Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

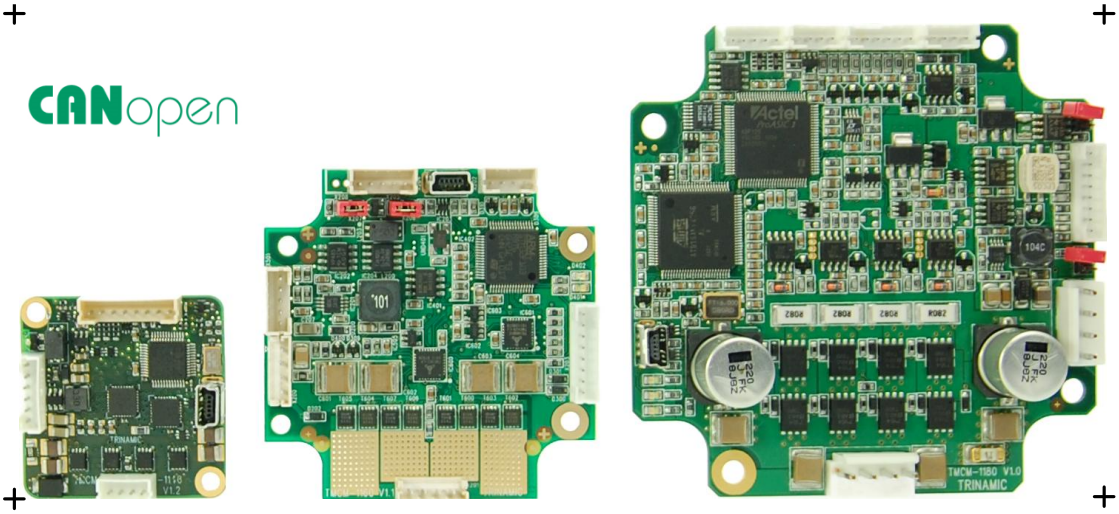
Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

CANopen Firmware Version V3.16

CANopen MANUAL

TMCM-1140 TMCM-1160 TMCM-1180

1-Axis Controller/Driver Modules
CANopen
Motor Mountable (NEMA17, NEMA23/24, NEMA34)
Integrated sensOstep™ Encoder



UNIQUE FEATURES

coolStep™

stallGuard2™

TRINAMIC Motion Control GmbH & Co. KG
Hamburg, Germany

www.trinamic.com

TRINAMIC
MOTION CONTROL

Table of Contents

Table of Figures

1 Preface

This document specifies objects and modes of operation for the TMCM-1140, TMCM-1160, and TMCM-1180 with CANopen firmware. As these products are stepper motor controller and driver modules the use of the CiA DSP402 protocol (described in the *CiA CANopen drives and motion control device profile, Part 2*) is fundamental. The CANopen firmware is designed to fulfill the DS301 version 4.02 and DS402 version 3.0 standards. The CiA conformance has also been tested. This manual assumes that the reader is already familiar with the basics of the CANopen protocol (especially DS301 and DS402). On the following pages you will find a overview and afterwards the information will be more in detail.

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

1.1 General Features of the CANopen Implementation

MAIN CHARACTERISTICS

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

SDO communication

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

PDO communication

- Producer
- Consumer
- *RPDOs*
  - 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

1.2 Abbreviations

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

Table 1.1 Abbreviations

1.3 Firmware Update

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

TMCM-1140 AND TMCM1160

Connect the USB interface and use the bus-powered mode for a firmware update. Do not connect the power supply of the module.

TMCM-1180

Update the firmware using the TMCL-IDE. It is not necessary to use a specific interface.

TRINAMICs Unique Features – Easy to Use with CANopen

stallGuard2™ stallGuard2 is a high-precision sensorless load measurement using the back EMF on the coils. It can be used for stall detection as well as other uses at loads below those which stall the motor. The stallGuard2 measurement value changes linearly over a wide range of load, velocity, and current settings. At maximum motor load, the value goes to zero or near to zero. This is the most energy-efficient point of operation for the motor.

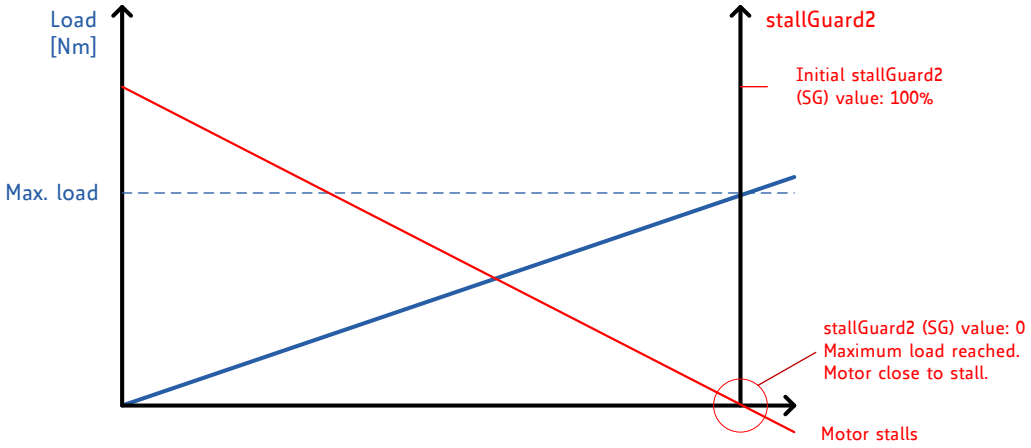


Figure 1.1 stallGuard2 load measurement SG as a function of load

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

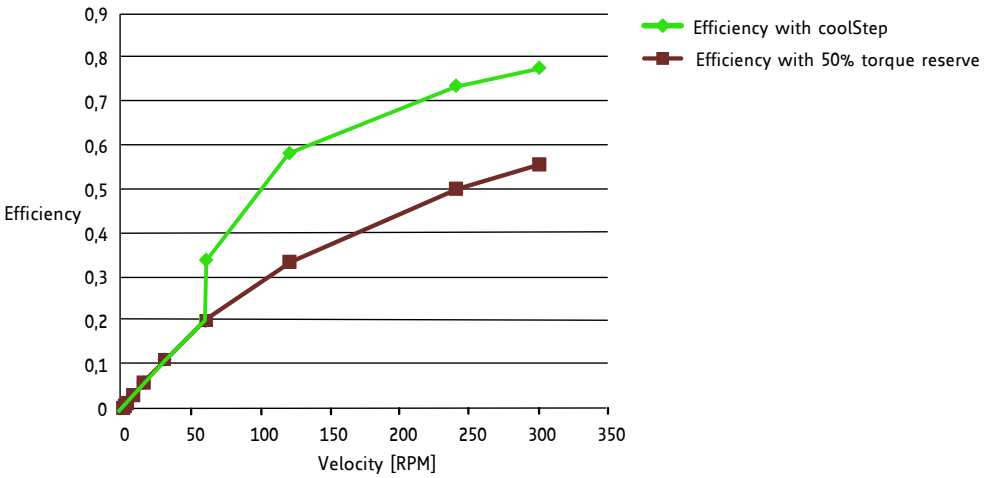


Figure 1.2 Energy efficiency example with coolStep

2 Communication

2.1 Reference Model

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

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

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

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

Table 2.1 Service primitives

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

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

Table 2.2 Service types

2.2 NMT State Machine

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

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

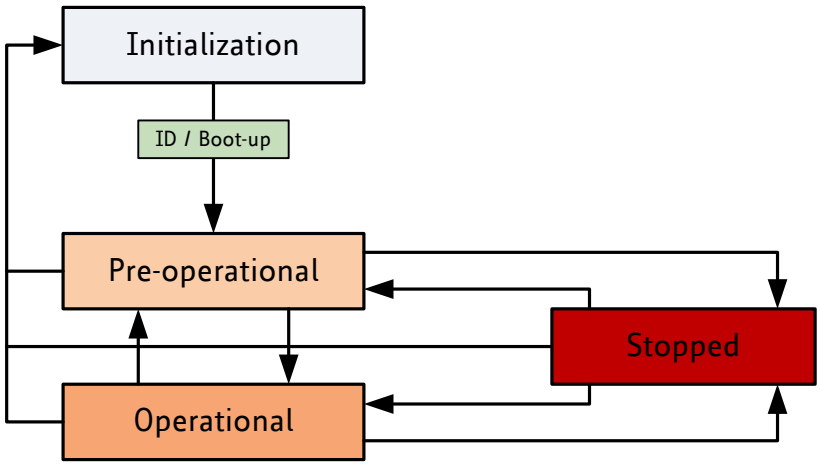


Figure 2.1 Overview CANopen NMT state machine

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

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

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

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

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

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

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

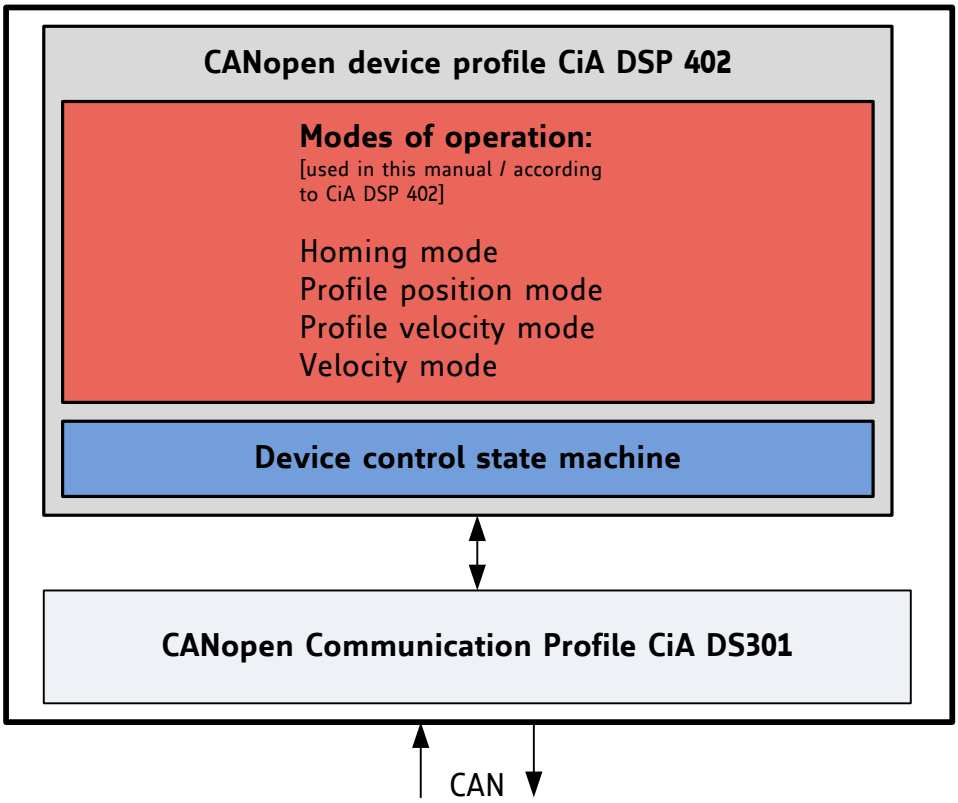


Figure 2.2 Communication architecture

2.3 Device Model

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

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

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

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

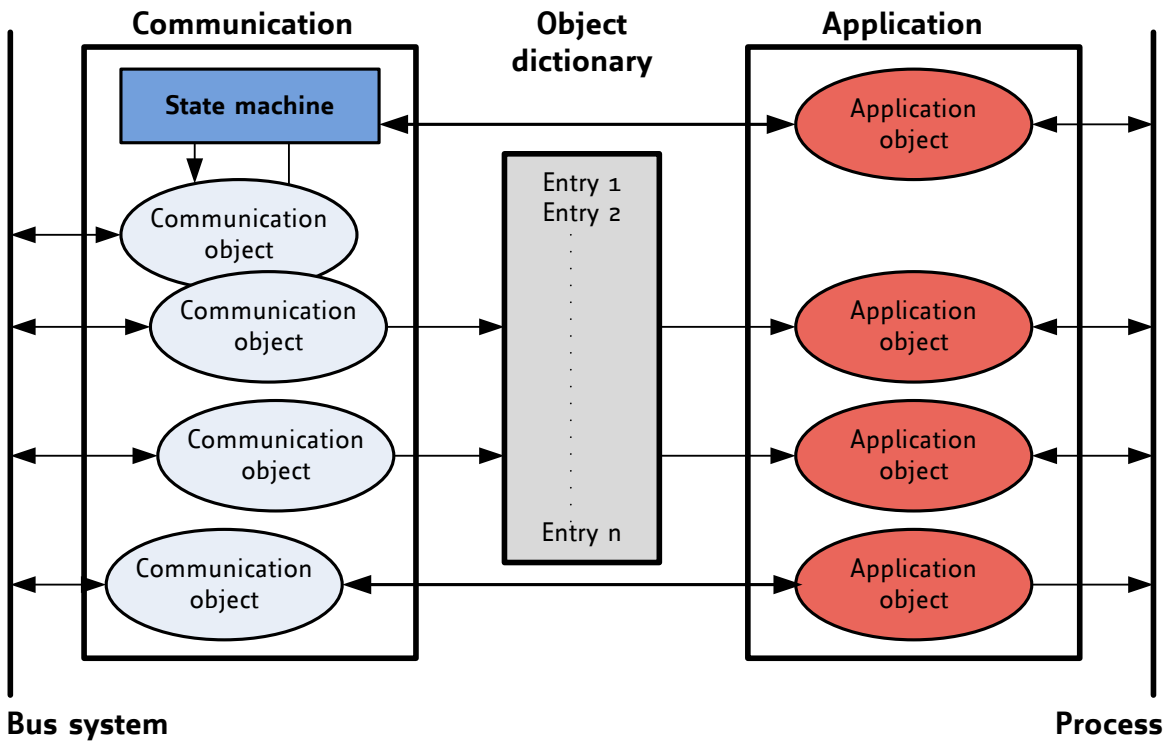


Figure 2.3 Device model

2.4 Object Dictionary

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

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

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

Table 2.3 Object Dictionary

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

*[2] The manufacturer 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-1140, TMCM-1160, and TMCM-1180.

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

3 Communication Objects

3.1 Detailed Object Specifications

3.1.1 Object 1000_h: Device Type

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

3.1.2 Object 1001_h: Error Register

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

STRUCTURE OF THE ERROR REGISTER

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

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

3.1.3 Object 1005_h: COB-ID SYNC Message

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

3.1.4 Object 1008_h: Manufacturer Device Name

This object contains the manufacturer device name.

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

Sub-Index	Access	PDO Mapping	Value Range	Default Value
00_h	const.	no	no	TMCM-1140 TMCM-1160 TMCM-1180 *depends on module*

3.1.5 Object 1009_h: Manufacturer Hardware Version

This object contains the hardware version description.

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

3.1.6 Object $100A_h$: Manufacturer Software Version

This object contains the software version description.

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

3.1.7 Object $100C_h$: Guard Time

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

3.1.8 Object $100D_h$: Life Time Factor

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

3.1.9 Object 1010_h: Store Parameters

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

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

SEVERAL PARAMETER GROUPS ARE DISTINGUISHED:

- Sub-index 0_h: contains the largest sub-index that is supported.
- Sub-index 1_h: saves all parameters.
- Sub-index 2_h: saves communication parameters $100C_h$, $100D_h$, 1015_h, 1017_h, and 1029_h.
- Sub-index 4_h: saves motor axis 1 parameters (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 1	Device does not save parameters autonomously Device saves parameters autonomously
0	0 1	Device does not save parameters on command Device saves parameters on command

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

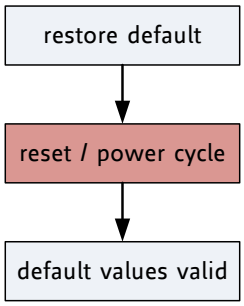


Figure 3.1 Restore procedure

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

Structure of restore read access

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

3.1.11 Object 1014_h: COB-ID Emergency Object

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

3.1.12 Object 1015_h: Inhibit Time EMCY

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

3.1.13 Object 1016_h: Consumer Heartbeat Time

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

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

Table 3.1 Structure of consumer heartbeat time entry

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

3.1.14 Object 1017_h: Producer Heartbeat Time

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

3.1.15 Object 1018_h: Identity Object

The object 1018_h contains general information about the device.

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

3.1.16 Object 1023_h: OS Command

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

3.1.17 Object 1029_h: Error Behavior

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

- Communication error
- Application error

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

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

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-1405_h	Receive PDO parameter	RECORD	PDO CommPar
1400_h	RPDO 1	RECORD	PDO CommPar
1401_h	RPDO 2	RECORD	PDO CommPar
1402_h	RPDO 3	RECORD	PDO CommPar
1403_h	RPDO 4	RECORD	PDO CommPar
1405_h	RPDO 6	RECORD	PDO CommPar

ENTRY DESCRIPTION

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

3.1.19 Objects 1600_h-1605_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. The values are hexadecimally coded.

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

Sub-Index	Description	Access	PDO Mapping	Value Range	Default Value
02_h	Mapping entry 2	rw	no	UNSIGNED32	Index 1600_h: 0 Index 1601_h: 60600008 $_h$ Index 1602_h: $607A0020_h$ Index 1603_h: $60FF0020_h$ Index 1605_h: 60420010_h
03_h	Mapping entry 3	rw	no	UNSIGNED32	Index 1600_h: 0 Index 1601_h: 0 Index 1602_h: 0 Index 1603_h: 0 Index 1605_h: 0

3.1.20 Objects 1800_h-1805_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-1805_h	Transmit PDO communication parameter	RECORD	PDO CommPar
1800_h	TPDO 1 transmit communication parameter	RECORD	PDO CommPar
1801_h	TPDO 2 transmit0 communication parameter	RECORD	PDO CommPar
1802_h	TPDO 3 transmit communication parameter	RECORD	PDO CommPar
1803_h	TPDO 4 transmit communication parameter	RECORD	PDO CommPar
1805_h	TPDO 6 transmit communication parameter	RECORD	PDO CommPar

ENTRY DESCRIPTION

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

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

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

OBJECT DESCRIPTION

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

ENTRY DESCRIPTION

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

4 Device Profile Objects (CiA402) and Modes of Operation

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

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

THE FOLLOWING OPERATION MODES CAN BE CHOSEN:

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

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

4.1 Detailed Object Specifications

4.1.1 Object $605A_h$: 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 *slow down ramp* and transit into *switch on disabled*
2	Slow down on *quick stop ramp* and transit into *switch on disabled*
5	Slow down on *slow down ramp* and stay in *quick stop active*
6	Slow down on *quick stop ramp* and stay in *quick stop active*

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
$605A_h$	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 shutdown option code always has the value 0 as only this is supported.

VALUE DEFINITION

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
605B$_h$	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. The slow down ramp is the deceleration value of the used mode of operation.

VALUE DEFINITION

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

OBJECT DESCRIPTION

Index	Name	Object Code	Data Type
605C$_h$	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 *slow down ramp* and stay in *operation enabled*

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