 0) $22A5_h$ (motor 1) $24A5_h$ (motor 2)	Smart energy threshold speed	Variable	UNSIGNED16

ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00_h	rw	no	0... 2147483647 [pps]	0

5.2.38 Object $20A6_h$, $22A6_h$, and $24A6_h$: Smart Energy Slow Run Current

These objects can be used for setting the motor current below the threshold speed.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$20A6_h$ (motor 0) $22A6_h$ (motor 1) $24A6_h$ (motor 2)	Smart energy slow run current	Variable	UNSIGNED8

ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00_h	rw	no	0... 255 $\left[\frac{\text{max.module current}}{255}\right]$	0

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

These objects show the *home offset* (4.3.3.4). The value is given in microsteps.

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00_h	ro	no	Refer to *home offset* (4.3.3.4)	0

5.2.40 Object 2101_h, 2301_h, and 2501_h: Actual stallGuard2 Load Value

These objects show the actual stallGuard2 load value for the motors. They are needed for configuring stallGuard2 and for finding a fitting velocity for each motor.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2101_h (motor 0) 2301_h (motor 1) 2501_h (motor 2)	Actual stallGuard2™ load value	Variable	UNSIGNED8

ENTRY DESCRIPTION

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

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

These objects show the hardware error flags of the motor driver IC for each motor axis.

THERE ARE THE FOLLOWING ERROR BITS:

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

The pulse divisor display shows the pulse divisor that has been calculated by a drive when user units are selected or that can be set using objects 2086_h/2286_h/2486_h (5.2.12) when internal units are selected.

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

These objects show the values of the objects 2087_h/2287_h/2487_h (5.2.13).

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

The ramp divisor display shows the ramp divisor that has been calculated by the drive when user units are selected or that can be set using objects 2085_h/2285_h/2485_h (5.2.11) when internal units are selected.

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

These objects show the values of objects 2088_h/2288_h/2488_h (5.2.14).

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

5.2.47 Object 2700_h: TMCL Direct Communication

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00_h	ro	no	Make signature $656b616d_h$	0

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

5.2.48 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.

read:	0 = manufacturer specific mode is inactive 1 = manufacturer specific mode is active
write:	$656b616d_h$ = make signature $6c6c696b_h$ = kill signature

THE MANUFACTURER SPECIFIC MODE HAS THE FOLLOWING FEATURES:

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

5.2.49 Object 2702_h: Digital Inputs

Bits 16... 23 of this object reflect the states of the digital inputs of the module.

Bit	Input
16	IN_0
17	IN_1
18	IN_2
19	IN_3
etc.	

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

5.2.50 Object 2703_h: Digital Outputs

With this object the digital outputs (general purpose outputs) can be set. Bits 16... 23 of sub index 1 switch outputs OUT_0 and OUT_1 of the module. Bits 16... 23 of sub index 2 determine which outputs can be switched.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2703_h	Digital outputs	Array	

ENTRY DESCRIPTION

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

5.2.51 Object 2704_h: CAN Bit Rate

With this object it is possible to change the CAN bit rate. To do this, first write the new value to this object. Then, store the new setting by writing the save signature to object 2706_h (5.2.53). After that, reset the module. The new setting becomes active now.

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

HOW TO CHANGE THE CAN BIT RATE:

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

5.2.52 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 (5.2.53). After that, reset the module. The new setting becomes active now.

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

HOW TO CHANGE THE NODE ID:

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

5.2.53 Object 2706_h: Store

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

Save signature:	65766173_h

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2706_h	Store	Variable	UNSIGNED32

ENTRY DESCRIPTION

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

5.2.54 Object 2707_h: CAN Bit Rate Load

This object shows the selected CAN bit rate.

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

5.2.55 Object 2708_h: Node ID Load

This object shows the selected node ID.

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

5.2.56 Object $270E_h$: Device Analogue Inputs

This object defines the analog inputs that can be used. The TMCM-3110 supports two analog inputs, AIN0 and AIN4. They can be accessed using sub-index 01_h and subindex 05_h.

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$270E_h$	Device analogue inputs	ARRAY	UNSIGNED32

ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00_h	Highest sub-index supported	ro	-	8
01_h	Analogue input 1	ro	0... 4095	no
02_h	Analogue input 2	ro	0... 4095	no
03_h	Analogue input 3	ro	0... 4095	no
04_h	Analogue input 4	ro	0... 4095	no
05_h	Analogue input 5	ro	0... 4095	no
06_h	Analogue input 6	ro	0... 4095	no
07_h	Analogue input 7	ro	0... 4095	no
08_h	Analogue input 8	ro	0... 4095	no

6 Emergency Messages

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

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

ERROR CODES FOR TMCM-3110

Error code	Add. byte					Description
	1	2	3	4	5	
0x0000	0	0/1/2	0	0	0	**Fault reset** The fault reset command has been executed.
0x1000	1	0/1/2	0	0	0	**Generic error: open load bridge A** The motor driver indicates open load on bridge A. It is possible that the motor cable is broken or that there is an error in the power amplifier itself.
0x1000	2	0/1/2	0	0	0	**Generic error: open load bridge B** The motor driver indicates open load on bridge B. It is possible that the motor cable is broken or that there is an error in the power amplifier itself.
0x2310	0	0/1/2	0	0	0	**Overcurrent high side** The motor driver indicates an overcurrent on the high side. This can be caused by a short circuit in the driver stage.
0x2311	0	0/1/2	0	0	0	**Overcurrent bridge B** The motor driver indicates that there is overcurrent on bridge B. This can be caused by a short circuit in the motor itself or in the motor driver stage.
0x2312	0	0/1/2	0	0	0	**Overcurrent bridge A** The motor driver indicates that there is overcurrent on bridge A. This can be caused by a short circuit in the motor itself or in the motor driver stage.
0x3230	0	0/1/2	0	0	0	**stallGuard2™ error** The actual load value exceeds the stallGuard2 limit.
0x4310	1	0/1/2	0	0	0	**Overtemperature pre-warning** The temperature in the motor driver exceeds the pre-warning limit.
0x4310	2	0/1/2	0	0	0	**Overtemperature error** The motor driver has been switched off because the temperature limit has been exceeded.
0x5441	0	255	0	0	0	**Shutdown switch active** The enable signal is missing (due to the shutdown switch) and the motor driver has been switched off.
0x6320	0	255	0	0	0	**Parameter error** The data in the received PDO is either wrong or cannot be accepted due to the internal state of the drive.
0x8110	1	255	0	0	0	**CAN controller overflow** The receive message buffer of the CAN controller hardware is full and some CAN messages are lost.
0x81110	2	255	0	0	0	**CAN Tx buffer overflow** The software CAN transmit buffer is full and thus some CAN messages are lost.
0x81110	3	255	0	0	0	**CAN Rx buffer overflow** The software CAN receive buffer is full and so some CAN messages are lost.
0x8120	0	255	0	0	0	**CAN error passive** The CAN controller has detected communication errors and has entered the CAN Error passive state.
0x8140	0	255	0	0	0	**CAN controller recovered from bus-off state** The CAN controller had detected too many errors and had changed into the bus-off state. The drive has been stopped and disabled. This message is sent after the CAN controller has recovered from bus-off state and is bus-on again.
0x8611	0	0/1/2	0	0	0	**Following error** The deviation between motor position counter and encoder position counter has exceeded the following error window.
0xff00	0	0/1/2	0	0	0	**Undervoltage** The supply voltage is too low to drive a motor.
0xff01	1	0/1/2	0	0	0	**Positive software limit** The actual position is outside the range defined by object 0x607d.

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

7 Life Support Policy

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

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

8 Revision History

8.1 Firmware Revision

Version	Date	Author	Description
3.16	2013-APR-18	OK	Initial version for TMCM-3110

Table 8.1 Firmware revision

8.2 Document Revision

Version	Date	Author	Description
1.00	2013-JUN-20	SD	Initial version

Table 8.2 Document revision

9 References

[TMCM-3110] TMCM-3110 hardware manual