

CANopen Firmware Version V3.20

CANopen MANUAL

+



+

TMCM-142-IF

1-Axis Controller/Driver Module
CAN Interface
+18V... 78V DC / 5A RMS
Encoder Feedback

+



+

PD-146-60-SE

Mechatronic Device
Controller/Driver
2-Phase NEMA24 Stepper Motor
Up to 3.1Nm Holding Torque
CAN Interface
24V DC / 2.8A RMS
Integrated Encoder

TRINAMIC Motion Control GmbH & Co. KG
Hamburg, Germany

www.trinamic.com

Table of Contents

1	Preface	5
1.1	General Features of the CANopen Implementation	5
1.2	Abbreviations	6
1.3	Firmware Update	6
2	Communication	7
2.1	Reference Model.....	7
2.2	NMT State Machine.....	8
2.3	Device Model.....	10
2.4	Object Dictionary	11
3	Communication Objects	12
3.1	Detailed Object Specifications	12
3.1.1	Object 1000 _h : Device Type.....	12
3.1.2	Object 1001 _h : Error Register.....	12
3.1.3	Object 1005 _h : COB-ID SYNC Message	13
3.1.4	Object 1008 _h : Manufacturer Device Name.....	13
3.1.5	Object 1009 _h : Manufacturer Hardware Version	13
3.1.6	Object 100A _h : Manufacturer Software Version	13
3.1.7	Object 100C _h : Guard Time	14
3.1.8	Object 100D _h : Life Time Factor	14
3.1.9	Object 1010 _h : Store Parameters	14
3.1.10	Object 1011 _h : Restore Default Parameters.....	15
3.1.11	Object 1014 _h : COB-ID Emergency Object	17
3.1.12	Object 1015 _h : Inhibit Time EMCY.....	17
3.1.13	Object 1016 _h : Consumer Heartbeat Time.....	17
3.1.14	Object 1017 _h : Producer Heartbeat Time	17
3.1.15	Object 1018 _h : Identity Object	18
3.1.16	Object 1023 _h : OS Command.....	18
3.1.17	Object 1029 _h : Error Behavior	18
3.1.18	Objects 1400 _h -1403 _h : Receive PDO Communication Parameter	19
3.1.19	Objects 1600 _h -1603 _h : Receive PDO Mapping.....	19
3.1.20	Objects 1800 _h -1803 _h : Transmit PDO Communication Parameter.....	20
3.1.21	Objects 1A00 _h -1A03 _h : Transmit PDO Mapping Parameter.....	21
4	Device Profile Objects (CiA402) and Modes of Operation	22
4.1	Detailed Object Specifications	22
4.1.1	Object 605A _h : Quick Stop Option Code	22
4.1.2	Object 605B _h : Shutdown Option Code	23
4.1.3	Object 605C _h : Disable Operation Option Code.....	23
4.1.4	Object 605D _h : Halt Option Code.....	23
4.1.5	Object 605E _h : Fault Reaction Option Code	23
4.1.6	Object 6060 _h : Modes of Operation.....	24
4.1.7	Object 6061 _h : Modes of Operation Display	24
4.1.8	Object 606A _h : Sensor Selection Code	24
4.1.9	Object 608F _h : Position Encoder Resolution.....	25
4.1.10	Object 60FD _h : Digital Inputs.....	25
4.1.11	Object 6502 _h : Supported Drive Modes.....	25
4.2	Profile Position Mode.....	27
4.2.1	Detailed Object Specifications	27
4.2.1.1	Object 6040 _h : Controlword.....	28
4.2.1.2	Object 6041 _h : Statusword.....	29
4.2.1.3	Object 6062 _h : Position Demand Value.....	30
4.2.1.4	Object 6063 _h : Position Actual Internal Value	30
4.2.1.5	Object 6064 _h : Position Actual Value.....	30
4.2.1.6	Object 6065 _h : Following Error Window	31
4.2.1.7	Object 6067 _h : Position Window	31
4.2.1.8	Object 6068 _h : Position Window Time.....	31
4.2.1.9	Object 60A4 _h : Profile Jerk	32

4.2.1.10	Object 606C _h : Velocity Actual Value	32
4.2.1.11	Object 607A _h : Target Position	32
4.2.1.12	Object 607D _h : Software Position Limit.....	33
4.2.1.13	Object 6081 _h : Profile Velocity.....	33
4.2.1.14	Object 6083 _h : Profile Acceleration.....	33
4.2.1.15	Object 6084 _h : Profile Deceleration	34
4.2.1.16	Object 6085 _h : Quick Stop Deceleration.....	34
4.2.1.17	Object 6086 _h : Motion Profile Type.....	34
4.2.2	How to Move a Motor in <i>pp</i> Mode	35
4.3	Homing Mode	36
4.3.1	Connecting Home Switch, Left Switch, and Right Switch.....	37
4.3.1.1	TMC-142-IF.....	37
4.3.1.2	PDx-146-60-SE.....	38
4.3.2	Homing Methods	39
4.3.3	Detailed Object Definitions.....	42
4.3.3.1	Object 6040 _h : Controlword.....	42
4.3.4	Object 6041 _h : Statusword	43
4.3.4.1	Object 606C _h : Velocity Actual Value	43
4.3.4.2	Object 607C _h : Home Offset.....	44
4.3.4.3	Object 6098 _h : Homing Method.....	44
4.3.4.4	Object 6099 _h : Homing Speeds	45
4.3.4.5	Object 609A _h : Homing Acceleration	45
4.3.4.6	Object 2100 _h : Home Offset Display	45
4.3.5	How to Start a Homing in <i>hm</i> Mode	46
4.4	Profile Velocity Mode.....	47
4.4.1	Detailed Object Definitions.....	47
4.4.1.1	Object 6040 _h : Controlword.....	47
4.4.1.2	Object 6041 _h : Statusword	48
4.4.1.3	Object 6062 _h : Position Demand Value.....	48
4.4.1.4	Object 6063 _h : Position Actual Internal Value	48
4.4.1.5	Object 6064 _h : Position Actual Value.....	49
4.4.1.6	Object 6065 _h : Following Error Window	49
4.4.1.7	Object 606C _h : Velocity Actual Value	49
4.4.1.8	Object 607D _h : Software Position Limit.....	50
4.4.1.9	Object 6083 _h : Profile Acceleration.....	50
4.4.1.10	Object 6084 _h : Profile Deceleration	50
4.4.1.11	Object 6085 _h : Quick Stop Deceleration.....	51
4.4.1.12	Object 60FF _h : Target Velocity	51
4.4.2	How to Move a Motor in <i>pv</i> Mode.....	51
5	Manufacturer Specific Area	52
5.1	Detailed Object Specifications	52
5.1.1	Object 2000 _h : Microstep Resolution.....	52
5.1.2	Object 2001 _h : Fullstep Resolution	52
5.1.3	Object 2002 _h : Brake Delay Times.....	53
5.1.4	Object 2003 _h : Maximum Current.....	53
5.1.5	Object 2004 _h : Standby Current.....	53
5.1.6	Object 2005 _h : Limit Switches.....	54
5.1.7	Object 200A _h : Enable Drive Delay Time	54
5.1.8	Object 200B _h : Encoder Parameters	55
5.1.9	Object 200C _h : Brake Current Feed	55
5.1.10	Object 2089 _h : Standby Delay	55
5.1.11	Object 208A _h : Mixed Decay Run.....	56
5.1.12	Object 208B _h : Mixed Decay Standby.....	56
5.1.13	Object 208C _h : Velocity Dimension Index	56
5.1.14	Object 208E _h : Acceleration Dimension Index	56
5.1.15	Object 2100 _h : Home Offset Display.....	57
5.1.16	Object 2102 _h : Driver Error Flags	57
5.1.17	Object 2107 _h : Microstep Resolution Display	57

5.1.18	Object 2108 _h : Absolute Resolver Value	57
5.1.19	Object 2700 _h : TMCL Direct Communication	58
5.1.20	Object 2701 _h : Manufacturer Specific Mode	58
5.1.21	Object 2702 _h : Digital inputs	58
5.1.22	Object 2703 _h : Digital Outputs	59
5.1.23	Object 2704 _h : CAN Bit Rate	59
5.1.24	Object 2705 _h : Node ID	59
5.1.25	Object 2706 _h : Store	60
5.1.26	Object 2707 _h : CAN Bit Rate Load	60
5.1.27	Object 2708 _h : Node ID load	60
5.1.28	Object 270E _h : Device Analog Inputs	61
5.1.1	Object 2710 _h : Pull-up Resistors	61
6	Emergency messages	62
7	Life Support Policy	64
8	Revision History	65
8.1	Firmware Revision	65
8.2	Document Revision	66
9	References	67

Table of Figures

Figure 2.1	Overview CANopen NMT state machine	8
Figure 2.2	Communication architecture	9
Figure 2.3	Device model	10
Figure 3.1	Restore procedure	16
Figure 4.1	Finite state machine	27
Figure 4.2	Homing mode function	36
Figure 4.3	Connectors of TMC-142-IF	37
Figure 4.4	Connectors of PDx-146-60-SE	38
Figure 4.5	Homing on negative switch and index pulse	39
Figure 4.6	Homing on positive limit switch and index pulse	39
Figure 4.7	Homing on positive home switch and index pulse	40
Figure 4.8	Homing on negative home switch and index pulse	40
Figure 4.9	Homing without index pulse	41
Figure 4.10	Homing on index pulse	41
Figure 4.11	Home offset definition	44
Figure 5.1	Brake output timing	53

1 Preface

This document specifies objects and modes of operation of the TRINAMIC TMCM-142-IF and PD-146-60-SE with CANopen firmware. As the TMCM-142-IF and the TMCM-146 are stepper motor controller and driver modules the use of the CiA DS402 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 a short introduction or rather a short overview and afterwards the information will be more in detail.

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

1.1 General Features of the CANopen Implementation

MAIN CHARACTERISTICS

Communication according to standard CiA-301 V4.1

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

CAN ID: 11 bit

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

NMT services: NMT slave

SDO communication

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

PDO communication

- Producer
- Consumer
- *RPDOs*
 - 1, 2, 3, 4, 6
 - 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
 - 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
 Heartbeat: consumer and producer

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 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.

2 Communication

2.1 Reference Model

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

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

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

SERVICE PRIMITIVES	
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 consists of the following states:

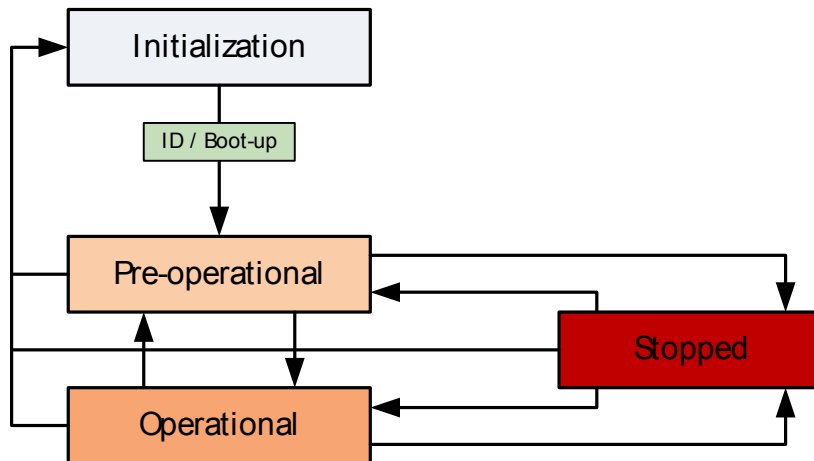


Figure 2.1 Overview CANopen NMT state machine

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 is disabled in this state, the device can communicate via SDO.

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

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

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

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

The figure below shows the situation of the state machine in this device profile.

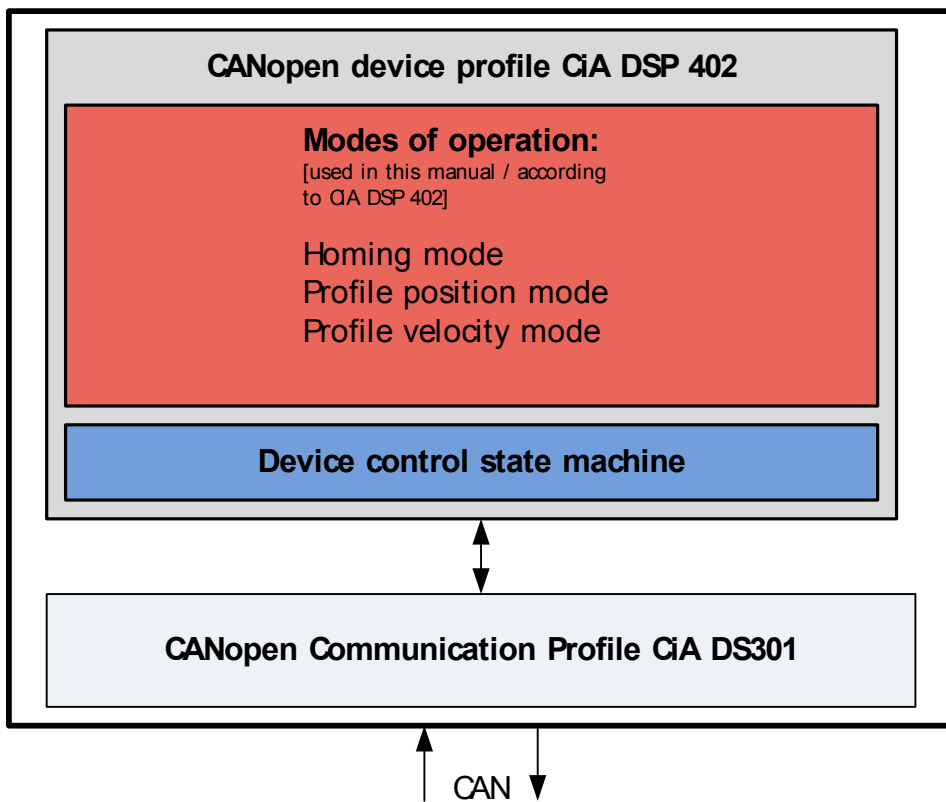


Figure 2.2 Communication architecture

2.3 Device Model

A device is structured like the following:

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

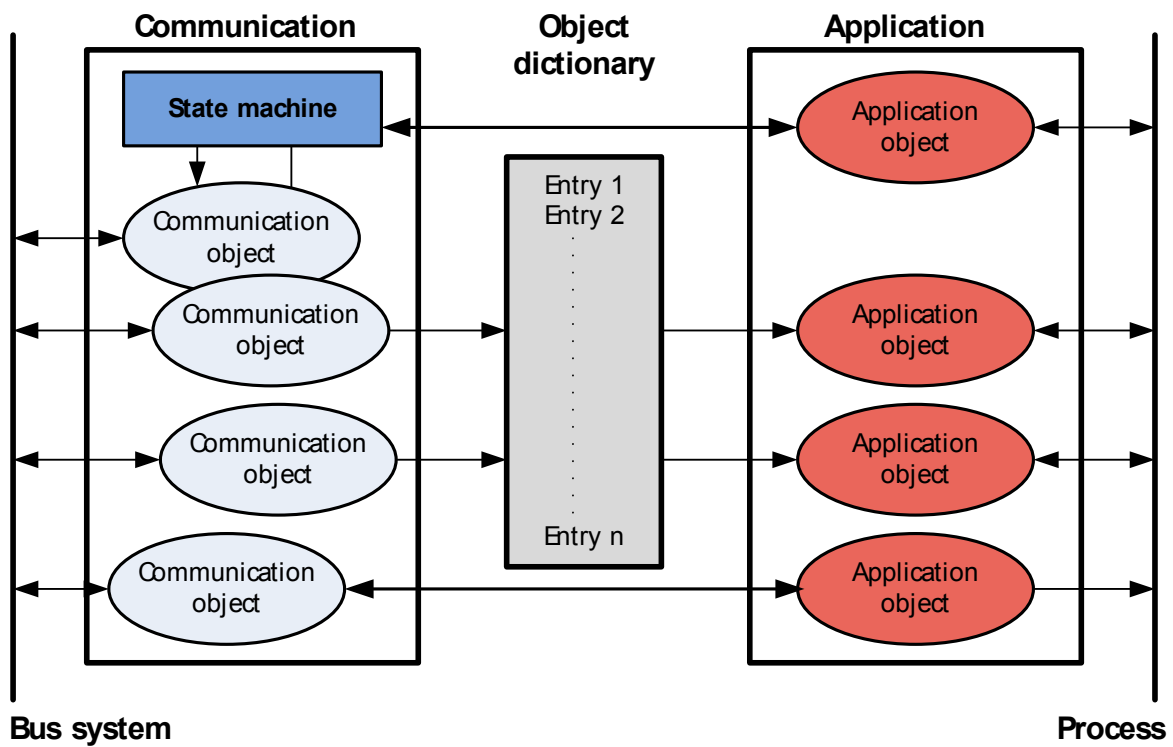


Figure 2.3 Device model

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 shown below:

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

Table 2.3 Object Dictionary

- *¹ The communication profile area at indices 1000_h through 1FFF_h contains the communication specific parameters for the CAN network. These entries are common to all devices.
- *² The manufacturer segment at indices 2000_h through 5FFF_h contains manufacturer specific objects. These objects control the special features of the TRINAMIC motion control devices TMCM-142 and PD-146.
- *³ The standardized device profile area at indices 6000_h through 9FFF_h contains all data objects common to a class of devices that can be read or written via the network. The device profiles use entries from 6000_h to 9FFF_h to describe the device parameters and the device functionality.

3 Communication Objects

3.1 Detailed Object Specifications

3.1.1 Object 1000_h: Device Type

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

3.1.2 Object 1001_h: Error Register

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

STRUCTURE OF THE ERROR REGISTER

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

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

3.1.3 Object 1005_h: COB-ID SYNC Message

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

3.1.4 Object 1008_h: Manufacturer Device Name

This object contains the manufacturer device name.

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 _h	Const.	no	no	TMC-142 or PD-146

3.1.5 Object 1009_h: Manufacturer Hardware Version

This object contains the hardware version description.

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 _h	Const.	no	no	<i>depends on device, e.g. 1.0</i>

3.1.6 Object 100A_h: Manufacturer Software Version

This object contains the software version description.

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

3.1.7 Object 100C_h: Guard Time

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
100C _h	Guard time	Variable	UNSIGNED16

ENTRY DESCRIPTION

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

3.1.8 Object 100D_h: Life Time Factor

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
100D _h	Life time factor	Variable	UNSIGNED8

ENTRY DESCRIPTION

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

3.1.9 Object 1010_h: Store Parameters

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

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

SEVERAL PARAMETER GROUPS ARE DISTINGUISHED:

- Sub-index 0_h: contains the largest sub-index that is supported.
- Sub-index 1_h: saves all parameters.
- Sub-index 2_h: saves communication parameters 100C_h, 100D_h, 1015_h, 1017_h, and 1029_h.
- Sub-index 4_h: saves motor axis 1 parameters (6040_h... 6084_h)
- Sub-index 7_h: saves device parameters / TRINAMIC specific parameters (2000_h... 270E_h).

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

Storage write access structure

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

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

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

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

Storage read access structure

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

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

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

Sub-Index	Description	Access	PDO Mapping	Value Range	Default Value
00 _h	Largest sub-index supported	ro	no	1 _h -7F _h	no
01 _h	Save all parameters	rw	no	UNSIGNED32	no
02 _h	Save special communication parameters	rw	no	UNSIGNED32	no
04 _h	Save motor axis 1 parameters	rw	no	UNSIGNED32	no
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 (6040_h... 6084_h)
- 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 1_h – 7F_h, reset communication for sub-index 2_h) or power cycled.

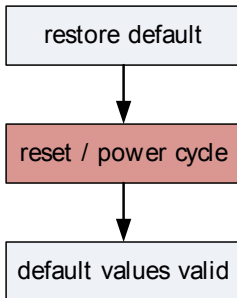


Figure 3.1 Restore procedure

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	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
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 286_h.
- The manufacturer-specific Product code (sub-index 2_h) identifies a specific device version.
- The Manufacturer-specific Revision number (sub-index 3_h) consists of a major revision number and a minor revision number.

OBJECT DESCRIPTION

Index	Name	Object 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	ro	no	UNSIGNED32	reads 0x0286
02 _h	Product code	ro	no	UNSIGNED32	depends on the device e.g. 142 or 146
03 _h	Revision number	ro	no	UNSIGNED32	firmware revision number; reads e.g. 0x20003 for version 2.3

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-1403_h: Receive PDO Communication Parameter

This object contains the communication parameters for the PDOs 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 -1403 _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

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
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

3.1.19 Objects 1600_h-1603_h: Receive PDO Mapping

This object contains the mapping for the PDOs 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
1600 _h -1603 _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

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
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
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
03 _h	Mapping entry 3	rw	no	UNSIGNED32	Index 1600 _h : 0 Index 1601 _h : 0 Index 1602 _h : 0 Index 1603 _h : 0

3.1.20 Objects 1800_h-1803_h: Transmit PDO Communication Parameter

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

OBJECT DESCRIPTION

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

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
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
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

3.1.21 Objects 1A00_h-1A03_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 -1A03 _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

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
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
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
03 _h	Mapping entry 3	rw	no	UNSIGNED32	Index 1A00 _h : 0 Index 1A01 _h : 0 Index 1A02 _h : 0 Index 1A03 _h : 0

4 Device Profile Objects (CiA402) and Modes of Operation

The PDS 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
- 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: Quick Stop Option Code

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

VALUE DEFINITION

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
605A _h	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: Shutdown Option Code

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

VALUE DEFINITION

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

4.1.3 Object 605C_h: Disable Operation Option Code

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

VALUE DEFINITION

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

4.1.4 Object 605D_h: Halt Option Code

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

VALUE DEFINITION

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

4.1.5 Object 605E_h: Fault Reaction Option Code

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

VALUE DEFINITION

Value	Definition
2	Slow down on <i>quick stop ramp</i>

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

4.1.6 Object 6060_h: Modes of Operation

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

VALUE DEFINITION

Value	Definition
0	No mode*
1	Profile position mode
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	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: Modes of Operation Display

This object shows the operating mode that is set.

VALUE DEFINITION

Value	Definition
0	No mode*
1	Profile position 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	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: Sensor Selection Code

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

VALUE DEFINITION

Value	Definition
0	Encoder used
-1	No encoder

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
606A _h	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 608F_h: Position Encoder Resolution

This object indicates the configured encoder increments and number of motor revolutions. The position encoder resolution is calculated by the following formula (all values are dimensionless.):

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
608F _h	Position 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

4.1.10 Object 60FD_h: Digital Inputs

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

VALUE DEFINITION

Value	Definition
0 _b	Switched off
1 _b	Switched on

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
60FD _h	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

4.1.11 Object 6502_h: Supported Drive Modes

This object provides information on the supported drive modes and contains always the *value* 0x00000025 which means that the following modes are provided by the drive:

- 0 = no mode

- 1 = profile position mode
- 3 = profile velocity mode
- 6 = homing mode

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6502 _h	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	0x00000025

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

4.2.1 Detailed Object Specifications

The following text offers detailed object specifications. According to this it is necessary to show how the state machine works in detail:

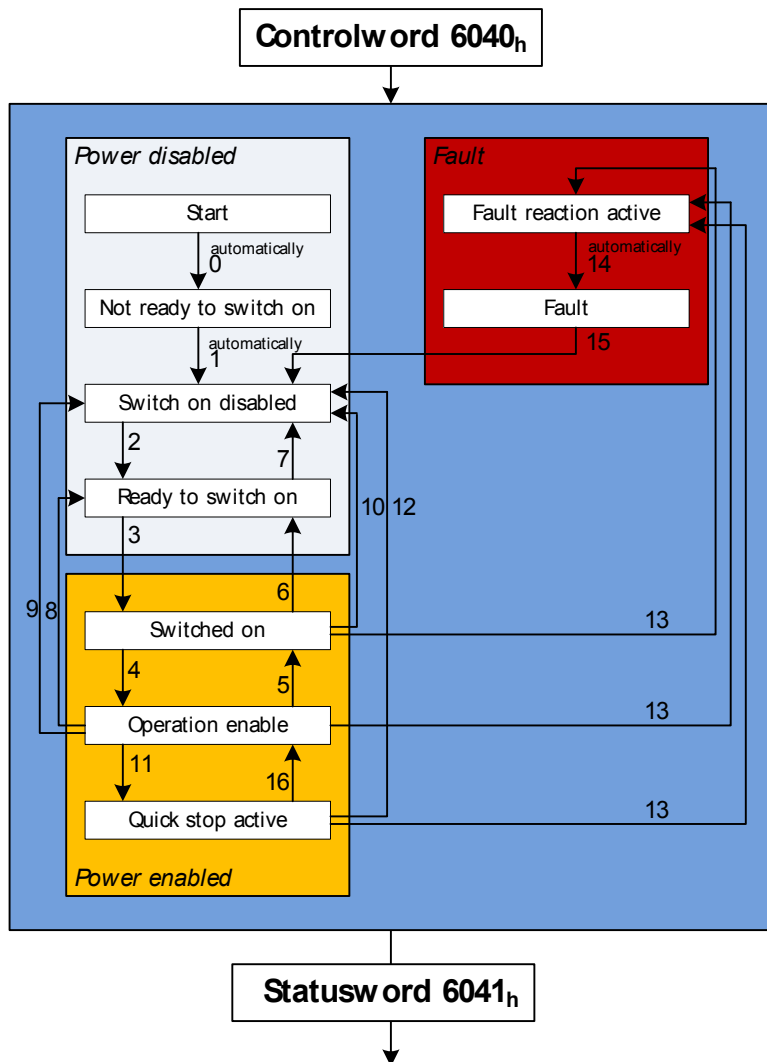


Figure 4.1 Finite state machine

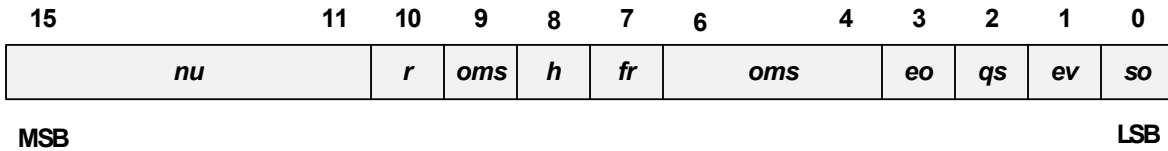
NOTES ON STATE TRANSITIONS

- 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: Controlword

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

STRUCTURE OF CONTROLWORD / VALUE DEFINITION



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

OPERATION MODE SPECIFIC BITS IN PP MODE

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

COMMAND CODING

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6040 _h	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: Statusword

This object 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	11	10	9	8	7	6	5	4	3	2	1	0
<i>ms</i>	<i>oms</i>	<i>ila</i>	<i>tr</i>	<i>rm</i>	<i>ms</i>	<i>w</i>	<i>sod</i>	<i>qs</i>	<i>ve</i>	<i>f</i>	<i>oe</i>	<i>so</i>	<i>rtso</i>

MSB

LSB

LEGEND: *ms* = manufacturer specific; *oms* = operation mode specific; *ila* = internal limit active; *tr* = target reached; *rm* = remote; *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 acknowledgement	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	Statusword	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

4.2.1.3 Object 6062_h: Position Demand Value

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6062 _h	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: Position Actual Internal Value

This object provides the actual value of the encoder or the motor. Please use the sensor selection object 606A_h (4.1.8) for selecting the motor or the encoder first. Object 6063_h indicates the actual position of the encoder or the motor, re-scaled to the microstep resolution. The value is given in microsteps.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6063 _h	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: Position Actual Value

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6064 _h	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: Following Error Window

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

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

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6065 _h	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: Position Window

This object indicates the configured symmetrical range of accepted positions relative to the target position. If the actual value of the position encoder is within the position window, this target position is regarded as having been reached. The value is given in increments. If the value of the position window is FFFF FFFF_h, the position window control is switched off.

If this object is set to zero, the target reached event will be signaled when the demand position (6062_h) has reached the target position (6064_h).

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6067 _h	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: Position Window Time

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6068 _h	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 60A_h: Profile Jerk

This object indicates the configured set of jerk parameters that shall be used during the profile movement. The bow of S-shaped ramps is set here. Object 60A_h can only be used when S-shaped ramps are selected by setting object 6086_h to 1.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
60A _h	Position jerk	ARRAY	UNSIGNED32

ENTRY DESCRIPTION

Sub-Index	Description	Entry category	Access	PDO Mapping	Value Range	Default Value
01 _h	Bow of S-shaped ramps	Mandatory	rw	Refer to CiA402-3	UNSIGNED32	1... 18

4.2.1.10 Object 606C_h: Velocity Actual Value

This object shows the actual velocity value of the motor. The value is given in internal or user-defined velocity units (depending on object 208C_h, described in paragraph 5.1.13).

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
606C _h	Velocity actual value	Variable	SIGNED32

ENTRY DESCRIPTION

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

4.2.1.11 Object 607A_h: Target Position

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
607A _h	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

4.2.1.12 Object 607D_h: Software Position Limit

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

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

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
607D _h	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

4.2.1.13 Object 6081_h: Profile Velocity

This object indicates the configured velocity normally attained at the end of the acceleration ramp during a profiled motion and is valid for both directions of motion. The profile velocity is the maximum velocity used when driving to a new position. It is given in internal or user specific units (depending on object 208C_h, paragraph 5.1.13)

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6081 _h	Profile velocity	Variable	UNSIGNED32

ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 _h	rw	Refer to CiA402-3	UNSIGNED32	<i>depends on the units</i>

4.2.1.14 Object 6083_h: Profile Acceleration

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

Please choose the units for object 6083_h with object 208E_h, described in paragraph 5.1.14. Object 208E_h reads 0 when internal units are selected or 179 when user units (PPS/s) are selected.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6083 _h	Profile acceleration	Variable	UNSIGNED32

ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 _h	rw	Refer to CiA402-3	UNSIGNED32	<i>depends on the units</i>

4.2.1.15 Object 6084_h: Profile Deceleration

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

Please choose the units for object 6084_h with object 208E_h, described in paragraph 5.1.14. Object 208E_h reads 0 when internal units are selected or 179 when user units (PPS/s) are selected. The deceleration is according to the acceleration.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6084 _h	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

4.2.1.16 Object 6085_h: Quick Stop Deceleration

This object indicates the configured deceleration used to stop the motor when the *quick stop function* is activated and the *quick stop code object* 605A_h is set to 2 [or 6]. The value is given in the same unit as *profile acceleration object* 6083_h. Object 6085_h defines the deceleration that is to be used during a quick stop operation.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6085 _h	Quick stop 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

4.2.1.17 Object 6086_h: Motion Profile Type

This object indicates the configured type of motion profile used to perform a profiled motion. Set this object to 0 for trapezoid ramps and to 1 for S-shaped ramps.

VALUE DEFINITION

Value	Definition
0	Linear ramp, trapezoidal profile
1	Sin ² ramp, S-shaped profile, refer to object 60A4 _h (profile jerk, paragraph 4.2.1.9)

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6086 _h	Motion profile type	Variable	UNSIGNED16

ENTRY DESCRIPTION

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

4.2.2 How to Move a Motor in *pp* Mode

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

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

4.3 Homing Mode

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

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

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

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

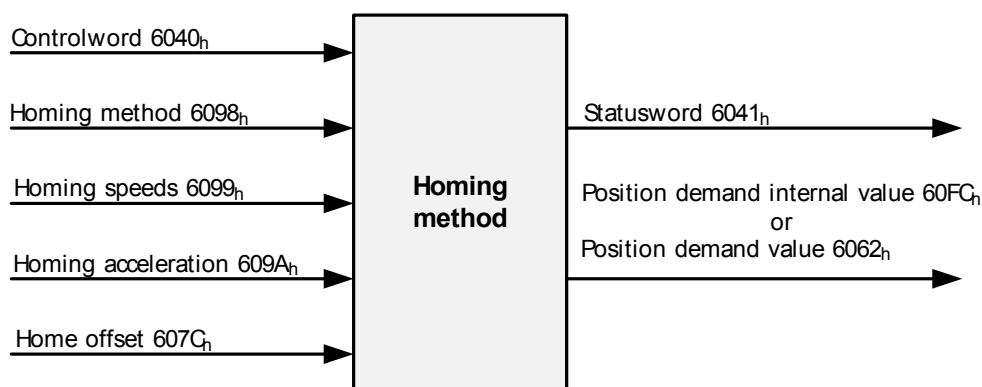


Figure 4.2 Homing mode function

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 where appropriate, and
- the position of the index pulse.

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

THERE ARE FOUR SOURCES OF HOMING SIGNAL AVAILABLE:

- negative and positive limit switches
- home switch
- index pulse of an encoder

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

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

4.3.1 Connecting Home Switch, Left Switch, and Right Switch

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

4.3.1.1 TCM-142-IF

The TCM-142-IF module consists of two PCBs, the TCM-142 base board and the standard TCM-IF adapter/interface add-on board.

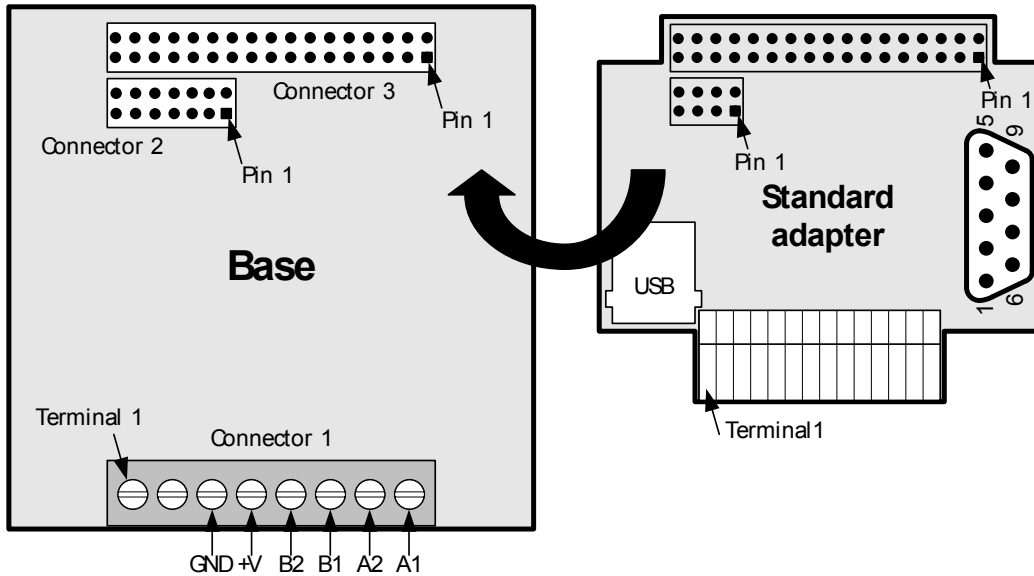


Figure 4.3 Connectors of TCM-142-IF

Please connect home switch, left switch, and right switch as follows. Use base connector 3 (or rather the corresponding connector of the adapter board).

CONNECTING SWITCHES

Pin (connector 3)	Name	Function
23	HOME (IN_8)	Digital user controlled input #9. Optically isolated, active low (power supply on pin 15)
19	STOP_L	Optically isolated, active low limit switch input <i>left</i>
7	STOP_R	Optically isolated, active low limit switch input <i>right</i>

4.3.1.2 PDx-146-60-SE

The PDx-146-60-SE provides two D-SUB connectors for external connections, a 9-pin D-SUB male connector for CAN communication and a 26-pin high density D-SUB male connector for power, additional I/Os, and switches.

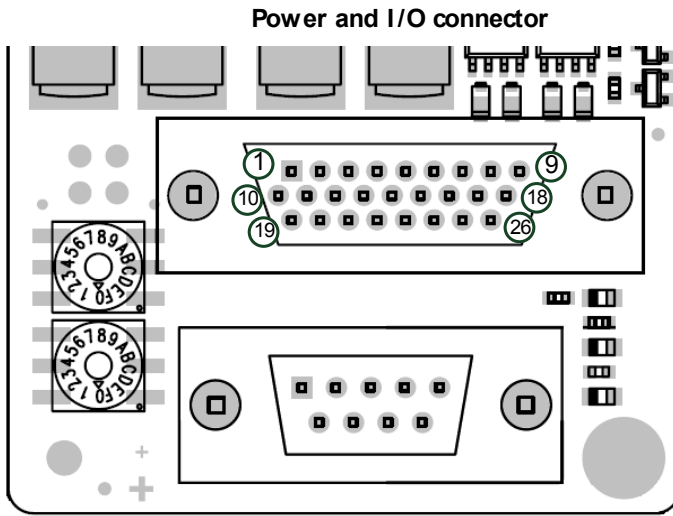


Figure 4.4 Connectors of PDx-146-60-SE

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

CONNECTING SWITCHES

Pin	Name	Function
16	STOP_L	Left stop switch input (+24V and +5V tolerant, internal 20k pull-down resistor, programmable 1k pull-up to +5V with protection diode)
17	STOP_R	Right stop switch input (+24V and +5V tolerant, internal 20k pull-down resistor, programmable 1k pull-up to +5V with protection diode)
18	HOME	Home switch input (+24V and +5V tolerant, internal 20k pull-down resistor, programmable 1k pull-up to +5V with protection diode)

4.3.2 Homing Methods

There are several different methods of homing. For choosing your homing method, refer to object 6098_h please.

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; then search the next encoder index pulse.
5	Search the negative edge of the home switch; then search the next encoder index pulse.
17	Search the left end switch.
18	Search the right end switch.
19	Search the positive edge of the home switch.
21	Search the negative edge of the home switch.
33	Search next index pulse in negative direction.
34	Search next index pulse in positive direction.
35	The actual position is used as home position. All position values (objects 6062 _h , 6063 _h , and 6064 _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.

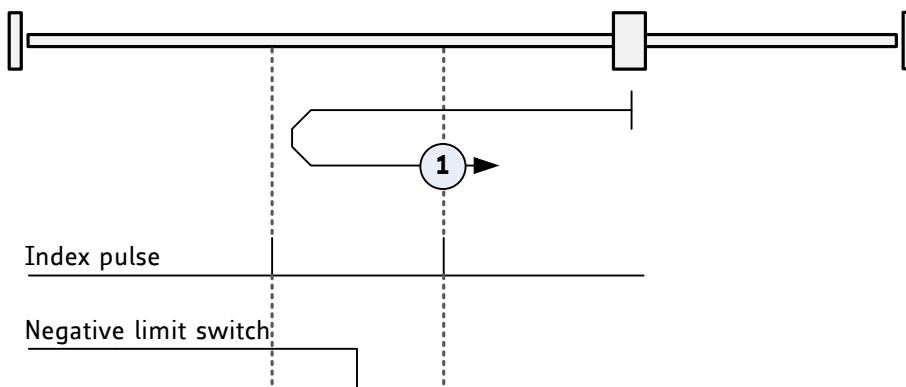


Figure 4.5 Homing on negative switch and index pulse

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.

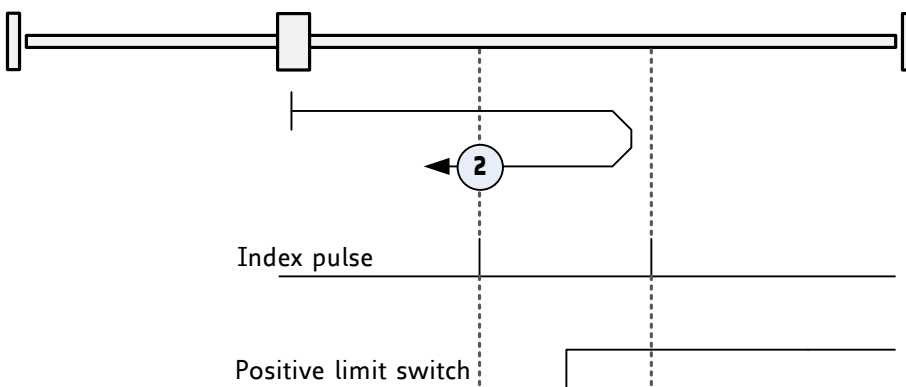


Figure 4.6 Homing on positive limit switch and index pulse

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.

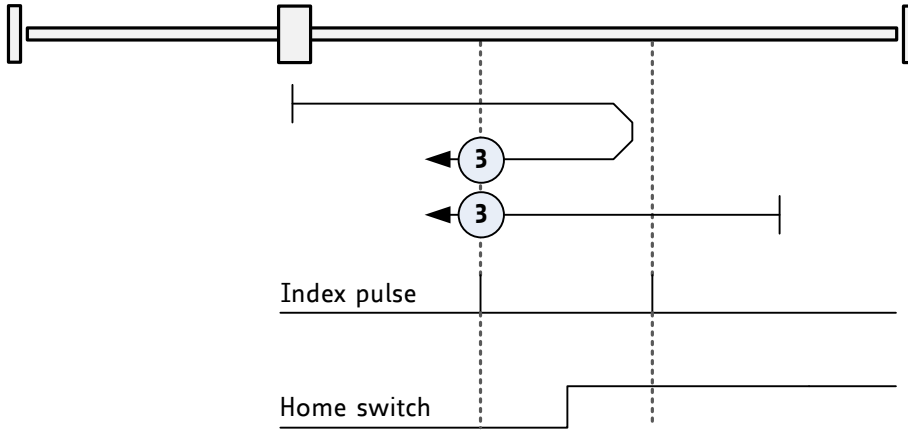


Figure 4.7 Homing on positive home switch and index pulse

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.

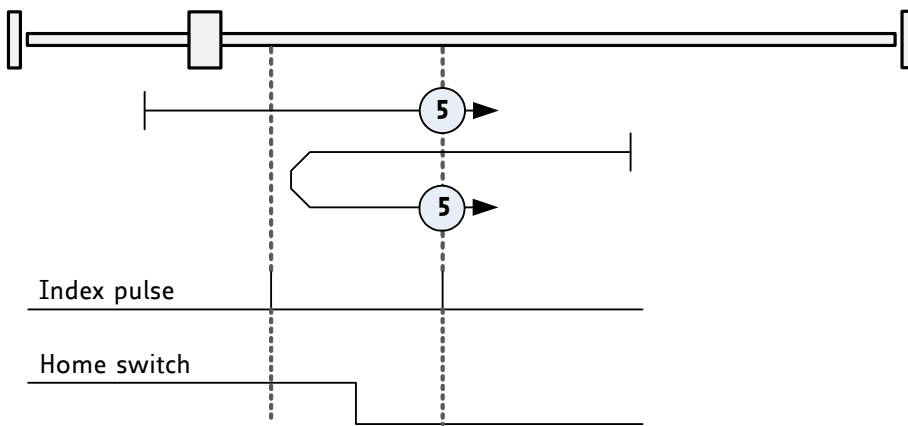


Figure 4.8 Homing on negative home switch and index pulse

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 4.9.

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.)

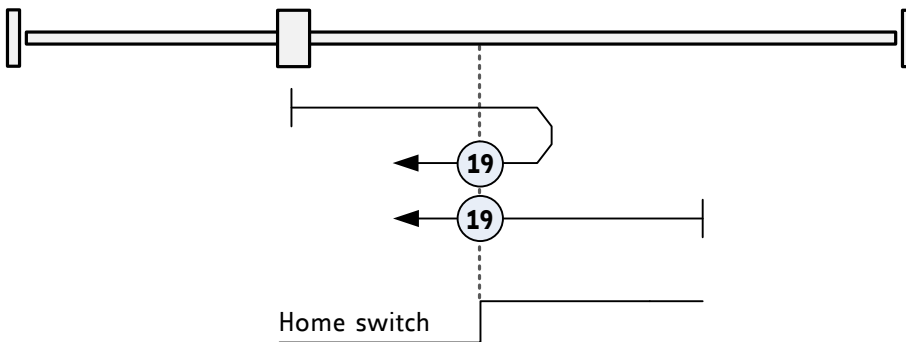


Figure 4.9 Homing without index pulse

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 4.10.

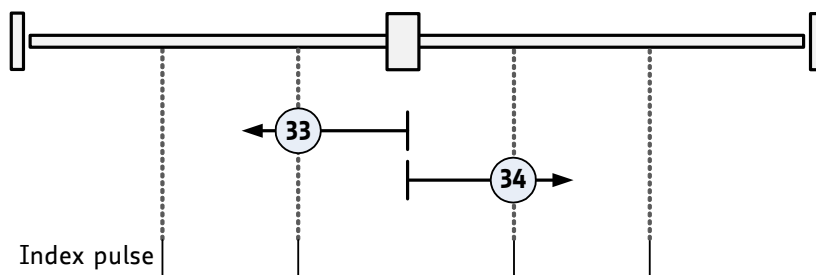


Figure 4.10 Homing on index pulse

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

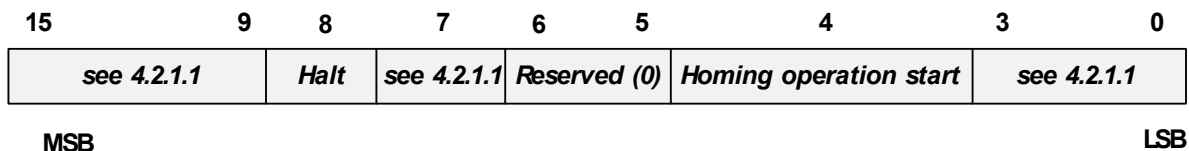
In this method, the current position shall be taken to be the home position. This method does not require the drive device to be in operation enabled state.

4.3.3 Detailed Object Definitions

4.3.3.1 Object 6040_h: Controlword

This object 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



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	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.	<i>device and operation mode specific</i>

4.3.4 Object 6041_h: Statusword

This object 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	0
<i>see 4.2.1.2</i>	<i>Homing error</i>	<i>Homing attained</i>	<i>see 4.2.1.2</i>	<i>Target reached</i>	<i>see 4.2.1.2</i>		

MSB

LSB

OPERATION MODE SPECIFIC BITS IN 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	Statusword	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

4.3.4.1 Object 606C_h: Velocity Actual Value

This object shows the actual velocity value of the motor. The unit can be selected with object 208C_h, described in paragraph 5.1.13.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
606C _h	Velocity actual value	Variable	SIGNED32

ENTRY DESCRIPTION

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

4.3.4.2 Object 607C_h: Home Offset

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

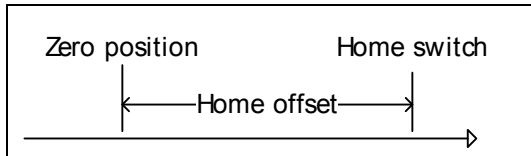


Figure 4.11 Home offset definition

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
607C _h	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

4.3.4.3 Object 6098_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 the home position. All position values (objects 6062 _h , 6063 _h , and 6064 _h) are set to zero, but the motor will not move.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6098 _h	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

4.3.4.4 Object 6099_h: Homing Speeds

This object indicates the configured speeds used during homing procedure. The values are given in units, which can be selected with object 208C_h, described in paragraph 5.1.13. Using object 6099_h a fast and a slow homing speed can be set. In most homing modes, the home switch is searched with the fast speed first. When the home switch has been found, the motor will be decelerated to the slow speed (using the homing acceleration, object 609A_h) and then stopped at the exact switch point.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6099 _h	Homing speeds	ARRAY	UNSIGNED32

ENTRY DESCRIPTION

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

4.3.4.5 Object 609A_h: Homing Acceleration

This object indicates the configured acceleration and deceleration to be used during homing operation. The values for accelerating to the fast homing speed and for decelerating to the slow homing speed can be set here. The values are given in units, which can be selected with object 208E_h, described in paragraph 5.1.14.

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

4.3.4.6 Object 2100_h: Home Offset Display

The value is given in microsteps.

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

4.3.5 How to Start a Homing in *hm* Mode

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

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

4.4 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:

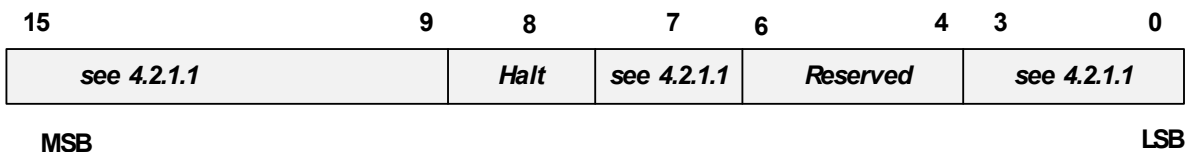
- Profile velocity
- Profile acceleration
- Profile deceleration
- Emergency stop
- Motion profile type

4.4.1 Detailed Object Definitions

4.4.1.1 Object 6040_h: Controlword

This object 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



There are no mode specific bits supported in this mode.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6040 _h	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

4.4.1.2 Object 6041_h: Statusword

This object 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	0
see 4.2.1.2	Max slippage error	Speed	see 4.2.1.2	Target reached	see 4.2.1.2		
MSB						LSB	

OPERATION MODE SPECIFIC BITS IN PV MODE

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6041 _h	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 6062_h: Position Demand Value

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6062 _h	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

4.4.1.4 Object 6063_h: Position Actual Internal Value

This object indicates the actual position of the encoder, re-scaled to the microstep resolution of the motor. 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 object 606A_h (described in paragraph 4.1.8).

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6063 _h	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

4.4.1.5 Object 6064_h: Position Actual Value

This object provides the actual value of the position measurement device. It always contains the same value as object 6063_h. The value is given in microsteps.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6064 _h	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

4.4.1.6 Object 6065_h: Following Error Window

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

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6065 _h	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

4.4.1.7 Object 606C_h: Velocity Actual Value

This object gives the actual velocity value derived either from the velocity sensor or from the position sensor.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
606C _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

4.4.1.8 Object 607D_h: Software Position Limit

This object indicates the configured maximal and minimal software position limits. These parameters define the absolute position limits for the position demand value and the position actual value. The position actual value is always checked against these limits. The limit position is always relative to the machine home position. Before being compared with the 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 moving the motor in positioning mode can be set here. Overstepping the window will lead to an emergency message and the motor will be stopped.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
607D _h	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

4.4.1.9 Object 6083_h: Profile Acceleration

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

Please choose the units for object 6083_h with object 208E_h, described in paragraph 5.1.14. Object 208E_h reads 0 when internal units are selected or 179 when user units (PPS/s) are selected.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6083 _h	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

4.4.1.10 Object 6084_h: Profile Deceleration

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

Please choose the units for object 6083_h with object 208E_h, described in paragraph 5.1.14. Object 208E_h reads 0 when internal units are selected or 179 when user units (PPS/s) are selected.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6084 _h	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

4.4.1.11 Object 6085_h: Quick Stop Deceleration

This object indicates the configured deceleration used to stop the motor when the *quick stop function* is activated and the *quick stop code object* 605A_h is set to 2 or 6. The *quick stop deceleration* is also used if the *fault reaction code object* 605E_h is 2 and the *halt option code object* 605D_h is 2. Object 6085_h defines the deceleration that is to be used during a quick stop operation.

The value is given in the same physical unit as *profile acceleration object* 6083_h. You can choose the units for object 6083_h with object 208E_h, described in paragraph 5.1.14. Object 208E_h reads 0 when internal units are selected or 179 when user units (PPS/s) are selected.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
6085 _h	Quick stop 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

4.4.1.12 Object 60FF_h: Target Velocity

This object indicates the configured target velocity and is used as input for the trajectory generator. Object 60FF_h sets the target velocity when using profile velocity mode. The drive then accelerates or decelerates to that velocity using the acceleration and deceleration set by objects 6083_h and 6084_h. The values are given in units, which can be selected with object 208C_h, described in paragraph 5.1.13.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
60FF _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

4.4.2 How to Move a Motor in pv Mode

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

- If you do not have any limit switches connected, first disable the limit switch inputs by writing 3 to object 2005_h.
- Select *pv* mode by writing 3 to object 6060_h.
- Write 6 to object 6040_h to switch to READY_TO_SWITCH_ON state.
- Write 7 to object 6040_h to switch to SWITCHED_ON state.
- Write 15 to object 6040_h to switch to OPERATION_ENABLED state.
- Write the desired target speed (e.g. 100000) to object 60FF_h. The motor now accelerates to that speed.
- Stop the motor by writing 0 to object 60FF_h.

5 Manufacturer Specific Area

The manufacturer segment contains manufacturer specific objects. These objects control the special features of the TRINAMIC motion control devices TMC-142 and PD-146.

5.1 Detailed Object Specifications

5.1.1 Object 2000_h: Microstep Resolution

This object sets the microstep resolution of the drive. A value of 0 means 2048 microsteps ($2048/2^x$). It is only writeable in the SWITCHED_ON_DISABLED state, but always readable.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2000 _h	Microstep resolution	Variable	UNSIGNED8

ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 _h	rw	no	0... 11	0

5.1.2 Object 2001_h: Fullstep Resolution

This object shows the fullstep resolution of the motor shipped with a PANdrive. It is normally 200 (1.8° motor).

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2001 _h	Fullstep resolution	Variable	UNSIGNED16

ENTRY DESCRIPTION

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

5.1.3 Object 2002_h: Brake Delay Times

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

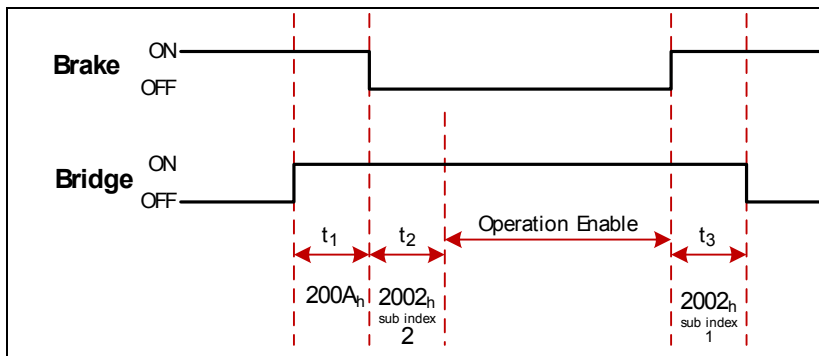


Figure 5.1 Brake output timing

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2002 _h	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

5.1.4 Object 2003_h: Maximum Current

This object defines the current used when the motor is moving. A value of 16 means 100% of the maximum current of the drive.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2003 _h	Maximum current	Variable	UNSIGNED8

ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 _h	rw	no	0... 16	7

5.1.5 Object 2004_h: Standby Current

This object defines the current used when the motor is standing. A value of 16 means 100% of the maximum current of the drive.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2004 _h	Standby current	Variable	UNSIGNED8

ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 _h	rw	no	0... 16	0

5.1.6 Object 2005_h: Limit Switches

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

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

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2005 _h	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.

5.1.7 Object 200A_h: Enable Drive Delay Time

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
200a _h	Enable drive delay time	Variable	UNSIGNED16

ENTRY DESCRIPTION

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

5.1.8 Object 200B_h: Encoder Parameters

This object defines the polarity of the encoder channels. It is only writable in SWITCHED_ON_DISABLED state.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
200b _h	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
	Bit 0 Polarity of channel A on index pulse (0=low, 1=high)				
	Bit 1 Polarity of channel B on index pulse (0=low, 1=high)				
	Bit 2 Polarity of channel N on index pulse (0=low, 1=high)				
Bit 3 Ignore channel A/B polarity on index pulse					
02 _h	Direction of rotation	rw	no	0/1	0
03 _h	Initialize position ^{*)}	rw	no	0/1	1

5.1.9 Object 200C_h: Brake Current Feed

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
200c _h	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

5.1.10 Object 2089_h: Standby Delay

Time after the motor is stopped until the motor current is changed to standby current (in units of 4.096ms). This can only be changed in SWITCHED_ON_DISABLED mode.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2089 _h	Standby delay	Variable	UNSIGNED16

ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 _h	rw	no	0... 4096	0

5.1.11 Object 208A_h: Mixed Decay Run

Use mixed decay when the motor is running if set to 1. This can only be changed in SWITCHED_ON_DISABLED mode.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
208a _h	Mixed decay run	Variable	UNSIGNED8

ENTRY DESCRIPTION

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

5.1.12 Object 208B_h: Mixed Decay Standby

Use mixed decay when the motor is standing if set to 1. This can only be changed in SWITCHED_ON_DISABLED mode.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
208b _h	Mixed decay standby	Variable	UNSIGNED8

ENTRY DESCRIPTION

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

5.1.13 Object 208C_h: Velocity Dimension Index

Writing 0 selects internal units, writing 181 sets PPS units for velocity and PPS/s units for acceleration. This can only be changed in SWITCHED_ON_DISABLED mode.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
208c _h	Velocity dimension index	Variable	UNSIGNED8

ENTRY DESCRIPTION

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

5.1.14 Object 208E_h: Acceleration Dimension Index

This object reads 0 when internal units are selected or 179 when user units (PPS/s) are selected. This depends on the setting of object 208E_h, paragraph 5.1.14)

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
208e _h	Acceleration dimension index	Variable	UNSIGNED8

ENTRY DESCRIPTION

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

5.1.15 Object 2100_h: Home Offset Display

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

5.1.16 Object 2102_h: Driver Error Flags

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

THERE ARE THE FOLLOWING ERROR BITS:

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2102 _h	Driver error flags	Variable	UNSIGNED8

ENTRY DESCRIPTION

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

5.1.17 Object 2107_h: Microstep Resolution Display

This object shows the microstep resolution (set by object 2000_h when internal units are selected or calculated when user units are selected).

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 _h	ro	no	0.. 11	0

5.1.18 Object 2108_h: Absolute Resolver Value

This object shows the actual value of the encoder.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2108 _h	Absolute resolver value	Variable	UNSIGNED32

ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00 _h	ro	no	0.. 4095	no

5.1.19 Object 2700_h: TMCL Direct Communication

After writing the *make signature* 0x656b616d to this object the drive switches to TMCL™ mode. The drive can then only be controlled via TMCL™ commands written to the *OS command* object 1023/01_h

The drive has to be reset (using the *DS-301 reset application command*) to return to normal CANopen functionality.

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 0x656b616d	0

5.1.20 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

5.1.21 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

5.1.22 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. The number of available digital outputs depends on the module type.

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2703 _h	Digital outputs	Array	

ENTRY DESCRIPTION

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

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

5.1.23 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. 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	1000

5.1.24 Object 2705_h: Node ID

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

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

HOW TO CHANGE THE NODE ID:

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

5.1.25 Object 2706_h: Store

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

Save signature: 65766173_h

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
2706 _h	Store	Variable	UNSIGNED32

ENTRY DESCRIPTION

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

5.1.26 Object 2707_h: CAN Bit Rate Load

Shows the selected CAN bit rate.

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

5.1.27 Object 2708_h: Node ID load

This object shows the selected node ID.

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

5.1.28 Object 270E_h: Device Analog Inputs

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

5.1.1 Object 2710_h: Pull-up Resistors

This object switches the pull-up resistors of the inputs on or off. Each bit switches one of the inputs.

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

6 Emergency messages

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

ERROR CODES FOR THE PD-146 AND THE TMC-142:

Error code	Add. byte					Description
	1	2	3	4	5	
0000 _h	0	0	0	0	0	Fault reset The fault reset command has been executed.
1000 _h	1	0	0	0	0	Generic error: open load bridge A The motor driver indicates open load on bridge A. It is possible that the motor cable is broken or that there is an error in the power amplifier itself.
1000 _h	2	0	0	0	0	Generic error: open load bridge B The motor driver indicates open load on bridge B. It is possible that the motor cable is broken or that there is an error in the power amplifier itself.
2310 _h	0	0	0	0	0	Overcurrent high side The motor driver indicates an overcurrent on the high side. This can be caused by a short circuit in the driver stage.
2311 _h	0	0	0	0	0	Overcurrent bridge B The motor driver indicates that there is overcurrent on bridge B. This can be caused by a short circuit in the motor itself or in the motor driver stage.
2312 _h	0	0	0	0	0	Overcurrent bridge A The motor driver indicates that there is overcurrent on bridge A. This can be caused by a short circuit in the motor itself or in the motor driver stage.
3230 _h	0	0	0	0	0	stallGuard™ error The actual load value exceeds the stallGuard™ limit. Please note, that the actual versions of PD-146 and TMC-142 do not have the stallGuard™ feature. We consider this for future versions.
4310 _h	1	0	0	0	0	Overtemperature pre-warning The temperature in the motor driver exceeds the pre-warning limit.
4310 _h	2	0	0	0	0	Overtemperature error The motor driver has been switched off because the temperature limit has been exceeded.
5441 _h	0	0	0	0	0	Shutdown switch active The enable signal is missing (due to the shutdown switch) and the motor driver has been switched off.
6320 _h	0	0	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.
8110 _h	1	0	0	0	0	CAN controller overflow The receive message buffer of the CAN controller hardware is full and some CAN messages are lost.
81110 _h	2	0	0	0	0	CAN Tx buffer overflow The software CAN transmit buffer is full and thus some CAN messages are lost.
81110 _h	3	0	0	0	0	CAN Rx buffer overflow The software CAN receive buffer is full and so some CAN messages are lost.
8120 _h	0	0	0	0	0	CAN error passive The CAN controller has detected communication errors and has entered the CAN Error passive state.
8140 _h	0	0	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.
8611 _h	0	0	0	0	0	Following error The deviation between motor position counter and encoder position counter has exceeded the following error window.
0xff00	0	0	0	0	0	Undervoltage The supply voltage is too low to drive a motor.
ff01 _h	1	0	0	0	0	Positive software limit The actual position is outside the range defined by object 0x607d.

ff01 _h	2	0	0	0	0	Negative software limit The actual position is outside the range defined by object 0x607d.
ff01 _h	3	0	0	0	0	Positive limit switch The positive limit switch has been touched outside of the homing function.
ff01 _h	4	0	0	0	0	Negative limit switch The negative limit switch has been touched outside of the homing function.

7 Life Support Policy

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

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

© TRINAMIC Motion Control GmbH & Co. KG 2011-2017

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

Specifications are subject to change without notice.



8 Revision History

8.1 Firmware Revision

Version	Date	Author	Description
3.00	2009-SEP-15	OK	First version for TMC457 based modules
3.01	2009-OCT-12	OK	Home switch polarity reversed Node ID and bitrate can be configured via manufacturer specific objects
3.02	2010-JAN-07	OK	Home switch polarity switchable (via object 2005 _h)
3.03	2010-JAN-29	OK	Internal error corrections and optimizations
3.05	2010-DEC-15	OK	Heartbeat consumer functionality added Non volatile storage of some motor parameters added Analog inputs can be read via object 270E _h
3.06... 3.17		OK	Not deployed.
3.18	25.04.2014	OK	<i>Attention!</i> Per default, the encoder is not active. Set the sensor selection code (object 606A _h) to meet your needs. New or modified objects: <ul style="list-style-type: none"> - 100C_h guard time new - 100D_h life time factor new - 1010_h store parameters updated - 1011 restore default parameters updated - 2108_h absolute resolver value new - 6098_h homing methods updated and new methods added - 2108_h absolute resolver value new
3.20	9.6.2017	OK	Fixed save/restore function. New object 0x2710 (pull-up resistors).

Table 8.1 Firmware revision

8.2 Document Revision

Version	Date	Author	Description
0.95	2009-SEP-15	OK	Initial version for TMC457 based modules
0.96	2009-OCT-12	OK	Description of homing modes corrected
1.00	2010-FEB-23	SD	Initial version based on version 0.96
1.01	2010-MAR-15	SD	Controlword and statusword corrected
1.02	2010-MAY-24	SD	Objects 2704 _h , 2705 _h , 2707 _h , 2708 _h corrected
1.03	2010-JUN-15	SD	Object 2706 _h and 6098 _h corrected, minor changes
1.04	2011-AUG-01	SD	Objects 1010 _h , 1011 _h , and 1016 _h added; new front page
2.00	2014-JUN-26	SD	<p><i>Attention!</i></p> <p>Per default, the encoder is not active. Set the sensor selection code (object 606A_h) to meet your needs.</p> <p>New or modified objects:</p> <ul style="list-style-type: none"> - 100C_h guard time new - 100D_h life time factor new - 1010_h store parameters updated - 1011 restore default parameters updated - 2108_h absolute resolver value new - 6098_h homing methods updated and new methods added - 2108_h absolute resolver value new - 200B_h updated <p>Changes related to the design.</p>
2.01	2017-JUL-26	OK	Firmware revision 3.20 included.

Table 8.2 Document revision

9 References

[TMC-142-IF]	TMC-142-IF Hardware Manual (see http://www.trinamic.com)
[TMC-142-IF]	TMC-142-IF TMCL™ Firmware Manual (see http://www.trinamic.com)
[PD-146-60-SE]	PD-146-60-SE Hardware Manual (see http://www.trinamic.com)
[PD-146-60-SE]	PD-146-60-SE TMCL™ Firmware Manual (see http://www.trinamic.com)
[TMC-IF]	TMC-IF Hardware Manual (see http://www.trinamic.com)
[QSH5718]	QSH5718 Manual (see http://www.trinamic.com)
[QSH6018]	QSH6018 Manual (see http://www.trinamic.com)