Field bus manual

Integrated positioning drives

IclA D065, CANopen

Operating system: 774.00, V1.0xx
Program No.: 775.00, V1.0xx

Order No.: 5960 0100 011

Edition: -000, 04.01

IclA®

SIG Positec Automation GmbH

Breslauer Str. 7
D-77933 Lahr

Tel. (07821) 946 02
Fax (07821) 946 220

http://www.sig-positec.de

5960 0100 011, -000, 04.01

Table of contents

5960 0100 011, -000, 04.01

5960 0100 011, -000, 04.01

5960 0100 011, -000, 04.01

5960 0100 011, -000, 04.01

Indexes

Abbreviations

Abbreviation	Explanation
CAN	Controller Area Network, Standardized Bus System
ccd	Command code: Command code, part of an SDO message
COB	Communication Object, base object for transporting data in a CAN network 2048 COB's are permitted in a CAN network and can be identified via a unique COB ID.
COB ID	Communication object identifier, part of the CANopen message for identifying objects and for defining bus access priorities
CS	Command Specifier: Code for identifying LMT services
DS, DSP DSP 402	Draft standard Draft standard proposal Default device profile for drives
EMCY	Emergency object, object for fast transmission of error messages in the network
Inc	Increments
ISO	International Standards Organization
LMT	Layer Management, basic services of the CAN Application Layer
LSB	Lowest significant bit, bit with the lowest value 2^0
MSB	Most significant bit, bit with the highest value, e.g. for a byte that is bit 2^7
NMT	Network Management, services provided by the CAN Application Layer
node ID	node identifier: node address
OSI	Open Systems Interconnection, reference model for data communication, representation as layer model with distributed tasks for each layer
PDO	Process Data Object for fast transmission of data in the CAN network, a difference is made between T_PDO (Transmit PDO) in order to transmit data and R_PDO (Receive PDO) to receive data
SDO	Service Data Object for transmitting system data, a difference is made between T_SDO (Transmit SDO) in order to transmit data and R_SDO (Receive SDO) to receive data
SYNC	Synchronization object, object for synchronizing devices on the network

Product names

Abbreviation	Product designation	Term used
IclA	Positioning drive	Positioning drive

5960 0100 011, -000, 04.01

Technical terms

Broadcast	Type of data transmission in the network, one device sends a message to all devices on the network
Bus arbitration	System in the field bus for avoiding data collision when several devices on the bus transmit simultaneously. The message with the highest priority prevails over other messages if a collision occurs. Priority is defined through the COB ID.
CAN terminal	Communications interface of the positioning controller for connecting to the CAN bus
Client	First the sender, then the receiver of CAN messages in the client-server relationship, starts transmission by transmitting to the server, the point of reference is the server object directory
Consumer	Receiver of CAN messages in the producer-receiver relationship of network devices
Default values	Factory settings, pre-set values when leaving the factory
Electronic data sheet	The electronic data sheet contains technical data on the motor and gearbox, and is transmitted on site.
Error class	Summary of malfunctions in groups corresponding to error responses
Master	First the sender, then the receiver of CAN messages in a master-slave relationship among network devices, the master controls the communication of the slaves.
Node address	Address of a network device, every device on the network has a unique node address.
Parameters	Device data and values which can be set by the user
Producer	Producer of CAN messages in the producer-consumer relationship between network devices
RS422 level	The signal status is determined by means of the differential voltage between a positive signal and an inverted negative signal. Two signal wires must therefore be connected to produce a signal.
Server	First receiver, then sender of CAN messages in the client-server relationship, responds to requests from a client, point of reference is the server object directory.
Slave	First receiver, then sender of CAN messages in a master-slave relationship between network devices, the slave responds to enquiries from a master.
State machine	A state machine defines operating states and transitions with which under CANopen the network or the behavior of network devices can be controlled and changed. The state machine of the NMT services describes the initialization phase and start-up phase for operating the network devices. The state machine for drive units defines control points and setting options for operating a positioning drive or a positioning controller.

5960 0100 011, -000, 04.01

Writing conventions and warning symbols

Action symbol „▶“ This symbol identifies step-by-step instructions which can be carried out in the sequence in which they appear. If the unit shows a recognizable response to a particular step in the instructions, this is shown after the description of the action.In this way you will receive direct confirmation of whether the particular step has been correctly carried out.

Enumeration symbol „•“ The enumeration symbol is used to list the individual points of an information set which is being described.If a sequence of steps or processes is being described, the first step to be carried out is listed first.

This symbol identifies general notes which supply additional information about the unit.

For passages which are preceded by this symbol, it may be necessary to discuss more detailed information with SIG Positec Automation's Service Department.You will find contact addresses for SIG Positec Automation in the manual under „Service address“.

5960 0100 011, -000, 04.01

5960 0100 011, -000, 04.01

1 CAN-Bus and CANopen

1.1 CAN-Bus

The CAN-Bus (CAN: Controller Area Network) was originally developed for fast, cost-effective data transmission in automotive engineering. In the meantime the CAN-Bus is also used in industrial automation, and has been further developed for communication at the field bus level.

Characteristics of the CAN-Bus

The CAN-Bus is a standardized, open bus through which units, sensors and actuators from different manufacturers communicate with each other. Characteristics of the CAN-Bus are

- Multi-master capability
  Every device on the field bus can send and receive data on its own without being dependent on a „controlling“ master functionality.
- Message-oriented communication
  Devices can be integrated into a running network without the whole system having to be re-configured. It is not necessary to announce the address of a new device on the network.
- Prioritization of messages
  For time-critical applications, high-priority messages are transmitted first.
- Residual error probability
  Various safety procedures in the network reduce the probability of a faulty data transmission going undetected to under 10^{-11}. In practical terms, transmission can be assumed to be 100% secure.

Transmission technology

Several network devices are connected in the CAN-Bus via a bus cable. Every network device can send and receive messages. Data between network devices are transmitted serially.

Network devices

Examples of CAN-Bus devices are

- Automation devices, e.g. PLC
- PCs
- Input/output modules
- Drive controllers
- Analytical devices, e.g. a CAN Monitor
- Operating units and input devices as man-machine interface, HMI (HMI: Human Machine Interface)
- Sensors and actuators.

5960 0100 011, -000, 04.01

1.2 CANopen

1.2.1 CANopen descriptive language

CANopen is a device and manufacturer-independent descriptive language for communicating in a CAN-Bus. Although CANopen was originally used in industrial applications to control process sequences, it is now being used in many areas of network communication, e. g. in medical engineering, building automation and vehicle control.

1.2.2 Communication layers

CANopen uses CAN-Bus technology for data communication.

CANopen builds on the basic network services of data communication, following the ISO-OSI layer model. Three layers secure data communication in the CAN-Bus

- CAN Physical Layer
- CAN Data Link Layer
- CANopen Application Layer

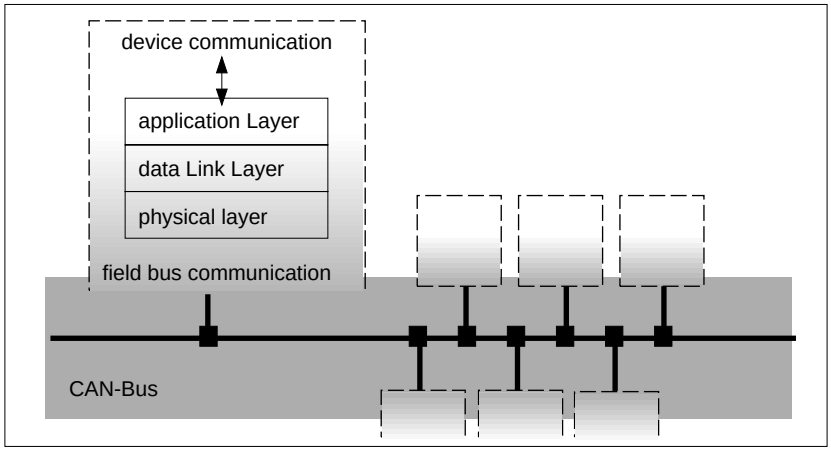

Fig. 1.1 ISO-OSI layer model for the CAN-Bus

CAN Physical Layer The physical layer defines the electrical properties of the CAN-Bus such as the plug connectors, length and type of cables, as well as bit coding and bit timing.

CAN Data Link Layer The data security layer looks after the connection between the devices on a network. It carries out error monitoring and error correction functions, and matches the data flow between sender and receiver.

CANopen Application Layer The application layer provides basic services and functions to enable basic network functions such as establishing and breaking off connections, and reading and writing data to be carried out.

5960 0100 011, -000, 04.01

1.2.3 Objects

All processes under CANopen are carried out via objects. Objects carry out various tasks, serving as communication objects for data transmission to the field bus, controlling the connection procedure or monitoring network devices. Device-specific objects are in direct contact with the device. Device functions can be used and changed through them.

Object dictionary

The central point of connection for all objects is the object dictionary of every network device. Other devices can find here a list of all the objects through which they can make contact with that device. Included in the list are objects for describing the data types and for carrying out communication tasks and device functions under CANopen.

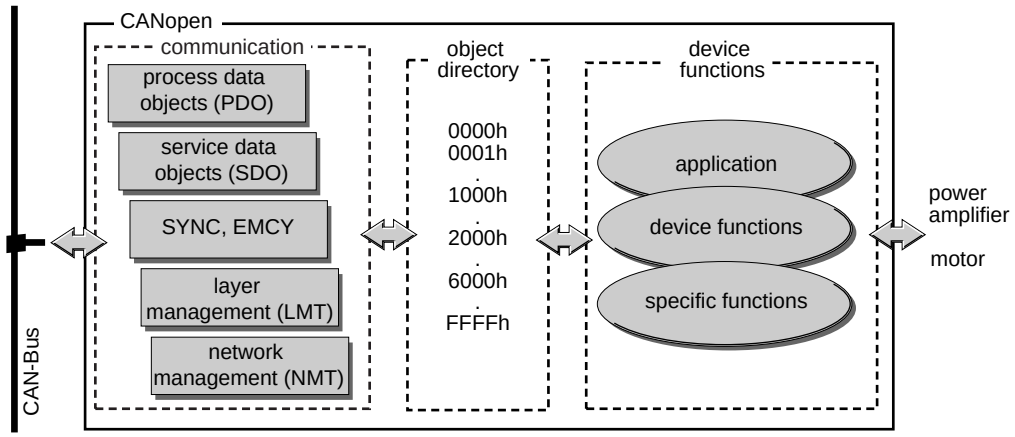

Fig. 1.2 Device model with object dictionary

Object index

Every object is addressed via a 16 bit index, which is represented as a four-digit hexadecimal number. The objects are grouped in the object dictionary.

Index (hex)	Object groups
0000	reserved
0001-009F	Static and complex data types
00A0-0FFF	reserved
1000-1FFF	Communication profile, standardized in the DS 301
2000-5FFF	Manufacturer-specific device profiles
6000-9FFF	Standardized device profiles, e g. in the DSP 402
A000-FFFF	reserved

You will find a list of all the objects which can be used for the device under CANopen in Chapter „Object Directory“ from page 9-1.

Object group: data types

With the data types, messages passing through the network as a stream of bits are assigned the same significance for sender and receiver. They are specified via the data type objects.

Profile object groups

CANopen objects take on various tasks in field bus operation. Profiles put the objects together in accordance with their assigned tasks.

5960 0100 011, -000, 04.01

1.2.4 CANopen profiles

Standardized profiles

Standardized profiles describe objects which can be used on different devices without any additional adaptation. The CAN in Automation Association (CiA) has standardized on various profiles. These include:

- the communications profile DS-301,
- the device profile, „Drives and motion profile“ DSP-402

Both profiles are part of the application layer in the ISO-OSI layer model. Additional services are available through manufacturer-specific device profiles and applications.

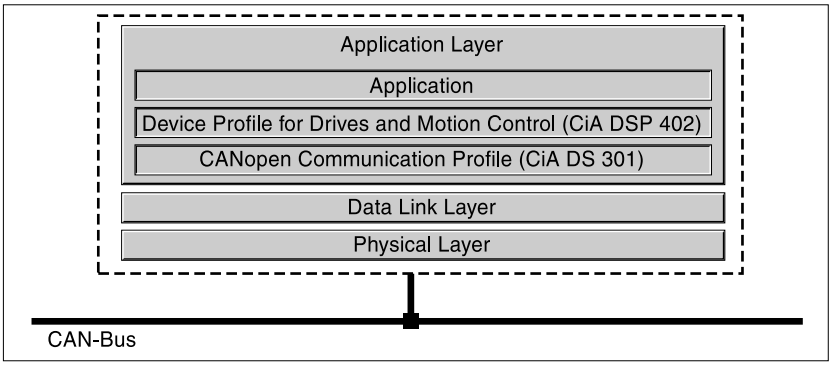

Fig. 1.3 Profiles as part of the application layer

Communications profile DS-301

The CANopen communications profile forms the interface between device profiles and the CAN-Bus. It was specified in 1995 under the name DS-301, and defines uniform standards for the exchange of data between different device types under CANopen.

The objects of the communication profile in the device take on the task of exchanging data and parameters with other network devices, and initialize, control and monitor the device in the network. Objects of the communication profile are:

- Process data objects PDO
- Service data objects SDO
- Objects with special functions for synchronization SYNC, and for error messages and response EMCY
- Network management objects NMT for initializing the device and monitoring it for errors and status.
- Layer management objects LMT for configuring communication parameters.

You will find details on the communication profile objects in Chapter „CANopen Communication“ from page 3-1.

Device profile DSP-402

The device profile, „Drives and motion profile“ DSP-402, describes standardized objects for the positioning, monitoring and setting of drives. Object tasks are

- monitoring devices and supervising status
- standardized parameter setting
- changing, monitoring and executing operating modes

5960 0100 011, -000, 04.01

Manufacturer-specific profiles

The basic functions of a device can be used with the objects of standardized device profiles. The entire functional scope only becomes available with manufacturer-specific device profiles. They define the objects with which the special functions of a device can be used under CANopen.

1.3 Documents and literature

CANopen Standards

CANopen documentation from the Association CAN in Automation (CiA).

- DS 201 to DS 207 CAN Application Layer (CAL)
  Version 1.1, Feb. 1996, The CAN in Automation Association
- CiA Draft Standard 301
  CANopen Application Layer and Communication Profile
  Version 3.0, October 1996, The CAN in Automation Association
- CiA Draft Standard Proposal DSP-402
  CANopen Device Profile for Drives and Motion Control
  Version 1.0, October 1997, The CAN in Automation Association

IclA documents

CANopen manual for positioning drives *IclA*,
SIG Positec Automation, see following table for order numbers

CANopen manual for positioning drives *IclA*,
SIG Positec Automation, see following table for order numbers

Designation	Order Number
Manual, printed, DIN A4 format	
- German (D)	5960 0100 006
- English (GB)	5960 0100 007
- French (F)	5960 0100 008
- Italian (I)	5960 0100 009
CANopen manual, printed, DIN A4 format	
- German (D)	5960 0100 010
- English (GB)	5960 0100 011
- French (F)	5960 0100 012
- Italian (I)	5960 0100 013
CD-ROM with all *IclA* manuals in all languages in pdf format,	5960 0100 014

CAN field bus technology

Controller-Area Network
Basics, Protocols, Building Blocks, Applications

Etschberger, Konrad, Carl Hanser Verlag, Munich Vienna
ISBN 3-446-19431-2

CAN Association

CAN in Automation (CiA)
Am Weichselgarten 26
D-91058 Erlangen
www.can-cia.de

5960 0100 011, -000, 04.01

5960 0100 011, -000, 04.01

2 Field bus operation

2.1 Field bus devices in the CAN-Bus

Different field bus devices from SIG Positec Automation can be operated in the same field bus segment. The commands for a positioning device differ however from those for other SIG Positec Automation devices in the field bus.

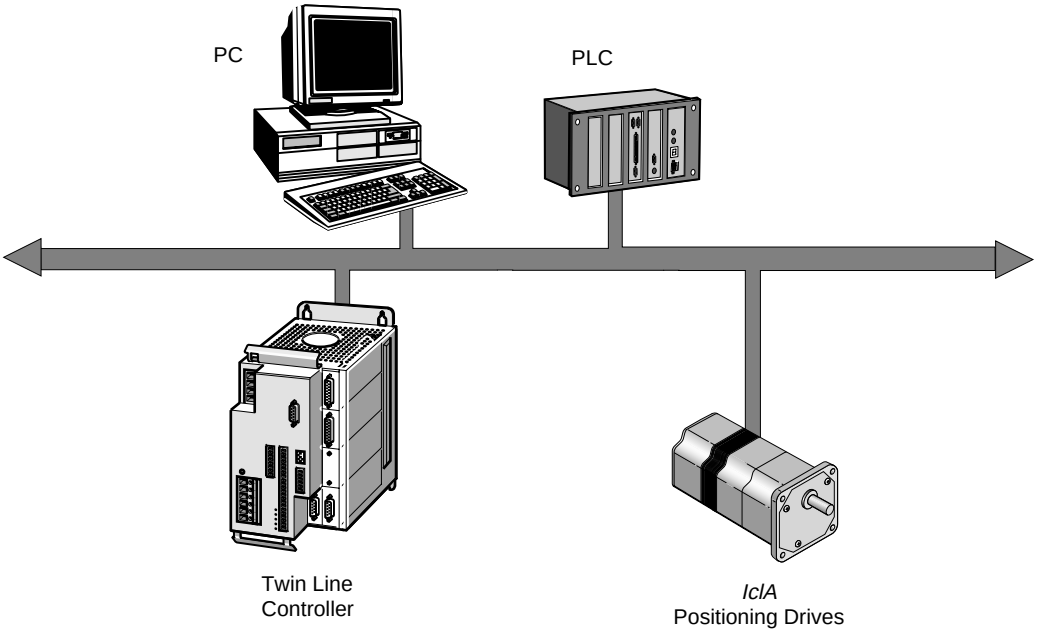

Fig. 2.1 Field bus devices from SIG Positec in a network

2.2 Operating modes and functions in field bus operation

The positioning device works with the following operating modes and functions in field bus operation:

Operating modes:

- Positioning mode with relative and absolute positioning
- Manual movement
- Referencing.

Operating functions:

- Ramp functions
- Emergency stop function
- Monitoring functions.

The parameter settings of the positioning device can be called up and changed via the field bus, signal interface inputs can be monitored and diagnostic and error monitoring functions set to work.

5960 0100 011, -000, 04.01

5960 0100 011, -000, 04.01

3 CANopen Communication

3.1 Communication profile

CANopen carries out the communication between network devices via object directories and objects. A network device can use Process Data Objects (PDO) and Service Data Objects (SDO) in order to read the object data from the object dictionary of another device and - if permitted - write them back with different values.

Accessing the objects of the network devices enables parameter values to be swapped, movement functions of individual CAN-Bus devices to be initiated or e. g. status information to be requested.

3.1.1 Object dictionary

Every CANopen device administers an object dictionary in which all the objects required for CANopen communication with the device are listed.

Index, Sub-index

The objects are addressed in the object dictionary via a 16 bit long index. One or more 8 bit long sub-index entries for each object specify individual data fields in the object. Index and sub-index are shown in hexadecimal notation, identifiable by the attached „h“.

The following example shows index and sub-index entries for the *software position limit* object (607Dh) for identifying software limit switch positions.

Index	Sub-index	Name	Explanation
607Dh	00h	-	Number of data fields
607Dh	01h	min. position limit	Lower value limit switch
607Dh	02h	max. position limit	Upper value limit switch

Dictionary set-up

The objects are fitted into the object dictionary structure, sorted by index values. The following table shows an overview of the object dictionary in accordance with the CANopen specification.

Index range (hex)	Object groups
0000h	reserved
0001h-001Fh	static data types
0020h-003Fh	complex data types
0040h-005Fh	manufacturer-specific data types
0060h-007Fh	static data types for device profiles
0080h-009Fh	complex data types for device profiles
00A0h-0FFFh	reserved
1000h-1FFFh	communication profile [1)]
2000h-5FFFh	manufacturer-specific profiles [1)]
6000h-9FFFh	standardized device profiles [1)]
A000h-FFFFh	reserved

1) supported by positioning device

5960 0100 011, -000, 04.01

Object descriptions in the manual

For CANopen programming with the positioning device, the objects in the following object groups are described separately:

- 1xxxh objects: Communication objects in this Chapter
- 6xxxh objects: standardized device profile objects in the Chapters entitled „Operate states and Operating modes“ and „Operating Functions“.
- 2xxxh objects: manufacturer-specific objects in so far as they are needed for controlling the device, in the Chapters entitled „Operate states and Operating modes“ and „Operating Functions“.

Standardized objects

Standardized objects form the basis for using identical application programs for different network clients on one device type. The only condition is that the devices list the objects in their directories. Standardized objects are defined in the CANopen profiles, in the communication profile and in various device profiles which are arranged by device groups.

3.1.2 Communication objects

Overview

Communication objects are standardized in the CANopen communication profile, DS-301. The objects can be divided into four groups corresponding to their tasks:

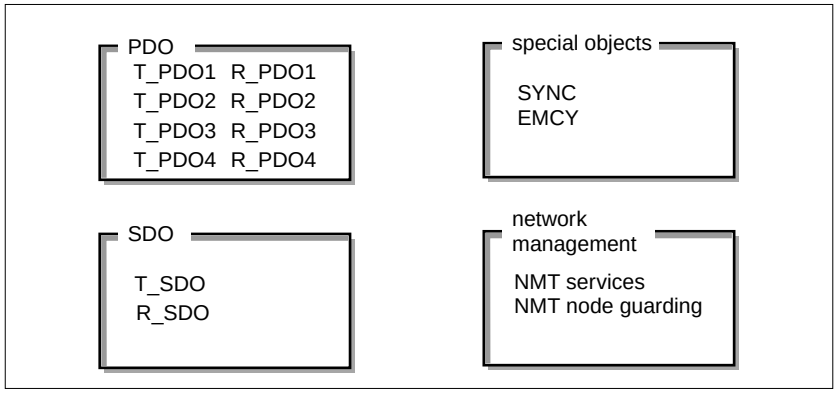

Fig. 3.1 Communication objects, seen from the perspective of the device: T_..: „Transmit“, R_..: „Receive“

- Process Data Objects (PDO) for real-time transmission of process data
- Service Data Objects (SDO) for read/write access to the object dictionary
- Objects for the synchronization and error messaging of CAN nodes:

  SYNC object (synchronization object) for synchronizing network devices

  EMCY object (emergency object) for displaying errors in a device or in its periphery.

- Network and Layer Management Services:

  NMT services for initialization and network control (NMT: network management)

  NMT guarding objects for monitoring network devices

  LMT services (LMT: layer management) with objects for setting node addresses and transmission rates

5960 0100 011, -000, 04.01

CAN Message

Data are exchanged on the CAN-Bus in the form of CAN messages. A CAN message transmits the communication object and multiple management and control information that are responsible for error- and loss-free data transfer.

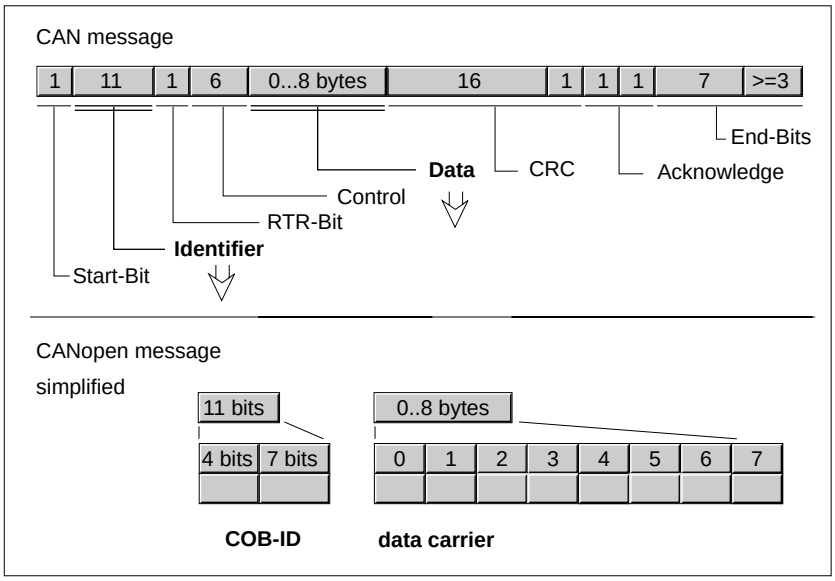

Fig. 3.2 CAN message CANopen message in simplified form

CANopen message

For work with CANopen objects and data exchange, the CAN message can be shown in a simplified form, as most of the bits are used to ensure that data transmission is free of error. These bits are automatically removed from the received message by the data security layer, the data link layer of the layer model, and inserted before a message is sent.

The two bit fields, „Identifier“ and „Data“ make up the simplified CANopen message. The „Identifier“ corresponds to the „COB-ID“ and the „Data“ field to the data carrier of a CANopen message which can be max. eight bytes in size.

COB-ID

The COB-ID (Communication object identifier) fulfils two tasks in controlling communication objects:

- Bus arbitration: Defining transmission priorities
- Identification of communication objects

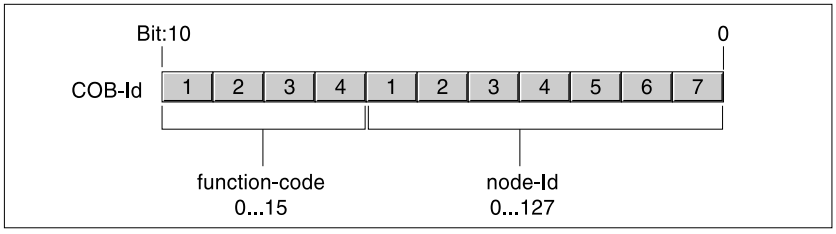

Fig. 3.3 COB-ID with function code and node address

An 11 bit COB Identifier is defined for the positioning device in accordance with the CAN 2.0A specification, consisting of two parts:

- Function code, 4 bits in size
- Node ID, 7 bits in size.

5960 0100 011, -000, 04.01

Function code

The function code classifies the communication objects. As the function code bits in the COB-ID are more significant, the function code also controls transmission priorities at the same time: objects with a small function code are assigned high priority in transmission, e. g. an object with the function code „1“ will be transmitted before an object with the function code „3“, if both access the bus at the same time.

Node address

Every network device is configured before network operation. In the process it is assigned a unique, seven bit long node address (node-ID) between 1 and 127 (7Fh). The device address „0“ is reserved for „broadcast“ transmissions, in which messages are sent simultaneously to all devices on the network.

COB-IDs of Communication Objects

The following table shows the COB-IDs of all communication objects in accordance with their factory settings. The „Index of object parameters“ column indicates the index of special objects with which the communication objects settings can be read or changed by SDO.

Communication object	Function code	Node address (node-ID) [1...127]	COB-ID decimal (hexadecimal)	Index of object parameters
NMT start/stop service	0 0 0 0	0 0 0 0 0 0 0	0	–
SYNC object	0 0 0 1	0 0 0 0 0 0 0	128 (80h)	1005h....1007h
EMCY object	0 0 0 1	x x x x x x x	128 (80h) + node-ID	1014h
T_PDO1	0 0 1 1	x x x x x x x	384 (180h) + node-ID	1800h
R_PDO1	0 1 0 0	x x x x x x x	512 (200h) + node-ID	1400h
T_PDO2	0 1 0 1	x x x x x x x	540 (280h) + node-ID	1801h
R_PDO2	0 1 1 0	x x x x x x x	768 (300h) + node-ID	1401h
T_SDO	1 0 1 1	x x x x x x x	1408 (580h) + node-ID	1200h
R_SDO	1 1 0 0	x x x x x x x	1536 (600h) + node-ID	1200h
NMT error control	1 1 1 0	x x x x x x x	1792 (700h) + node-ID	100Ch..100Eh
LMT Services	1 1 1 1	1 1 0 0 1 0 x	2020 (7E4h), 2021 (7E5h)	
NMT Identify Service [1)]	1 1 1 1	1 1 0 0 1 1 0	2022 (7E6h)	
DBT Services [1)]	1 1 1 1	1 1 0 0 x x x	2023 (7E7h), 2024 (7F8h)	
NMT Services [1)]	1 1 1 1	1 1 0 1 0 0 x	2025 (7E9h), 2026 (7EAh)	

1) not supported by the positioning device

Example of COB-ID selection

For a device with the node address 5, the COB-ID of the communication object T_PDO1 is:

384+node-ID = 384 (180h) + 5 = 389 (185h).

Data carrier

The data carrier of the CANopen message can take up to eight bytes of data. Besides the data carrier for SDOs and PDOs, special carrier types are also defined in the CANopen profile:

- Error data carrier
- Remote data carrier for requesting a message
- LMT protocol carrier

Data carriers are described by the relevant communication objects.

5960 0100 011, -000, 04.01

3.1.3 Communication Relationships

CANopen uses three relationships for communication between network devices:

- The Master-Slave relationship
- The Client-Server relationship
- The Producer-Consumer relationship

The Master-Slave relationship

A „Master“ in the network controls the traffic of messages. A „Slave“ only responds to requests from the master.

The Master-Slave relationship is used with network management objects in order to ensure controlled network start-up and to monitor network device connections.

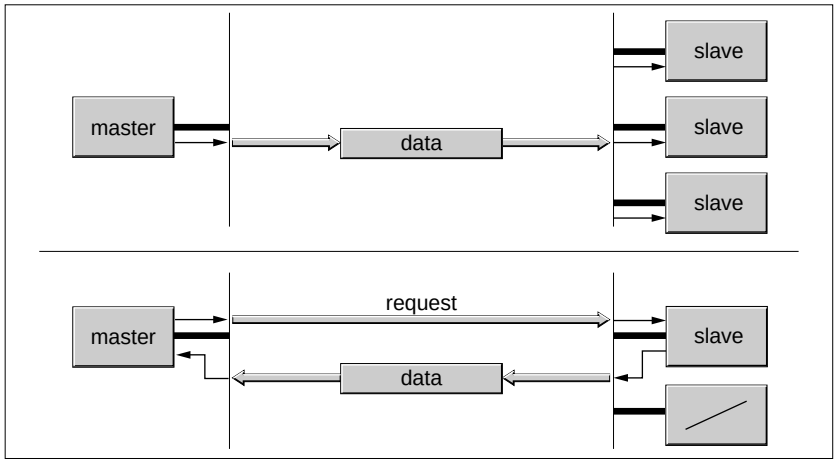

Fig. 3.4 The Master-Slave relationship

The exchange of messages can be carried out with or without confirmation. If the master sends an unconfirmed CAN message, it may be received by one, several or no slaves.

In order to obtain a message confirmation, the master requests message data from the slave.

The Client-Server relationship

A Client-Server communication is established between two nodes. The server is the device whose object list is used during the exchange of data. The client addresses and initiates the exchange of messages, and expects confirmation from the server.

A Client-Server relationship is used with SDOs in order to transmit configuration data and long messages.

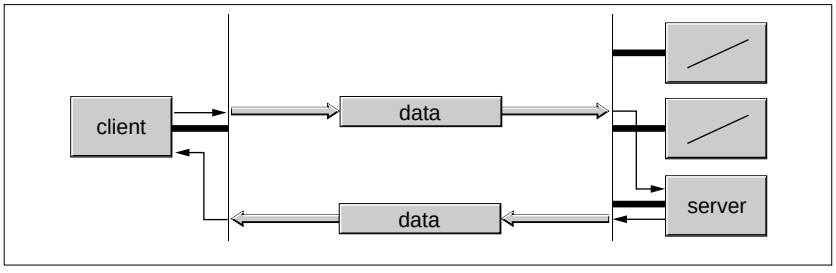

Fig. 3.5 The Client-Server relationship

5960 0100 011, -000, 04.01

The client addresses and transmits a CAN message to a server. The server evaluates the message and sends reply data by way of confirmation.

The Producer-Consumer relationship

The Producer-Consumer relationship is used for the exchange of messages containing process data, as the relationship allows fast exchange of data without any overhead data.

A producer sends data, a consumer receives them.

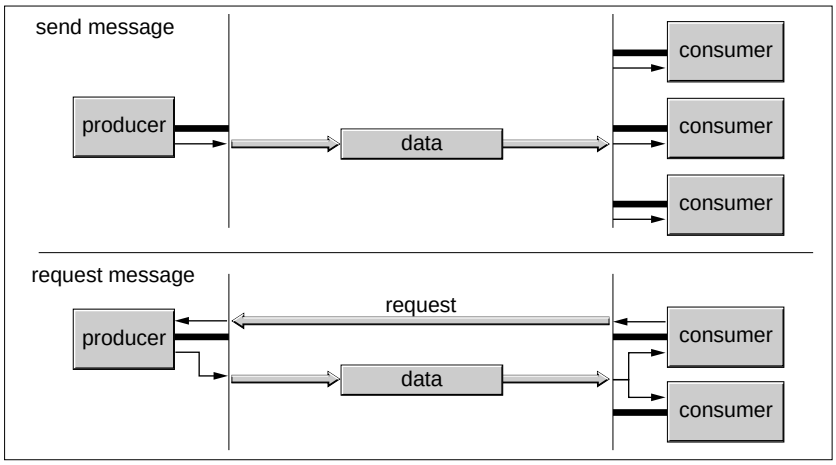

Fig. 3.6 The Producer-Consumer relationship

The producer sends a message which may be received by one, several or no network devices. It receives no confirmation of receipt. Message transmission can be triggered by

- an internal event, e. g. by a „target position reached“ signal
- through the synchronization object SYNC
- by a request from a consumer.

You will find details on the function of the Producer-Consumer relationship and on requesting messages in the section „Process data communication“ from page 3-16.

5960 0100 011, -000, 04.01

3.2 Service data communication

3.2.1 Overview

Service Data Objects *(SDO)* can be used to access the entries in an object dictionary through its index and sub-index. The values of the objects can be read and - if permitted - also changed.

Every network device has at least one server-SDO in order to be able to respond to read or write requests from another device. A client-SDO is only needed in order to request SDO messages from the object dictionary of other devices, or to change them there.

Data are sent by means of T_SDO in the client or server-SDO, and received by means of R_SDO.

SDOs have a higher COB-ID than PDOs and are therefore transmitted on the CAN-Bus with a lower priority status.

3.2.2 SDO data exchange

A Service Data Object transmits parameter data between two network devices. The exchange of data follows the client-server relationship. The server is the device to which an SDO message refers.

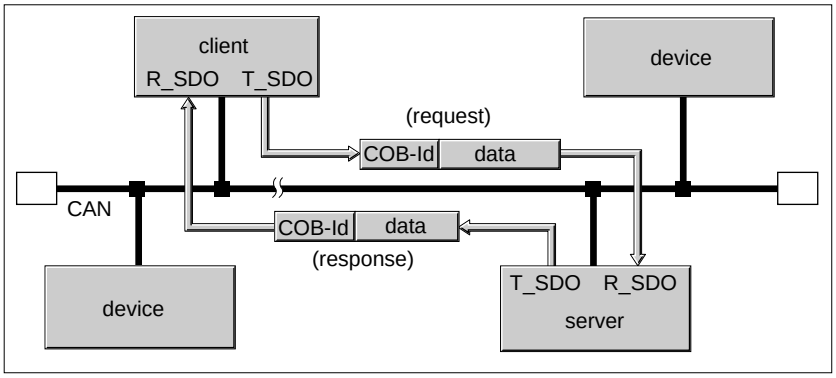

Fig. 3.7 SDO message exchange with request and response

Message types The client-server communication is initiated by the client in order to transfer parameter values to the server or to fetch them from the server. In both cases, the client initiates communication with a request and receives a response from the server.

5960 0100 011, -000, 04.01

3.2.3 SDO message

The positioning drive can transmit SDO messages of any length. Here, we differentiate between the following messages :

- Messages with a data length ≤ 4 bytes
- Messages with a data length > 4 bytes

If the message length is greater than 4 bytes it must be split up into several telegrams. The first telegram transmits the first 4 bytes of the message. The remainder of the message is transmitted in one or multiple additional messages which can have a length of up to 7 bytes.

Data length ≤ 4 bytes

An SDO message with a data length of up to 4 bytes consists, simplified, of the COB ID and of the SDO data frame that can be used to transfer up to 4 bytes of data.

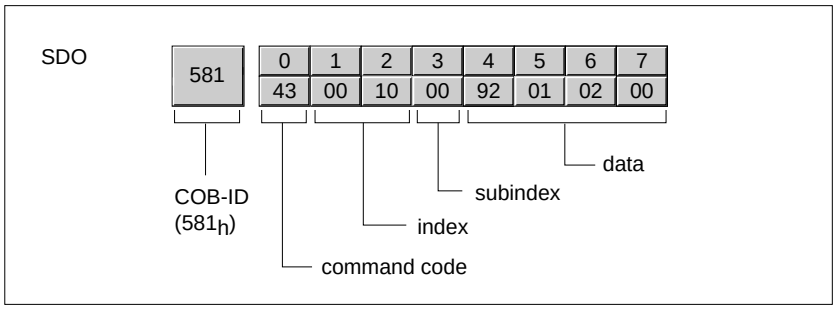

Fig. 3.8 Example of an SDO message

COB-ID

R_SDO and T_SDO have different COB-IDs, see table on page 3-4. SDO settings can be read via the *server SDO parameters* object (1200h).

Data carrier

The data carrier of an SDO message consists of

- the command code (ccd: command code), in which the SDO message type and the length of the value being transmitted are encrypted.
- the index and sub-index which point to the object whose data are being transported in the SDO message. In the event of an error, the faulty SDO is itself specified in the index and sub-index.
- data which can comprise up to four bytes.

Evaluation of numerical values

Index and data are transmitted flush left in Intel format. If the SDO contains numerical values of over a byte in length, the data must be converted byte by byte before and after transmission.

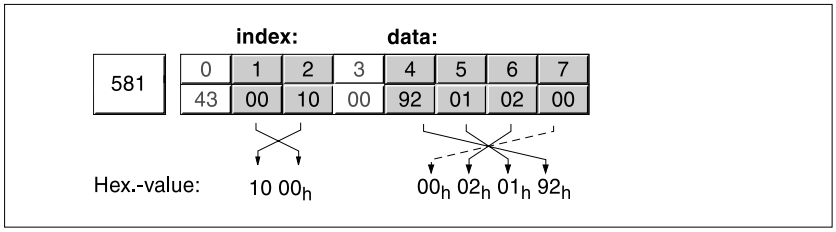

Fig. 3.9 Conversion of numerical values larger than 1 byte

5960 0100 011, -000, 04.01

Data length > 4 bytes

SDO messages with a data length greater than 4 bytes, for example, 8-byte values of the data type “visible string 8” are split up into multiple telegrams. The structure of the first telegram is the same as an SDO message with ≤ 4 bytes. Follow-up messages do not contain an index or subindex, thus increasing the space for transmitting data.

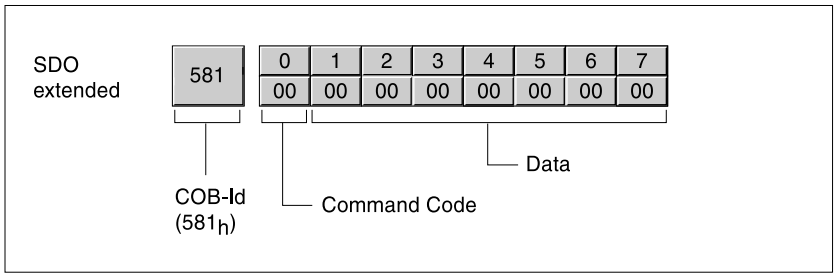

Fig. 3.10 Example of a telegram of an extended SDO message

The first 4 bytes are transmitted in the first SDO message. The remaining bytes are respectively transmitted in blocks with a length of maximum 7 byte. When transferring data > 4 bytes, the data are not converted byte for byte. Rather, they are transmitted continuously in ASCII format.

Data frame Extended SDO Message

The data frame of an extended SDO message consists of:

- Command Code (ccd: command-code), containing encrypted data of the SDO message type, a so-called “Toggle bit“ (means, transitions from one state to another), the data length of the transmitted value and an end-of-message bit.
  The toggle bit values for each one of the telegrams of an extended SDO message must continuously toggle between 0 and 1, that is, the toggle bit of the first telegram is “0”, that of the second telegram is “1”, that of the third telegram is again “0” etc.
  The bit indicating the end of a complete SDO message always has the status “0”, save for the last telegram of the SDO message, where it is toggled to the value “1”.
- Data containing up to seven bytes.

5960 0100 011, -000, 04.01

3.2.4 Reading and writing data with a length of smaller or equal 4 bytes

Writing data

The client initiates a write request by communicating index, sub-index, data length and value.

The server sends a response confirming whether the data have been correctly processed. The response contains the same index and sub-index, but not data.

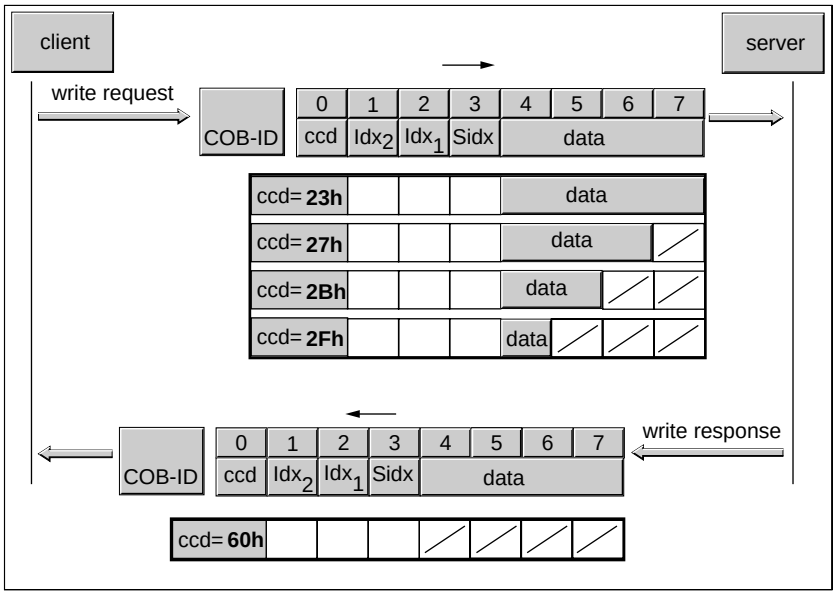

Fig. 3.11 Writing a parameter value

Unused bytes in the data field are shown in the graphic as an oblique stroke. Their contents are not defined.

ccd coding

The following table shows the command code used for writing parameter values. It varies with the type of message and length of data transmitted.

Messages	**Length of data used**				**Explanation**
	4 bytes	**3 bytes**	**2 bytes**	**1 byte**	
write request	23h	27h	2Bh	2Fh	transmit parameters
write response	60h	60h	60h	60h	response
error response	80h	80h	80h	80h	error

5960 0100 011, -000, 04.01

Error response

If a message could not be evaluated without errors the server transmits an error message containing the telegram “Abort SDO Transfer”.

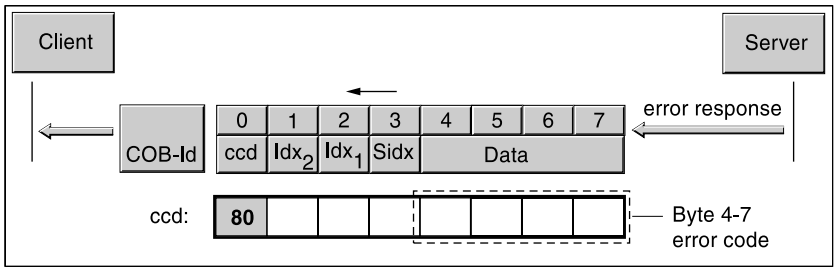

Fig. 3.12 Error response, cause of error encoded in bytes 4 ..7 (error code)

You will find details on evaluating error responses in the section „SDO error message“ on page 7-8.

Reading data

The client initiates a read request by communicating the index and sub-index which point to the object or object value whose value it wishes to read.

The server confirms the request by sending the required data. The SDO response contains the same index and sub-index. The length of the response data is given in the command code „ccd“.

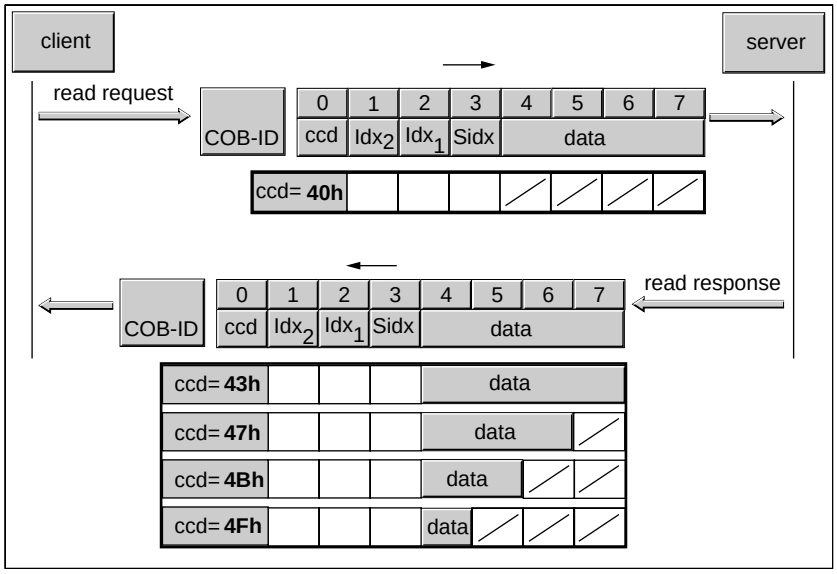

Fig. 3.13 Reading a parameter value

Unused bytes in the data field are shown in the graphic as an oblique stroke. Their contents are not defined.

5960 0100 011, -000, 04.01

ccd coding

The following table shows the command code used for transmitting a read value. It varies with the type of message and length of data transmitted.

Messages	Length of data used				Explanation
	4 bytes	3 bytes	2 bytes	1 byte	
read request	40h	40h	40h	40h	request read value
read response	43h	47h	4Bh	4Fh	return read value
error response	80h	80h	80h	80h	error

Error response

If a message could not be evaluated without errors the server transmits an error message containing the telegram “Abort SDO Transfer”. For details on debugging error messages refer to the section „SDO error message“ on page 7-8.

5960 0100 011, -000, 04.01

3.2.5 Reading data with a length greater than 4 bytes

If an SDO message is to be transferred that contains values with a length exceeding 4 bytes it must be split up into multiple telegrams. Every telegram consists of 2 parts:

- Request by the SDO client,
- Confirmation by the SDO server.

The SDO client's request contains the control segment "ccd" with the toggle bit and a data segment. The acknowledgement telegram also contains a toggle bit in segment "ccd". The toggle bit of the first telegram is "0", in the follow-up telegram it toggles between "1" and "0".

Reading data

The client starts a read request by transmitting the index and subindex that points to the object or object value he wants to read out.

The server acknowledges this request by transmitting the index, subindex, data length and the first 4 bytes of the requested data. Command code indicates that that data with a length > 4 bytes have been transmitted. The command code for the read response of the server following the first message is 41h.

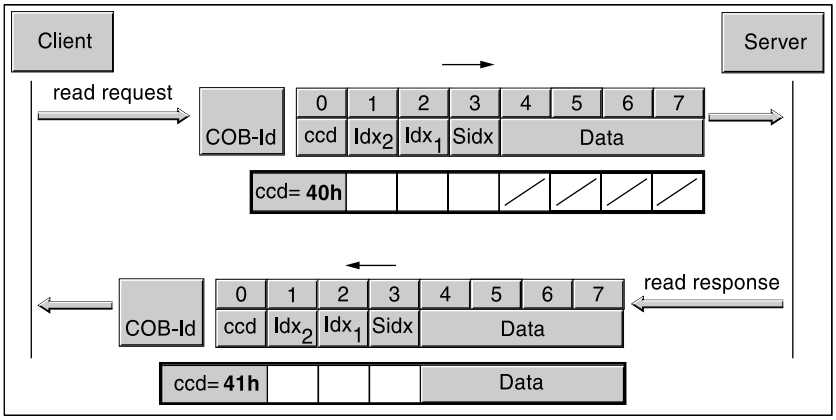

Fig. 3.14 Sending the first message

The remaining data is requested in the next telegrams and transmitted in 7-byte packets from the server.

5960 0100 011, -000, 04.01

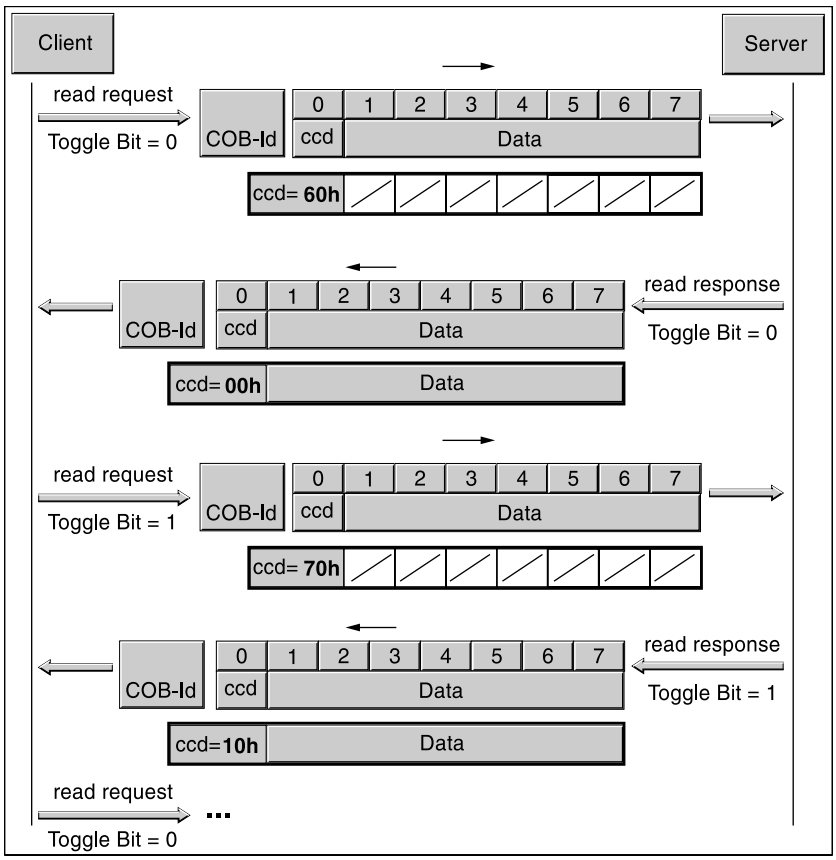

Fig. 3.15 Reading parameter values with telegrams of the extended message

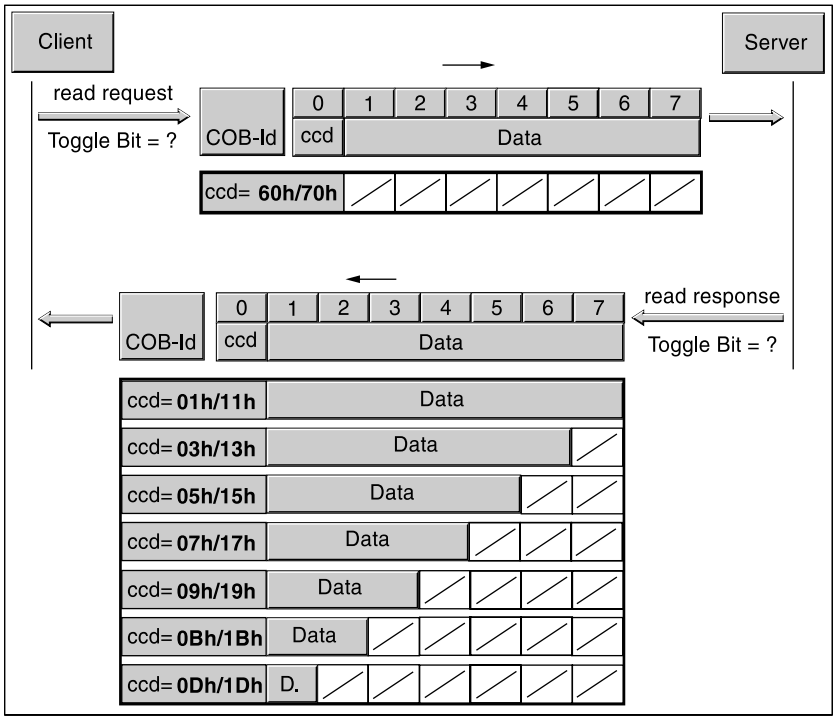

Fig. 3.16 Transmit last message

The graphic display indicates unused bytes in the data field with a dash. Their content is not defined.

5960 0100 011, -000, 04.01

ccd encoding The table below shows the command code required for the transmission of a read value. It is dependent on the message type, the toggle bit value, transmitted data length and the value of the bit indicating the end of the complete SDO message.

Message type	Used data length							Meaning
	7 bytes	**6 bytes**	**5 bytes**	**4 bytes**	**3 bytes**	**2 bytes**	**1 byte**	
read request Toggle Bit = 0	60h	60h	60h	60h	60h	60h	60h	Acknowledgement with Toggle Bit = 0
read request Toggle Bit = 1	70h	70h	70h	70h	70h	70h	70h	Acknowledgement with Toggle Bit = 1
read response Toggle Bit = 0	00h	–	–	–	–	–	–	Sending parameters with Toggle Bit = 0
read response Toggle Bit = 1	10h	–	–	–	–	–	–	Sending parameters with Toggle Bit = 1
read response to the last message Toggle Bit = 0	01h	03h	05h	07h	09h	0Bh	0Dh	Send parameters with last message and Toggle Bit = 0
read response to the last message Toggle Bit = 1	11h	13h	15h	17h	19h	1Bh	1Dh	Send parameters with last message and Toggle Bit = 1
error response	80h	80h	80h	80h	80h	80h	80h	Error

Error response If a message could not be evaluated without errors the server transmits an error message containing the telegram “Abort SDO Transfer”. For details on debugging error messages refer to the section „SDO error message“ on page 7-8.

5960 0100 011, -000, 04.01

3.3 Process data communication

3.3.1 Overview

Process Data Objects (PDO: process data object) are used for real-time data exchange of process data such as the actual or set positions or operating status of the device. Transmission can be carried out very quickly because no additional administration data are sent and because no response is required from the receiver.

The flexible data length of a PDO message also increases the data throughput. A PDO message can transmit up to eight bytes of data. If only two bytes are used, only two bytes are transmitted.

The length of a PDO message and assignment of the data fields is defined via the PDO mapping procedure. You will find information on PDO mapping on page 3-20.

PDO messages can be swapped between network devices which produce or work with process data.

3.3.2 PDO data exchange

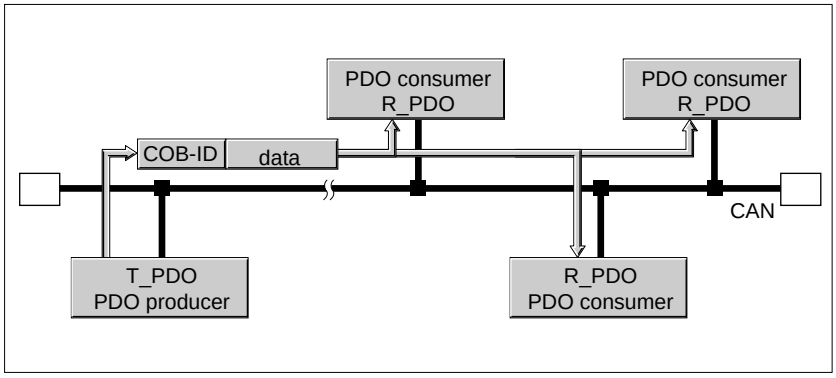

Fig. 3.17 PDO message exchange

Data exchange with PDOs follows the producer-consumer relationship and can be triggered in three ways

- synchronized
- event-driven, asynchronously
- at the request of a consumer, asynchronously

The task of controlling how the synchronized data are processed, is performed by the SYNC object. Synchronous PDO messages are transmitted immediately like all other PDO messages, but they are not evaluated until the next SYNC. Synchronized data exchange allows several drives to be started at the same time, for example.

PDO messages which are called up on request or as a result of an event, are evaluated by the network device immediately. For example, an emergency stop message should be sent asynchronously to enable the device to carry out an immediate shutdown.

The transmission type can be set separately for every PDO via subindex 02h (transmission type).The objects are listed in the table on page 3-19.

5960 0100 011, -000, 04.01

Requesting messages

One or more network devices with consumer functions can request PDO messages from a producer.The producer is identified via the COB-ID in the request, and it responds with the required PDO.

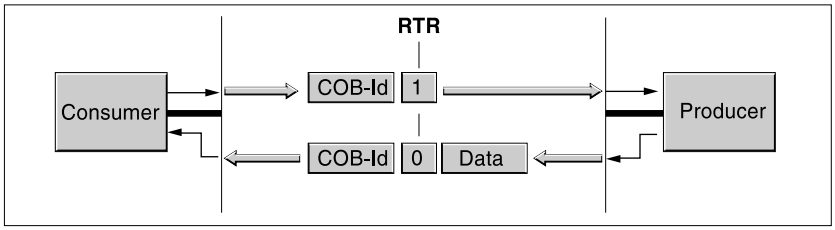

Fig. 3.18 Requesting a message with RTR=1

The RTR bit in a CAN message is used to detect a request (RTR: remote transmission request). The COB-ID remains the same for both messages:

RTR=0: transmission of data

RTR=1: request for data.

Setting the RTR request

Each PDO can be set separately to define whether it should respond to RTR requests or not. The objects for this are listed in the table on page 3-19. Sub-index 02h of the object defines the transmission type. The PDO only responds to a request by RTR bit if RTR transmission has been switched on for that PDO. The sub-index values for using the RTR bit are:

Objects 1400h/1401h, 1800h/1801h Sub-index 02h, „transmission type“	**Explanation**
252	RTR activated, synchronous
253	RTR activated, asynchronous

You will find an overview of all values relating to sub-index 02h in the object dictionary for the relevant object.

The positioning device cannot request PDOs, but it can respond to PDO requests.

5960 0100 011, -000, 04.01

3.3.3 PDO Messages

T_PDO, R_PDO

There is one PDO each available for sending and receiving PDO messages.

- The T_PDO to transmit PDO messages (T: transmit),
- The R_PDO to receive PDO messages (R: receive).

The number of T_PDOs and R_PDOs used by the device, can be determined via the *number of PDOs supported* object (1004h).

The positioning device uses four PDOs:

- an asynchronous receive PDO, R_PDO1
- a synchronous receive PDO, R_PDO2
- an asynchronous transmit PDO, T_PDO1
- a synchronous transmit PDO, T_PDO2

Receive PDOs

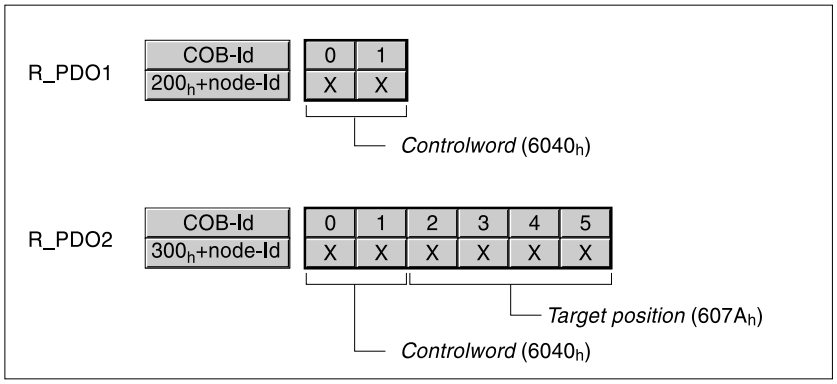

Fig. 3.19 receive PDOs R_PDO1 and R_PDO2

R_PDO1

The first receive PDO contains the status machine's control word, object *Controlword* (6040h), with which the operating status of the drive can be set.

R_PDO1 is evaluated asynchronously.

R_PDO2

The second receive PDO is used to receive the control word and the target position of a movement command, the *Target position* (607Ah) object.

The positioning device processes R_PDO2 synchronously with reception of the next SYNC object. R_PDO2 can be used for the synchronized start-up of several drives.

You will find details on the SYNC object in the section „Synchronization“ from page 3-21, and information on the control word of the status machine in the Chapter entitled „Operate states and Operating modes“ on page 5-7.

5960 0100 011, -000, 04.01

Transmit PDOs

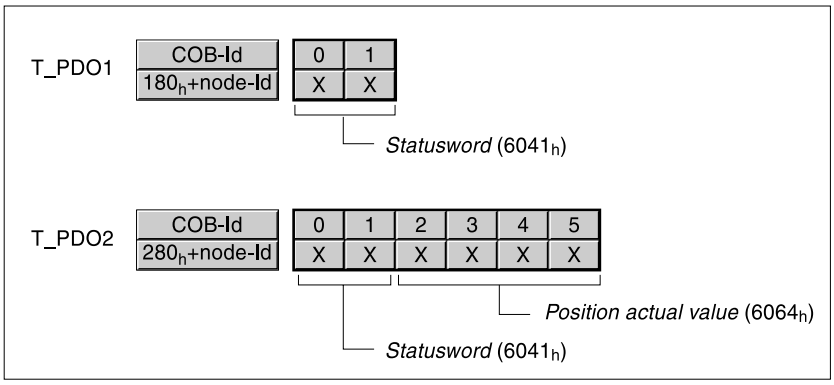

Fig. 3.20 Transmit-PDO T_PDO1 and T_PDO2

T_PDO1 The first transmit PDO contains the status word, the *Statusword* object (6041h), of the status machine.

T_PDO1 is transmitted on an event-driven basis, every time the status word changes, and is therefore asynchronous.

T_PDO2 The second transmit PDO contains the status word and the current position of the motor, the *Position actual value* object (6064h).T_PDO2 is transmitted after a SYNC object has been received.

You will find information on the status word of the status machine in the Chapter entitled „Operate states and Operating modes“ on page 5-8.

PDO Settings PDO settings can be read and changed by means of four communication objects:

Object	**Explanation**
1st receive PDO parameter (1400h)	Settings for R_PDO1
2nd receive PDO parameter (1401h)	Settings for R_PDO2
1st transmit PDO parameter (1800h)	Settings fur T_PDO1
2nd transmit PDO parameter (1801h)	Settings for T_PDO2

5960 0100 011, -000, 04.01

3.3.4 PDO mapping

Up to 8 bytes of data from different parts of the object dictionary can be transmitted in a PDO message. Showing the data in a PDO message is termed PDO mapping.

A distinction is made between static and dynamic PDO mapping.

Static PDO mapping shows the data from different objects in accordance with a defined, invariable setting. Dynamic PDO mapping allows flexible selection of different process data during operation.

The positioning device carries out static PDO mapping for the four PDOs R_PDO1, R_PDO2, T_PDO1 und T_PDO2.

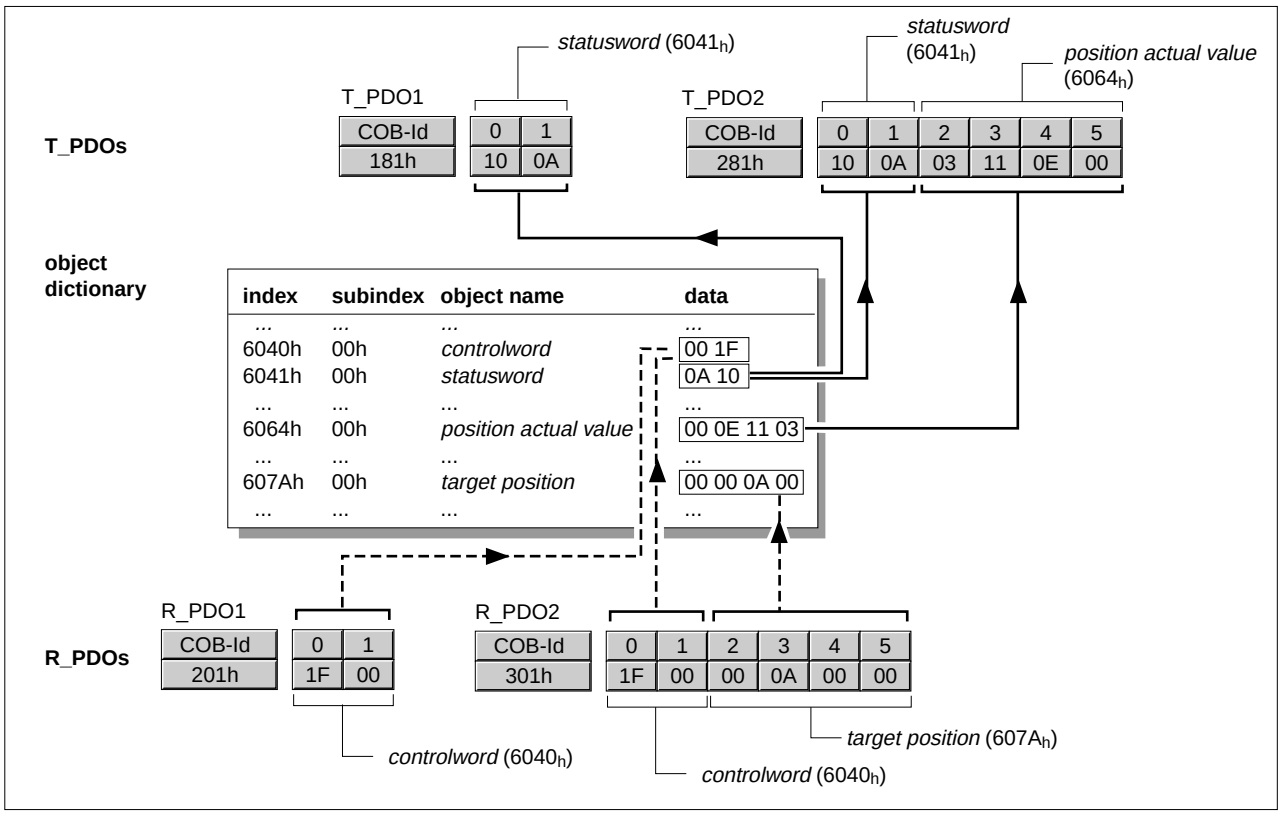

Fig. 3.21 PDO mapping, here for a network device with node address = 1

Communication objects for PDO mapping

Every PDO is assigned an object in the object dictionary for PDO mapping. The table shows PDO mapping objects for both R_PDOs and T_PDOs. As the positioning device only supports static PDO mapping, the object data cannot be changed.

Object	Explanation
1st receive PDO mapping (1600h)	Object for R_PDO1
2nd receive PDO mapping (1601h)	Object for R_PDO2
1st transmit PDO mapping (1A00h)	Object for T_PDO1
2nd transmit PDO mapping (1A01h)	Object for T_PDO2

5960 0100 011, -000, 04.01

3.4 Synchronization

The SYNC synchronization object controls the synchronous exchange of messages between network devices, e. g. to enable several drives to be started up simultaneously.

The exchange of data follows the producer-consumer relationship. The SYNC object is sent to all devices by a network device, and can be evaluated by all devices which support synchronous PDOs.

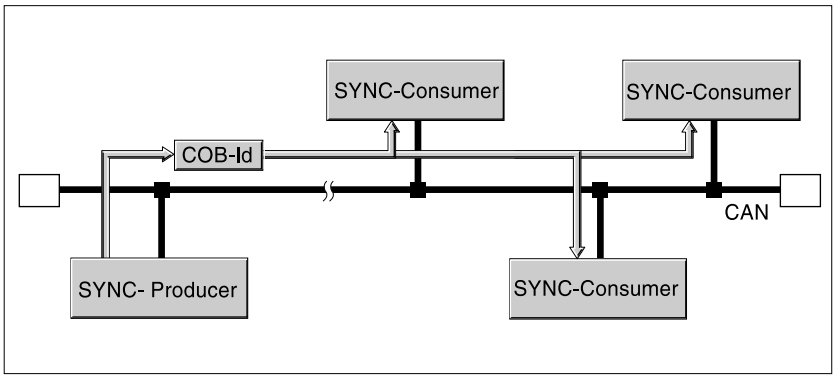

Fig. 3.22 SYNC message

Synchronous data transmission

Viewed from the SYNC receiver, first the status data is transmitted in a T_PDO, then the new control data is received via R_PDO. The control data is not processed until the next SYNC message is received. The SYNC object itself does not transfer data.

Cyclical and acyclical transmission

The synchronous exchange of messages can be carried out cyclically or acyclically.

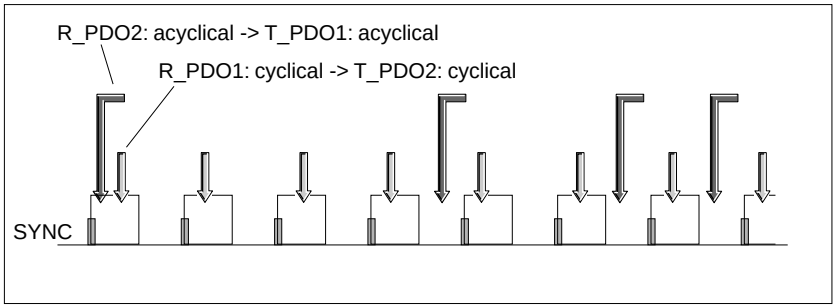

Fig. 3.23 Cyclical and acyclical transmission

With cyclical transmission PDO messages are exchanged continuously in a defined cycle, e. g. with every SYNC message.

If a synchronous PDO message is transmitted acyclically, it can be sent or received at any time, but only becomes valid when the next SYNC message arrives.

Whether a PDO operates cyclically or acyclically is stored in the *transmission type* sub-index (02h) of the relevant PDO parameter, e. g. for R_PDO1 in the *1st receive PDO parameter* object (1400h), sub-index 02h.

5960 0100 011, -000, 04.01

COB-ID, SYNC object

In order to ensure fast transmission of the SYNC object, it is sent without confirmation and with high priority.

The COB-ID of the SYNC object is set to the standard value of 128 (80h). This value can be changed after the network has been initialized, by means of the *COB-ID SYNC Message* object (1005h).

„Motorstart“ PDO

The R_PDO2 of the positioning device in basic position is received synchronously and acyclical. R_PDO2 is the „Motorstart“ PDO with which a synchronous start-up of the drive can be carried out.

5960 0100 011, -000, 04.01

3.5 Emergency service

The Emergency service reports internal device faults over the CAN-Bus. The error message is sent with an EMCY object to all network devices in accordance with the consumer-producer relationship.

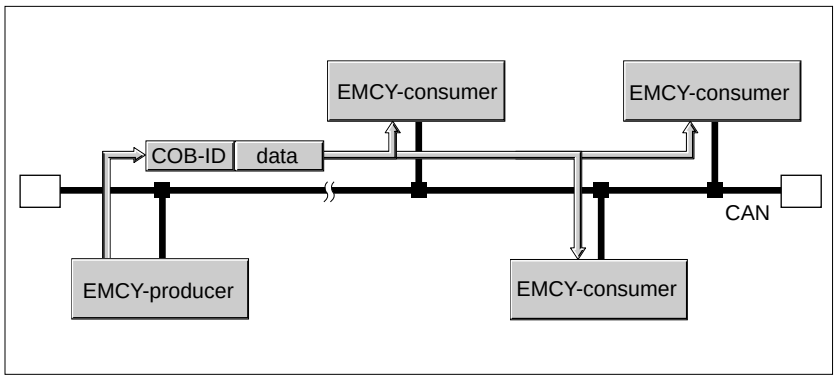

Fig. 3.24 Error messages via EMCY objects

Boot-Up message

The communications profile, DS 301, Version 3.0, defines an additional task for the EMCY object: sending a boot-up message. A boot-up message informs all network devices that the device sending the message is ready for operation in the CAN network.

A boot-up message consists of the COB-ID of the EMCY object, and is transmitted without any data. The standard setting of the COB-ID is 128 (80h)+ node-Id.

5960 0100 011, -000, 04.01

3.5.1 Error evaluation and error handling

EMCY message

If an device fault or warning occurs, the device switches into error status in accordance with the CANopen status machine, see the Chapter entitled „CANopen state machine“ on page 5-2. At the same time it sends an EMCY message with the error register and error code.

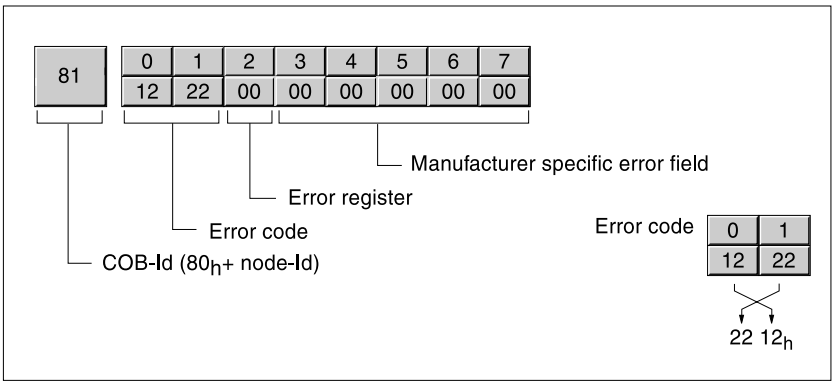

Fig. 3.25 EMCY message

Byte 0, 1 - Error code: value also stored in the *Error code* object (603Fh).

Byte 2 - Error register: value also stored in the *Error register* (1001h) object.

Bytes 3-7, manufacturer-specific error field: not supported by the positioning device.

COB-ID

The COB-ID is calculated from the node address for every device in the network which supports an EMCY object.

COB-ID = Function code EMCY object (80h) + node-ID

The function code of the COB-ID can be changed by means of the *COB-ID emergency* object (1014h).

Error register and error code

The error register reports the error status of the device in bit-coded form. Bit 0 remains set as long as the error remains. The other bits designate the error type. The exact cause of the error can be determined through the error code. The error code is transmitted as a 2-byte value in Intel format, and must be converted byte by byte for evaluation.

You will find a list of all error messages as well as device responses and remedial measures in the Chapter entitled „Diagnostics and Error Correction“ from page 7-1 .

Error memory

The positioning device stores the error register in the *Error register* object (1001h), and the last error to have occurred in the *Error code* object (603Fh). At the same time the last five error messages are saved in the *Pre-defined error field* object (1003h). The report of the last error to have occurred can be found in sub-index 01h of the memory, older reports in higher memory locations in sub-index 02h to 05h. All error entries are deleted if the drive is switched off or if the sub-index *number of errors* (00h) of the object is set to „0“.

5960 0100 011, -000, 04.01

3.6 Network management services

Network Management (NMT) is part of the CANopen communication profile, and is used to initialize the network and network devices as well as to start, stop and monitor the devices during network operation.

NMT services are carried out in a master-slave relationship. The NMT master addresses individual NMT slaves via their node address. A message with the node address „0“ goes to all NMT slaves simultaneously.

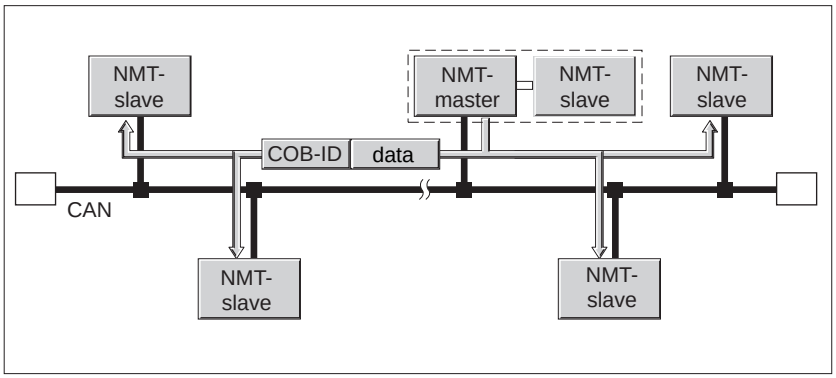

Fig. 3.26 NMT services via the master-slave relationship

The positioning device can only take on the function of an NMT slave.

NMT Network Services

NMT Services can be divided into two groups:

- Services for monitoring devices, for initializing devices for CANopen communication and for controlling the behavior of devices during network operation,
- Services for monitoring connections in order to ensure error-free network operation

5960 0100 011, -000, 04.01

3.6.1 NMT Services for monitoring devices

NMT Status Machine

The NMT status machine describes the initialization and operating states of an NMT slave in network operation.

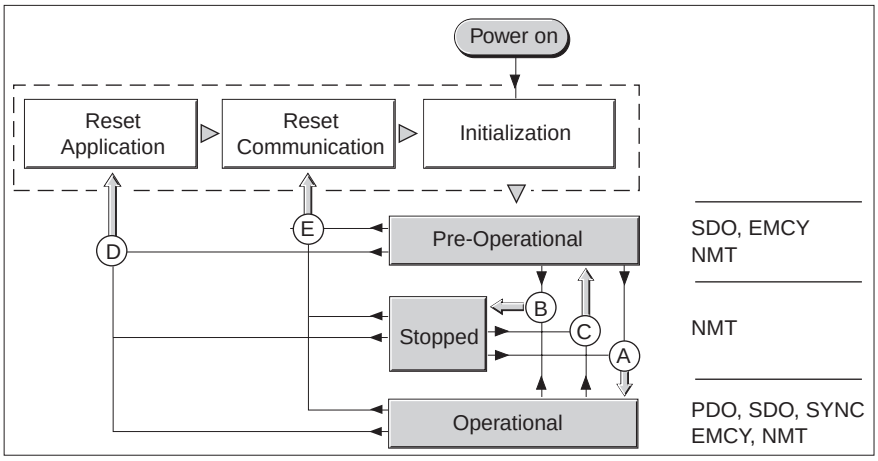

Fig. 3.27 NMT Status Machine and available communication objects

All the communication objects which can be used in the relevant network status, are shown on the right-hand side of the graphic.

Initialization

When the power supply is switched on (power on), an NMT slave automatically goes through an initialization phase which prepares it for CAN-Bus operation. After initialization is finished, the slave switches to „Pre Operational“ status and sends a boot-up message by EMCY object, containing no data. From now on, an NMT master can control the operational behavior of the NMT slave in the network by using five NMT services, shown as the letters A to E in the above graphic.

NMT Service	Transition	Explanation
Start remote node	A	Switch to „Operational“ status Commence normal network operation with all devices
Stop remote node	B	Switch to „Stopped“ status Stop device communication in the network. If connection monitoring is activated, it remains on
Enter Pre-Operational	C	Enter „Pre-Operational“ status All communication objects can be used apart from PDOs. The „Pre-Operational“ status can be used for configuration by SDOs: - PDO mapping - Start of synchronization - Start of connection monitoring - Execution of LMT services
Reset node	D	Switch to „Reset application" status Load stored device profiles data and switch automatically to „Reset communication“ status after the „Pre-Operational“ status
Reset communication	E	Switch to „Reset communication" status Load stored communication profiles data and switch automatically to „Pre-Operational“ status

5960 0100 011, -000, 04.01

Remanent data

If the power supply is switched on (Power on), the device loads the remanent object data. If the object data have previously been reset by means of the *Restore default parameters* object (1011h), the default values are loaded from the device data sheet.

NMT message

The NMT services for monitoring devices are transmitted as unconfirmed messages with the COB-ID = 0. They therefore receive as standard the highest transmission priority on the CAN-Bus.

The NMT device service data carrier consists of two bytes.

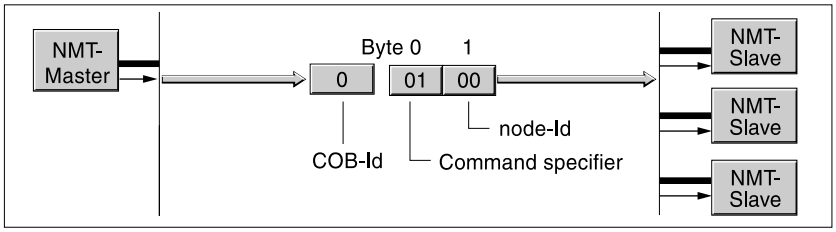

Fig. 3.28 NMT message

The first byte, the „Command specifier“, specifies the NMT service used.

Command Specifier	NMT Service	Transition
1 (01h)	Start remote node	A
2 (02h)	Stop remote node	B
128 (80h)	Enter Pre-Operational	C
129 (81h)	Reset node	D
130 (82h)	Reset communication	E

The second byte addresses the receiver of the NMT message with a note address between 1 and 127 (7Fh). A message with the node address „0“ is addressed to all NMT-Slaves.

5960 0100 011, -000, 04.01

3.6.2 NMT services for connection monitoring

The connection monitoring function monitors the communication status of network devices to enable the system to react to the failure of a device or to an interruption in the network.

There are two connection monitoring NMT services available with the positioning device under CANopen:

- „Node guarding“ for monitoring the connections of an NMT slave
- „Life guarding“ for monitoring the connections of an NMT master

Node guarding

In carrying out its „node guarding“ and „life guarding“ monitoring functions, the NMT master requests an NMT status signal at regular intervals and expects to receive a response by the end of the interval. The interval is set via the parameter „guard time“.

The following graphic shows an error message generated due to not having received a response from the NMT slave by the end of the third cycle.

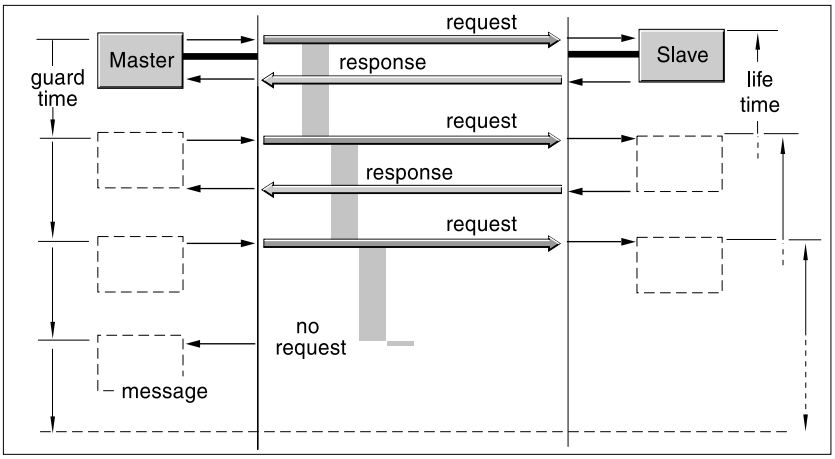

Fig. 3.29 „Node guarding“ and „Life guarding“ with intervals

The NMT master reports a connection fault to the upstream master program if

- the slave does not respond by the end of the „guard time“ interval
- if the slave's NMT status has changed without any prompting by the NMT master.

Life guarding

If „Life guarding“ is activated, the NMT master must send its enquiry to the slave within the adjustable „life time“ period.If the period is exceeded, the NMT slave assumes that the NMT master has failed, and signals a connection fault. The „life time" counter is restarted with every enquiry.

Intervals

The „guard time“ and „life time“ intervals are set via two communication objects:

- the „guard time“ interval via the *Guard time* object (100Ch)
- the „life time“ interval as a factor of the „guard time“ interval via the *Life time factor* object (100Dh)

  „life time“ = *guard time* (100Ch) x *life time factor* (100Dh)

5960 0100 011, -000, 04.01

COB-ID

Connection monitoring is carried out with the aid of the communication object *NMT error control* (700h+node-ID). The COB-ID is worked out from the node address for every NMT slave:

COB-ID = Function code *NMT error control* (700h) + node-ID.

The function code of the COB-ID can be changed in the „Pre-Operational“ NMT status using the *COB-ID guarding protocol* object (100Eh).

NMT message

The RTR bit of a CAN message is used to recognize a request (RTR: remote transmission request). The COB ID remains the same for both messages.

On receiving a request from the NMT master, the NMT slave responds by sending a data byte.

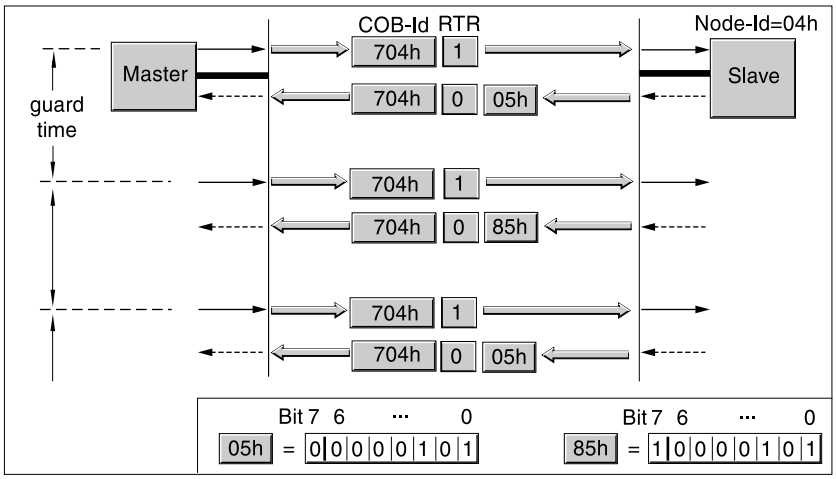

Fig. 3.30 Response of the NMT slave

Bits 0 to 6 designate the NMT status of the slave:

4 (04h): „Stopped“

5 (05h): „Operational“

127 (7Fh) : „Pre-Operational“

Bit 7 changes its status between „0“ and „1“ after every „guard time" interval, to enable the NMT master to recognize and ignore any second response in the same interval period. The first request at the start of the connection monitoring function, begins with bit 7 = 0.

Connection monitoring may not be activated during the initialization phase of a device. The status of bit 7 is reset as soon as the device passes the NMT status „Reset communication“.

Connection monitoring continues to run in the NMT status „Stopped“.

5960 0100 011, -000, 04.01

3.7 Layer Management services

LMT services (LMT: Layer Management) can be used to set the node address and transmission rate of network devices over the CAN-Bus. Devices which support the LMT configuration do not require hardware DIP switches.

LMT services are part of the CAN Application Layer and are described in DS 205-1 and DS205-2.

3.7.1 LMT data transmission

LMT objects

LMT services work on the principle of the master-slave relationship, and use two objects for communication.

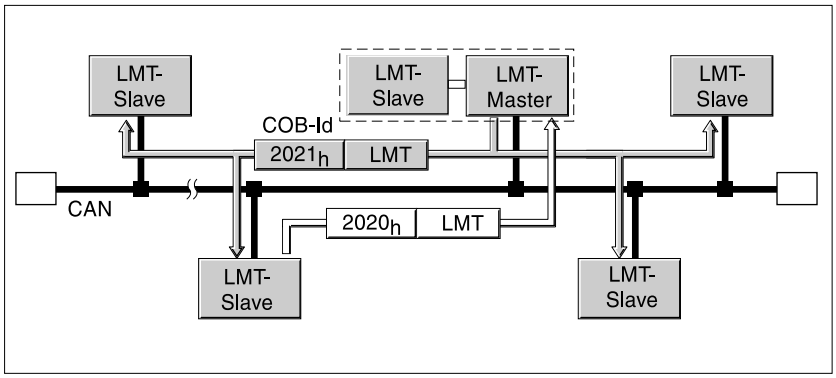

Fig. 3.31 LMT Master-Slave relationship

The LMT master transmits to one or all LMT slaves with the LMT master object, COB-Id=2021 (7E5h). If the LMT master sends a message to an LMT slave, the slave responds via the LMT slave object, COB-Id = 2020 (7E4h). LMT messages which are addressed to all LMT slaves, remain unconfirmed.

Attributes of an LMT slave

An LMT slave is specified via three attributes:

- LMT Address
- LMT Mode,
- LMT Class

An LMT slave is identified by its LMT address. The LMT mode displays whether the LMT slave is operating in configuration mode or in normal network operation. The LMT class gives information on the LMT services which an LMT slave supports. The positioning device is designed as an LMT „Class 2 slave“ and therefore supports all LMT services required for configuration.

5960 0100 011, -000, 04.01

3.7.2 Configuration and operation mode

In order to configure an LMT slave, it must be switched from the operation mode in which it is operating in line with its pre-assigned task, into configuration mode. Two LMT services take care of the change between operation mode and configuration mode.

- „Switch mode selective“ switches an LMT slave from operation mode to configuration mode.
- „Switch mode global“ switches all LMT slaves simultaneously into configuration or operation mode, selectable via the „mode“ parameter.

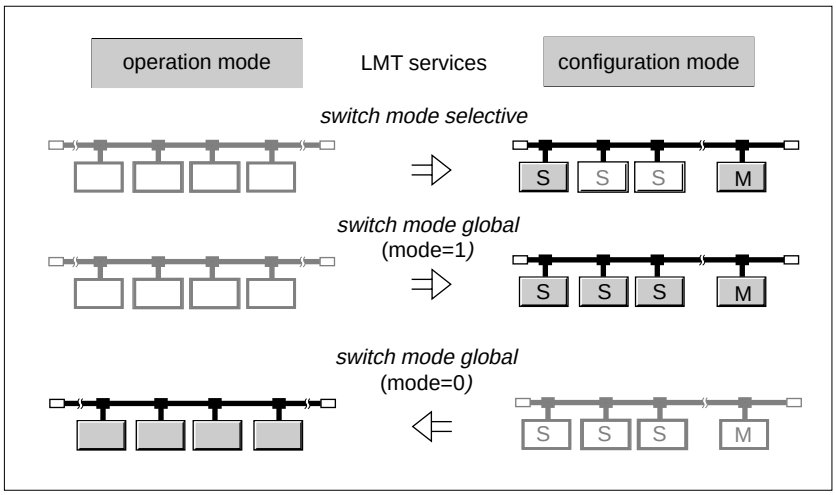

Fig. 3.32 LMT services for changing between operation mode and configuration mode

Both LMT services „Switch mode selective“ and „Switch mode global“ can be activated in normal network operation - in operation mode. All other LMT services can be carried out in configuration mode.

Setting configuration mode

Selection of the correct LMT service for changing into configuration mode depends on the LMT services which are to be carried out in configuration mode.

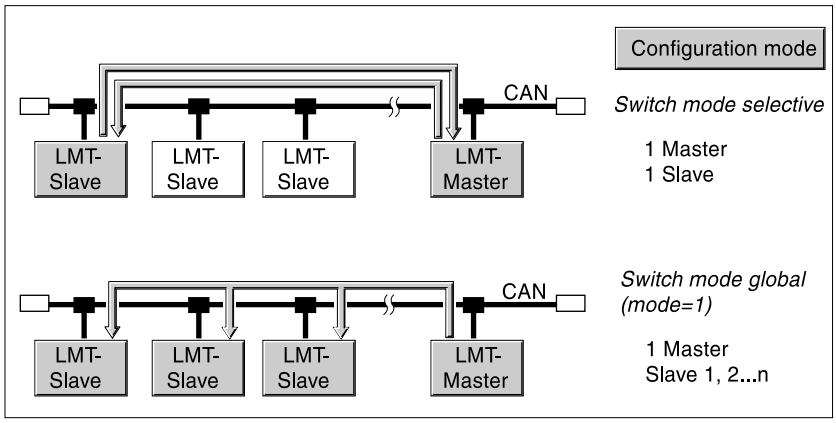

Fig. 3.33 LMT services with one or all LMT slaves in configuration mode

LMT services which have to be confirmed by the LMT slave, may only be carried out with an LMT slave in configuration mode. These include, for example, the LMT services for setting the node address and transmis-

5960 0100 011, -000, 04.01

sion rate. Other LMT services address all LMT slaves simultaneously. For these services configuration mode can be switched on with the LMT service „Switch mode global“.

3.7.3 LMT services in the positioning device

„Command specifier“ CS

LMT services are identified via the „Command specifier“ CS. CS is the first byte of the eight byte long data carrier of an LMT message.

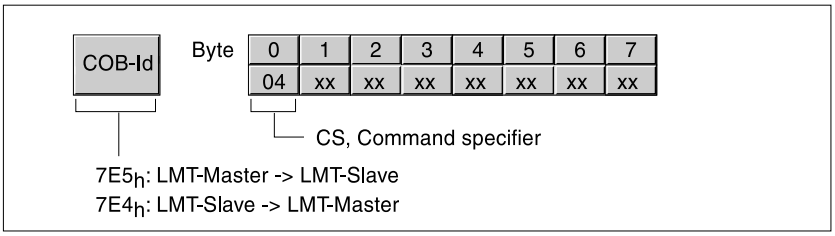

Fig. 3.34 LMT message, here LMT service „Switch mode global“, „xx“ are bytes with no significance for the choice of service

LMT services implemented in the positioning device and their „Command specifiers“ are:

CS	LMT service	Explanation	Slave
1 (01h) 2 (02h) 3 (03h)	Switch mode selective	switch a NMT slave into configuration mode, address the slave through the manufacturer’s name (1), product name (2) and serial number (3)	n
4 (04h)	Switch mode global	switch all LMT slaves into operation mode or configuration mode	n
17 (11h)	Configure modules ID	Assign a new node address to LMT slave	1
19 (13h)	Configure bit timing parameters	Set transmission rate for an LMT slave	1
21 (15h)	Activate bit timing parameters	Activate the new baud rate settings	n
23 (17h)	Store configuration	Store LMT configuration	1

The table shows in the „slave“ column whether only one LMT slave is allowed to operate in configuration mode for that LMT service (1) or all connected slaves (n).

5960 0100 011, -000, 04.01

Changing LMT mode

The LMT service „Switch mode global“ (CS=04h) switches all LMT slaves simultaneously into configuration or operation mode, depending on the parameter „mode“.

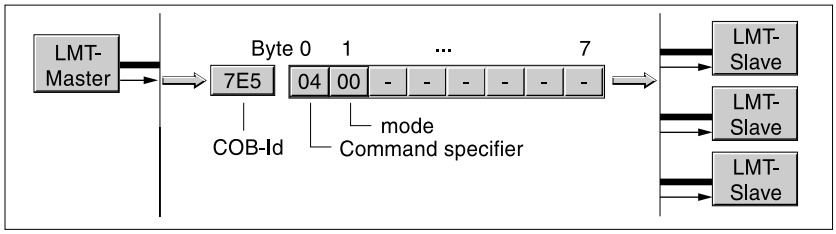

Fig. 3.35 LMT service „Switch mode global“

- Byte 0: CS=04h
- Byte 1: mode
  0: switch into operation mode
  1: switch into configuration mode.

The LMT message is transmitted without confirmation.

LMT Address

In order to switch only one LMT slave, the LMT address of the slave must be known. It is made up of three invariable parts which are stored in the device and which identify every device on the network unambiguously. Each part of the address must be transmitted in the following sequence with a separate „command specifier“.

- Manufacturer's name (manufacturer's name), ASCII encoded, CS=01h
- product name, ASCII encoded, CS=02h
- serial number, BCD encoded, CS=03h

LMT messages are transmitted without confirmation. After receiving the complete LMT address, the device automatically switches to configuration mode.

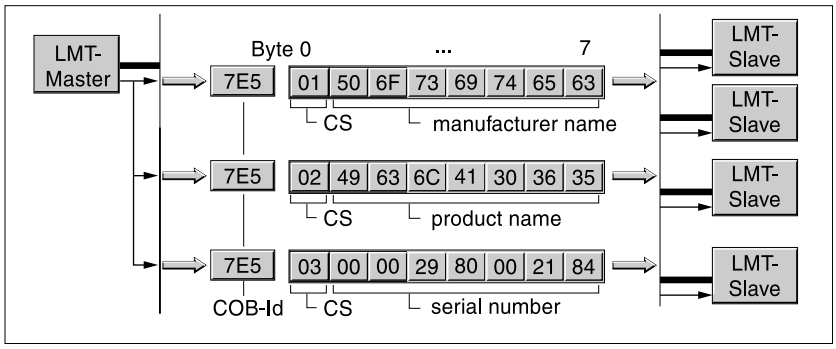

Fig. 3.36 LMT service „Switch mode selective“ with example data

The above graphic results in the following:

- Manufacturer's name „Positec“ (50h, 6Fh, 73h, 69h, 74h, 65h, 63h)
- Product name „IclA065“ (49h, 63h, 6Ch, 41h, 30h, 36h, 35h)
- Serial number „2980002184“ (00h, 00h, 29h, 80h, 00h, 21h, 84h)

„Switch mode global“ (CS=04h), mode=0 is used to switch back into operation mode.

The serial number can be read from the device by SDO via the *drive serial number* object (200Fh).

5960 0100 011, -000, 04.01

3.7.4 Node address

Setting NMT node address

The NMT node address (node-ID) is set by means of the LMT service „Configure modules ID“ (CS=11h). A value between 1 and 127 (7Fh) is transmitted in the „node-ID“ byte as the new node address.

The LMT service is transmitted with confirmation, and may only be carried out on an LMT slave in configuration mode. You will find an example of assigning a node address in the Chapter entitled „Set-up“ on page 4-3.

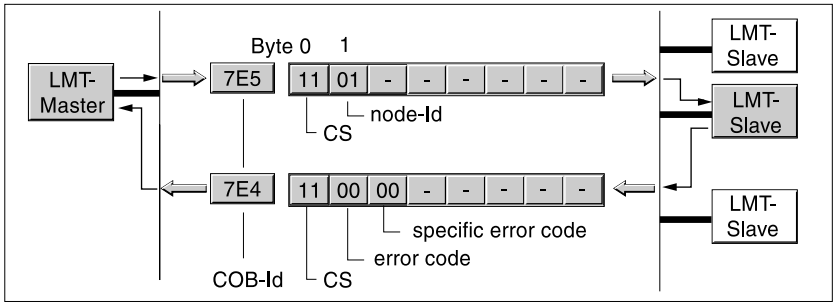

Fig. 3.37 LMT service „Configure modules ID“, here with node-ID=01h

Setting NMT node address, response from LMT slave

The LMT slave confirms the setting of the new node address or reports an error.

- Byte 0: CS=11h.
- Byte 1: error code:
  0: message correctly executed
  255: error has occurred
- Byte 2: specific error code
  If „error code“ = 255: information on error message

When it leaves the factory, the node address for the positioning device is pre-set to „127“.

Storing settings

The LMT service „store configuration“ (CS=17h) is used to save a changed setting, and this setting is then available every time the device is switched on.

The LMT service is transmitted with confirmation and may only be carried out on a slave in configuration mode.

At the start of the LMT service, only the „command specifier" is entered and communicated in the LMT message.

Storing setting, Response of LMT slave

- Byte 0: CS=17h
- Byte 1: error code
  0: message correctly executed
  1: Storing values is not supported
  255: error has occurred
- Byte 2: specific error code
  Information on error message, if „error code“ = 255.

5960 0100 011, -000, 04.01

3.7.5 Transmission rate

Two LMT services are used for setting the transmission rate. The baud rate is selected with the first but not yet activated. The second service makes sure that all the devices switch baud rates simultaneously, to enable them to communicate with each other shortly afterwards.

Changing transmission rate

The LMT service „Configure bit timing parameters“ changes the baud rate of a device in accordance with a baud rate table. At least one baud rate table is stored in the device.

- Byte 0: CS= 13h
- Byte 1: table selector
  Selecting the table for setting the transmission rate
  0: standardized table
  128..255: manufacturer-specific tables, not supported by the Positioning device.
- Byte 2: table index
  Index entry from the relevant table for selecting a baud rate

The values in the standard table are:

Table index	Baud rate [kBaud]	Table index	Baud rate [kBaud]
0	1000	5	100
1	800	6	50
2	500	7	20
3	250	8	10
4	125		

The LMT service is transmitted with confirmation and may only be carried out on an LMT slave in configuration mode. You will find an example of setting the transmission rate in the chapter entitled „Set-up“ on page 4-3.

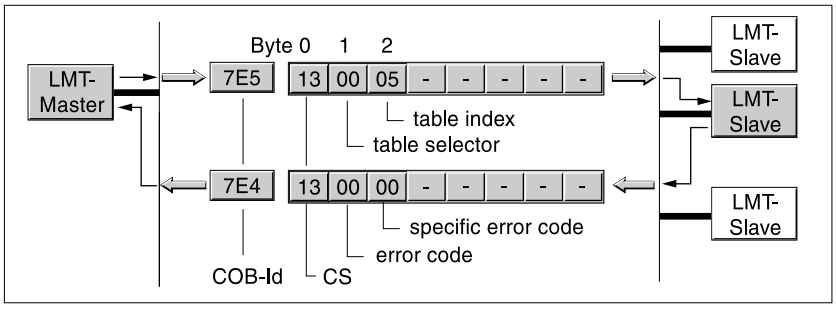

Fig. 3.38 LMT service “Configure bit timing parameters“

Changing the transmission rate Response of LMT slave

- Byte 0: CS=13h
- Byte 1: error code
  0: message correctly executed
  1: baud rate not supported
  255: error has occurred
- Byte 2: specific error code
  Information on error message, if „error code“ = 255.

When it leaves the factory, the transmission rate for the positioning device is pre-set to 20 Kbit/sec.

5960 0100 011, -000, 04.01

Storing settings

The LMT service „store configuration“ (CS=17h) is used to save the setting, and this setting will be available every time the device is switched on.

Activating transmission rate

If a new transmission rate has been set using the LMT service „Configure bit timing parameters“ (CS=13h), it must be changed simultaneously for all network devices using the service „Activate bit timing parameters“ (CS=15h).

Transmitting a delay, the „switch delay“ to all devices, guarantees that no device is transmitting at the old baud rate while other devices are already receiving at the new rate. The LMT message is transmitted without confirmation.

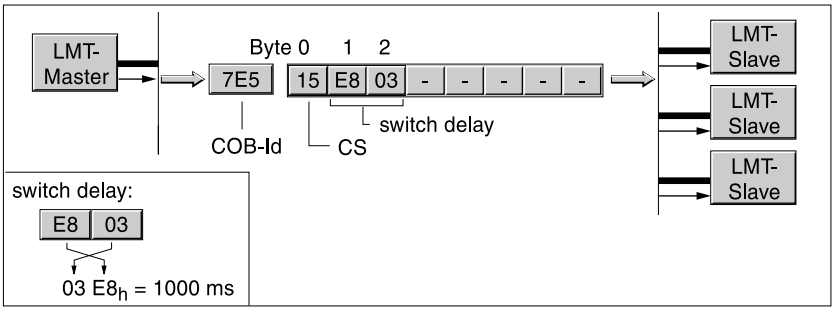

Fig. 3.39 LMT service “Activate bit timing parameters“ with „switch delay“

The delay is evaluated twice: during the first delay, every device continues to receive at the old baud rate. Once the delay has elapsed for a device, it must switch to the new baud rate. It may no longer transmit at the old rate. After the second delay has elapsed, all devices send and receive at the new rate.

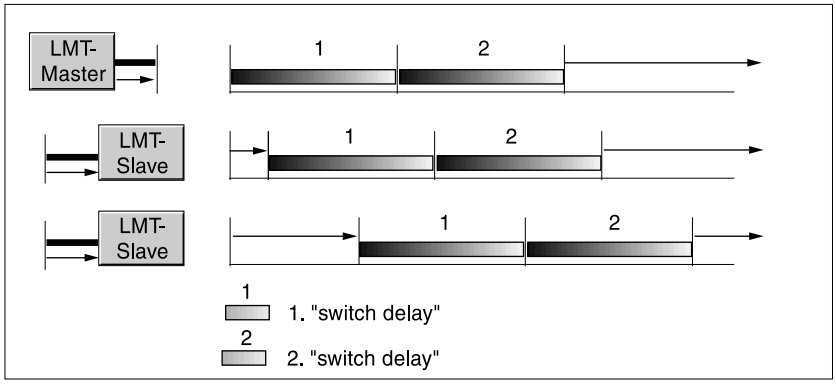

Fig. 3.40 Delaying the start of the „switch delay“ due to LMT slaves still transmitting

To avoid a situation where one device is already transmitting at the new transmission rate while another is still set to the old rate, the delay should be set to long enough to ensure that all devices can finish their current transmissions within a period of time, and that all LMT messages have been received.

5960 0100 011, -000, 04.01

4 Set-up

4.1 Set-up steps

Carry out set-up in three stages

- Configure connections
- Set reference parameters, movement parameters and sense of rotation
- Carry out test run.

Installation OK

The positioning device must be correctly installed. The steps for connecting and wiring up the positioning device in the CAN-Bus are described in the manual for the positioning device.

If the positioning drive has already been referenced, you can carry out a first manual test run without connecting up to the field bus. Details are given in the manual.

You can only check the reference status of the drive through the field bus. You will find an example of this in the section entitled „Checking referencing“ on page 5-21.

CAN-Bus master

Node address and transmission rate on the positioning device are preset for operation in the CAN-Bus. In order to change the settings, a field bus device with LMT master functionality must be built into the CAN-Bus.

It makes sense to use a CAN Monitor for developing programs in CANopen in order to be able to test, observe and evaluate the data being exchanged over the CAN-Bus.

5960 0100 011, -000, 04.01

4.2 Configuring connections

Node address and transmission rate are set via LMT services. The following pre-sets are stored in the positioning device :

- Node address 127
- Transmission rate 20kBaud.

Settings can be changed via an LMT master. There are two configuration choices available:

- positioning device is the sole device in the network: configuration in LMT mode „Switch mode global“
- positioning device is integrated while network operation is in progress: configuration in LMT mode „Switch mode selective“.

For the second option, the LMT address of the positioning device must be known or must be determined by SDO.

You will find information on LMT services in the Chapter entitled „Layer Management services“ from page 3-30.

4.2.1 Configuration in LMT-mode „Switch mode global“

▶ Switch off all CAN-Bus devices apart from the LMT master and the positioning device which you want to configure.

CANopen example „node address and baud rate“

The COB-IDs of the LMT services are:
- for the LMT master 2021 (7E5h)
- for the positioning device as LMT slave: 2020 (7E4h).

COB-ID	Data	Explanation
7E5	04 01 xx xx xx xx xx xx	change to configuration mode
7E5	11 01 xx xx xx xx xx xx	assign new node address 01h;
81		boot-up message
7E4	11 00 00 xx xx xx xx xx	response: message correctly executed
7E5	13 00 05 xx xx xx xx xx	set new baud rate 100 kBd
7E4	13 00 00 xx xx xx xx xx	response: message correctly executed
7E5	17 xx xx xx xx xx xx xx	save configuration
7E4	17 00 00 xx xx xx xx xx	response: message correctly executed
7E5	15 E8 03 xx xx xx xx xx	activate baud rate change, wait approx. 3 times “delay time”, then ..
7E5	04 00 xx xx xx xx xx xx	transition to operation mode

Baud rate settings correspond to the standard table, see page 3-34.

Before changing into operation mode, the new baud rate must also be switched on for the NMT master.

5960 0100 011, -000, 04.01

4.2.2 Configuration in LMT mode „Switch mode selective“

The LMT address of the positioning device is made up of three parts:

- Manufacturer's name „Positec” (50h, 6Fh, 73h, 69h, 74h, 65h, 63h)
- Product name „IclA065“ (49h, 63h, 6Ch, 41h, 30h, 36h, 35h)
- serial number

Every positioning device has its own serial number which can be determined by SDO through the manufacturer-specific object *identity object* (Index 1018h, Sidx 04h). You will also find the number on the type plate or barcode label on the device.

CANopen example „requesting serial number“

The example shows a request for the serial number of the network device with the address 127 (7Fh) by SDO through the *identity object* (Index 1018h, Sidx 04h).

The COB-ID for R_SDO is 600h+7Fh, and for T_SDO 580h+7Fh.

COB-ID	Data	Explanation
0	80 7F	NMT service Pre-operational, node address: 127 (7Fh)
67F	40 18 10 00 00 00 00	R_SDO: request for serial number
5FF	43 18 10 22 72 9E B1	T_SDO: serial number 2980002184

The positioning device can now be addressed by its manufacturer’s name, product name or serial number, in order to set transmission rate and node address:

CANopen example „node address“

The example shows how to set the node address in the network. The objects of the LMT services for the LMT master are: 2021 (7E5h) and for the positioning device as LMT slave: 2020 (7E4h).

COB-ID	Data	Explanation
7E5	01 50 6F 73 69 74 65 63	mode change; 1 = manufacturer’s name, Positec
7E5	02 49 63 6C 41 30 36 35	mode change; 2 = product name, IclA065
7E5	03 00 00 29 80 00 21 84	mode change; 3 = serial number, 2980002184
7E5	11 01 xx xx xx xx xx xx	assign new node address 01h
81		boot-up message
7E4	11 00 00 xx xx xx xx xx	response, message correctly executed
7E5	17 xx xx xx xx xx xx xx	save configuration
7E4	17 00 00 xx xx xx xx xx	response, message correctly executed
7E5	04 00 xx xx xx xx xx xx	transition to operation mode

5960 0100 011, -000, 04.01

4.3 Setting reference values and sense of rotation

The positioning device can only be operated in a referenced status. If the drive has not been referenced, the reference values must be set in homing mode.

If the drive has already been installed, the set sense of rotation must be taken into consideration for the test run.

Preparation for referencing

If the positioning device has been assigned an address, it can be enabled for network operation by means of the NMT service „Start remote node“. The operating modes can then be changed by means of PDOs. The example on page 5-25 in the Chapter entitled „Operate states and Operating modes“ shows all the steps required for preparing the drive for referencing.

Carrying out referencing

If the drive is not yet mechanically coupled to the system, you can carry out referencing in accordance with the example on page 5-25 in the Chapter entitled „Operate states and Operating modes“. Although the reference values in the example lead to valid referencing, they take no account of the set-up of your system.

You will find an example for calculating the right referencing values for your system set-up in the Chapter entitled „Referencing“ from page 5-19.

Setting the sense of rotation

The sense of rotation can be changed by means of two settings: via the sense of rotation factor and via position values.

You will find information on defining and setting the sense of rotation in the Chapter entitled „Direction reversal“ on page 6-5.

4.4 Test run

Manual movement

In order to carry out a test run in manual operation, the manual movement signals MAN_P and MAN_N of the signal interface must be connected.

▶ Activate the MAN_P or MAN_N signal. The drive rotates in a clockwise direction if the sense of rotation factor is positive, and anticlockwise if MAN_N is activated.

Positioning over the CAN-Bus

A test run over the CAN-Bus is carried out in the Positioning mode. The example on page 5-18 in the Chapter entitled „Operate states and Operating modes“ shows all the steps which must be implemented in order to carry out point-to-point positioning with a referenced drive.

5960 0100 011, -000, 04.01

5 Operate states and Operating modes

5.1 Overview: Operate state, mode transition and operating mode

The controls of a positioning drive are divided in two sections: controlling of the device with a master control unit (for example, a PLC) and operating function controlling by the device.

Operating states

After the positioning drive is switched on the drive toggles between several operate states for the execution of a drive job in a certain operating mode. The operate states are designed to allow controlled On/Off operation of the device and to enable a restart of operation following an error event. The operate states and mode transitions are described in the machine status description below the device profile DSP 402. Refer to figure Fig. 5.1.

Operating mode transitions

Operating mode transitions are carried out using three methods:

- via a command,
- automatically, for example, after the device is switched on,
- as a reaction to an error event.

Operating modes

The positioning drive can process run jobs in three operating modes:

- Manual mode,
- Positioning mode,
- Referencing.

All operating modes can be set and executed on the field bus.

Manual mode is set automatically by default after the power supply is switched on and if no error is pending. If no Field bus is connected the positioning drive is operated in manual mode only and controlled via the control inputs for manual mode.

Positioning operation can be executed with run profiles that are manufacturer specific and definable or defined according to the CANopen profile DSP 402.

Run jobs in all operating modes are processed in the operate state “Operation enable”.

5960 0100 011, -000, 04.01

5.2 Operating states and transitions

5.2.1 CANopen state machine

The relationship between the operate states and transitions of the positioning device are imaged under CANopen in the machine status of the DSP 402 for drives .

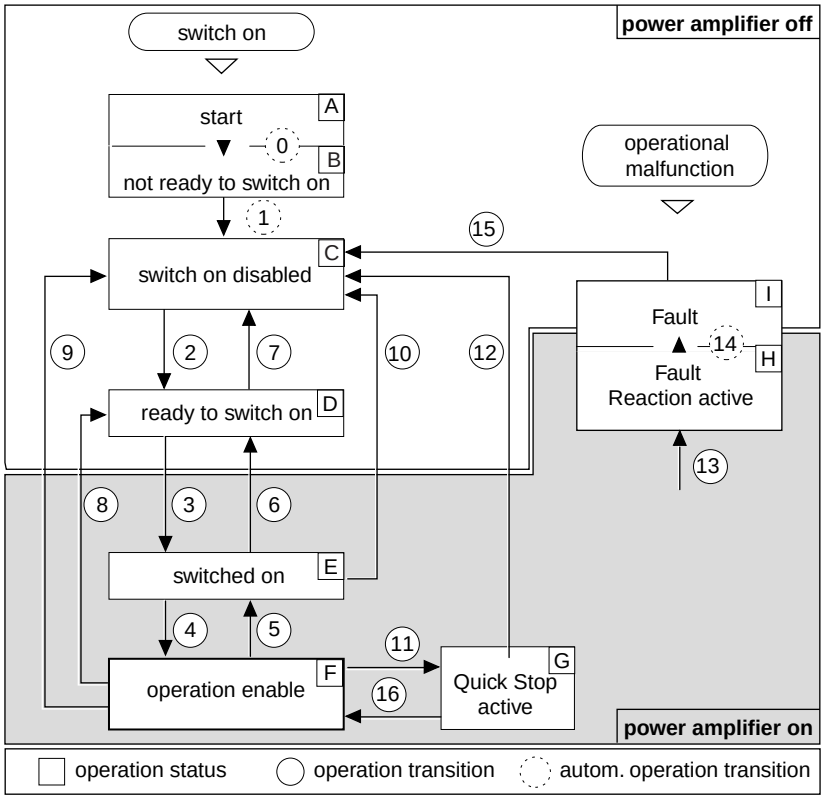

Fig. 5.1 Diagram of state machine.

Operating states The table shows the operate states of the machine status and their meaning.

Status	Operating status	Explanation
A	Start	24-V switched on, Initialization of internal electronics
B	Not ready to switch on	power amplifier is not ready to switch on
C	Switch on disabled	Switching on power amplifier is disabled
D	Ready to switch on	power amplifier ready to switch on
E	Switched on	power amplifier switched on, no active mode
F	Operation enable	The device is ready for operation.
G	Quick Stop active	Quick-Stop will be executed
H	Fault reaction active	Fault detected, fault reaction will be activated, if possible
I	Fault	

5960 0100 011, -000, 04.01

After switch-on, the positioning device remains in operating status 1 for approx. 4 seconds for initialization.

Variations in the positioning device from the state machine according to DSP 402

In order to keep the operating temperature of the compactly built positioning device as low as possible, its operating behavior varies in two regards from the state machine shown:

- The power amplifier is switched off as soon as the motor shaft has come to a halt. This affects the „Operation enable“, „Quick-Stop“ and „Fault reaction active“ states.
- The „Switched on“ status does not switch on the power amplifier. The final stage is automatically switched on when a run job is started and if the operate state is set to “Operation enable”. The final stage is switched off again after the run job has terminated .

These variations are not relevant for describing changes of status under CANopen. In the positioning device – as for any other device conforming to DSP 402 – all states and changes of status are permitted and necessary.

Transitions via command

Operate state transitions are triggered via command using the object *controlword* (6040h).

The table below illustrates the operate state transitions that can be triggered by command. The specified numbers of the transitions and the letters of the output and final states refer to the default profile, Fig. 5.1.

Transition Nr.	Transition from..to	Command	Response
(2)	C → D	Shutdown	Status transition
(3)	D → E	Switch On	Status transition; the intermediate circuit must be supplied with energy.
(4)	E → F	Enable Operation	The drive function is enabled; the device automatically switches on the final stage when a run job is pending.
(5)	F → E	Disable Operation	Ends the run job; the final stage is switched off; drive functions are disabled.
(6)	E → D	Shutdown	Status transition
(7)	D → C	Quick Stop	Status transition
(8)	F → D	Shutdown	Ends the run job; the final stage is switched off; drive functions are disabled.
(9)	F → C	Disable Voltage	Ends the run job; the final stage is switched off; drive functions are disabled.
(10)	E → C	Disable Voltage	Status transition
(11)	F → G	Quick Stop	Execute “Quick Stop”; stops the motor movement, if required; the drive stops; the final stage is switched off.
(12)	G → C	Disable Voltage	Status transition
(13)	→ H		Fault reaction after disrupted operation
(14)	H → I		Transition to “Error” state

5960 0100 011, -000, 04.01

Transition Nr.	Transition from..to	Command	Response
(15)	I → C	Fault Reset	Acknowledge the error in order to exit the “Error” state.
(16)	G → F	Enable Operation	Resume the run job from operate state “Quick Stop”

Automatic operate state transitions

The operate states 0 and 1 are automatically established when the device is switched on, while the result of an error event is a transition to operate state 14.

Fault response

Transition 13 triggers a fault response as soon as a monitoring signal reports a malfunction to which the unit must respond.

If status H “Fault reaction active” is enabled, intended actions taken on the device in case of non-fatal errors are, for example, interruption of the current positioning run, disabling the final stage, sending an error message. The device subsequently toggles to “Fault” state.

A malfunction can for example be reported by a temperature sensor. The unit interrupts the current movement command, executes a fault response, e. g. an emergency stop by means of Quick Stop or switching off the power amplifier, and then switches into the operating status „Fault“. Every malfunction is reported on the network by means of an EMCY message.

In order to exit the operate state “Fault”, the error cause must

- have been cleared,
- and bit 7 “Reset fault” of the object controlword must have been toggled from 0 to 1.

To reach “Operation enable” state after fault elimination, the device must pass the transitions 15, 2, 3 and 4.

5960 0100 011, -000, 04.01

5.2.2 Betriebszustände wechseln und überwachen

The operate states are set by the user in the CANopen object *controlword* (6040h) and monitored with the object *statusword* (6041h). The operate states are controlled and influenced by the device's internal monitoring and system functions such as temperature and current monitoring.

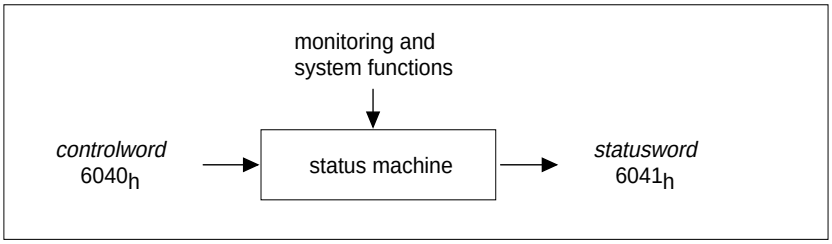

Fig. 5.2 Changing and monitoring operating status

controlword (6040h)

The 16 bit object controlwhord is used for several control jobs:

- Transition between the various operate states. For information on possible states and transitions please refer to the Chapter „Operating states and transitions“ as of page 5-2. Relevant for a status transition are the bits 0 to 3 and bit 7.
- Starting and interrupting specific operating mode functions, for example, starting a run job via bit 4. Bits 4 to 6 are used for specific operating mode settings. For details please refer to Chapter „Setting and monitoring operating modes“ as of page 5-9 a to the description of the respective operating modes in the Chapters 5.4 and 5.5.
- Stopping the positioning drive during a run operation. Bit 8 “Halt” is used to stop the drive. For details please refer to Chapter „Setting and monitoring operating modes“, starting on page 5-9.

The table below shows the designation of the controlword bits..

Bit	Significance	Identifier
0	LSB	Switch on
1		Disable voltage (low aktiv)
2		Quick stop (low aktiv)
3		Enable operation
4 - 6		Operation specific
7		Reset fault
8		Halt
9 - 10		reserved
11-15	MSB	Manufacturer specific, not used

5960 0100 011, -000, 04.01

statusword (6041h)

Use the 16-bit object statusword to carry out monitoring functions:

- Checking the operate state of the positioning controls. Bits 0 - 3, 5 and 6 are here relevant.
- Bit 4 indicates whether the final stage is ready to carry out a run job.
- Bits 7 to 15 are used to monitor run operation and the status of device specific states. For details on monitoring the run operation please refer to Chapter „Setting and monitoring operating modes“ , starting on page 5-9 and to the description of the respective operating modes in the Chapters 5.4 and 5.5. The bits for monitoring the device status are described in Chapter „Diagnostics and Error Correction“.

The table below shows the designation of the statusword bits.

Bit	Significance	Identifier
0	LSB	Ready to switch on
1		Switched on
2		Operation enabled
3		Fault
4		Voltage disabled
5		Quick stop
6		Switch on disabled
7		Warning
8		Manufacturer specific
9		Remote
10		Target reached
11		Internal limit active
12 - 13		Operation mode specific
14 - 15	MSB	Manufacturer specific

5960 0100 011, -000, 04.01

Operating states *changing*

Operating states are set via the control word by means of the *control-word* object (6040h).Bits 0 to 3 and bit 7 are relevant for a change of status.

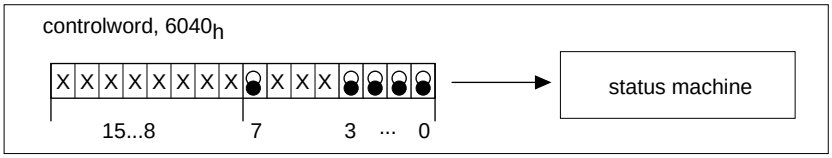

Fig. 5.3 Bits in the control word for changing the operating status

Obj. *controlword* (6040h) **State transition: Command**	**Change to status**	**Bit 7, Reset Fault**	**Bit 3, Enable operation**	**Bit 2, Quick Stop**	**Bit 1, Disable Voltage**	**Bit 0, Switch On**
(2), (6), (8): Shutdown	[D] Ready to switch on	X	X	1	1	**0**
(3) : Switch on	[E] Switched on	X	X	1	1	**1**
(7), (9), (10), (12) : Disable Voltage	[C] Switch on disabled	X	X	X	**0**	X
(7), (10) : Quick Stop	[C] Switch on disabled	X	X	**0**	1	X
(11) : Quick Stop	[G] Quick Stop active					
(5): Disable operation	[E] Switched on	X	**0**	1	1	1
(4), (16) : Enable operation	[F] Operation enable	X	**1**	1	1	1
(15) : Fault reset	[C] Switch on disabled	**0 -> 1**	X	X	X	X

☐ Status according to state machine

◯ State transition according to state machine

Bit states in the fields marked with an „X“ have no significance for the particular change of status. Bits 4 to 6 are used for settings specific to the operating mode. You will find details in the description of the individual operating modes further on in this chapter.

In order to achieve the fastest possible access to a change of status, the control word data are shown in the first two bytes of the PDOs, R_PDO1 and R_PDO2.

5960 0100 011, -000, 04.01

Operating states *monitoring*

Operating states are monitored via bits 0 to 3, 5 and 6 of the status word which can be read by means of the *statusword* object (6041h).

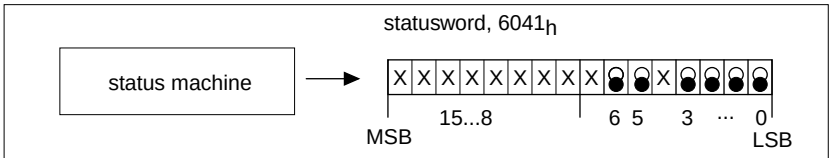

Fig. 5.4 Bits in the status word for recognizing the operating status

Obj. *statusword* (6041h) status	Bit 6, Switch on disable	Bit 5, Quick Stop	Bit 3, Fault	Bit 2, Operation enable	Bit 1, Switch on	Bit 0, Ready to switch on
B Not ready to switch on	0	X	0	0	0	0
C Switch on disabled	**1**	X	0	0	0	0
D Ready to switch on	0	1	0	0	0	**1**
E Switched on	0	1	0	0	**1**	1
F Operation enable	0	1	0	**1**	1	1
G Quick Stop active	0	**0**	0	1	1	1
I Fault	0	X	**1**	1	1	1

☐ Status according to state machine

Bit 4, voltage disable

Bit 4 displays whether the power amplifier is ready to process a movement command.

Bit 7-15

Bits 7 to 15 are used for monitoring movement mode and the status of certain conditions specific to the unit. You will find details on monitoring movement mode in the description of the individual operating modes. The bits for monitoring the status of the unit are described in the Chapter on error handling.

In order to achieve the fastest possible access to the operating status, the status word data are shown in the first two bytes of the PDOs, T_PDO1 and T_PDO2.

CANopen example „switching operating status“

The CANopen program example shows the switching of operating states for a network device with the node address 1.

COB-ID	Data	Explanation
0	01 00	NMT change of status for all network devices to ”Operational”
181	xx xx	if required, reporting the actual state (any)
201	00 00	R_PDO1: request status change to Switch on disabled
181	40 00	T_PDO1: change of data in the status word: Switch On Disabled
201	06 00	R_PDO1: request status change to Ready to Switch On
181	21 00	T_PDO1: change of data in the status word; Ready to Switch On
201	07 00	R_PDO1: request status change to Ready to Switch On -> Switched On
181	23 00	T_PDO1: change of data in the status word: Switched On
201	0F 00	R_PDO1: request status change to Switched On -> Operation Enable
181	27 00	T_PDO1: change of data in the status word: Operation Enable

5960 0100 011, -000, 04.01

5.3 Setting and monitoring operating modes

Overview There are three objects available in CANopen for changing between operating modes and displaying both possible modes and the current mode.

5.3.1 Checking and changing operating mode settings

List of possible operating modes By using the *supported drive modes* object (6502h), you can obtain information on all the operating modes that can be set on the positioning device.

Mode	Description from ...
Manual mode, manufacturer-specific	page 5-12
Positioning mode, standardized for CANopen Positioning mode, manufacturer-specific	page 5-14
Referencing, standardized for CANopen	page 5-19

Configuration mode The configuration mode is also displayed via the *supported drive modes* object (6502h). It can be set as an alternative to an operating mode, but it cannot be carried out at the same time as one. Configuration mode is used to set the drive's system parameters and application parameters.

Reading off the current mode The current mode can be read off by SDO via the *modes of operation display* object (6061h).

Changing modes The operating mode can be changed by SDOs using the *modes of operation* object (6060h).

5960 0100 011, -000, 04.01

5.3.2 Starting and interrupting the operating mode

The Positioning and Referencing operating modes are initiated over the field bus, and manual mode via the manual movement signals on the signal interface. Movement data and parameters are exchanged over the field bus interface.

Starting conditions

In order to be able to carry out the set mode, the following starting conditions must be met:

- unit not in fault condition
- valid setpoint values in manual and positioning, refer to Chapter „Referencing“ starting on page 5-19.

Starting the operating mode

Bits 4 to 6 are occupied by specific operating mode functions. The drive receives a request via bit 4 in the *controlword* object (6040h) to start the operating mode which has been set.

Operating mode	**Obj. *controlword* (6040h)**	**Explanation**
Positioning mode	Bit 4: *New setpoint*	0-> 1: Start positioning
	Bit 5: -	not assigned
	Bit 6: *absolute / relative*	0: absolute positioning 1: relative positioning
Referencing	Bit 4: *Homing operation start*	1: start referencing
	Bit 5, 6: -	not assigned

Interrupt operating mode

A running movement can be interrupted by means of bit 8 of the control word *controlword* (6040h).

Obj. *controlword* (6040h)	**Explanation**
Bit 8: *halt*	0: carry out processing 0->1: interrupt processing

5960 0100 011, -000, 04.01

5.3.3 Monitoring the operating mode

The drive reports status information on the operating mode set via bit 8 and bits 10 to 15 in the *statusword* object (6041h).

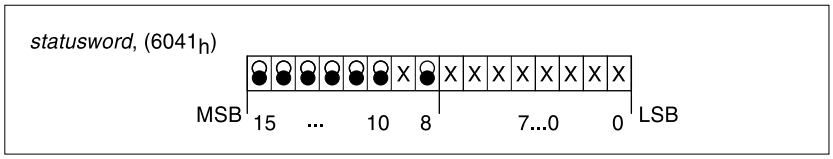

Fig. 5.5 Status reports on the operating mode

Target position and limit switches

Obj. *statusword* (6041h)	Explanation
Bit 8: work area left	only valid if bit 11 = 1: 0: limit switch position W_1 exceeded 1: limit switch position W_0 exceeded
Bit 10: *target reached*	1->0: new target position transmitted 0->1: requested target position reached
Bit 11: *internal limit active*	0->1: limit switch position W_0 or W_1 exceeded
Bit 14: work area left	0->1: limit switch position D_0 or D_1 exceeded
Bit 15: safety area left	0->1: limit switch position S_0 or S_1 exceeded

The limit switch positions are explained in the section entitled „Referencing“ from page 5-19 .

Status on the operating mode

Bits 12 and 13 are assigned differently depending on the operating mode.

Operating mode	Obj. 6041h, *statusword*	Explanation
Positioning mode	Bit 12: *setpoint acknowledge*	0: new position can be accepted 1: new target position received
	Bit 13: -	not assigned
Referencing	Bit 12: *homing attained*	0: referencing not yet carried out 1: referencing carried out
	Bit 13: *homing error*	0: no error 1: error when referencing

5960 0100 011, -000, 04.01

5.4 Manual mode

Overview

In manual mode the positioning device can be made to turn clockwise or anti-clockwise by means of the manual movement signals MAN_P and MAN_N at the signal interface.

A manual movement is only carried out within the reference work area. Connection to the field bus is not necessary for manual mode.

Function

Depending on the length of the manual movement signal, the positioning device either works in inching mode or continuous mode.

If a manual movement signal is activated briefly, the drive executes an adjustable number of increments in inching mode. If the signal is of longer duration, the drive switches into continuous mode. The time until changing from inching to continuous mode is adjustable (release time).

After manual run has been switched on and before it is started, the positioning device is delayed by a 50 ms debounce time.

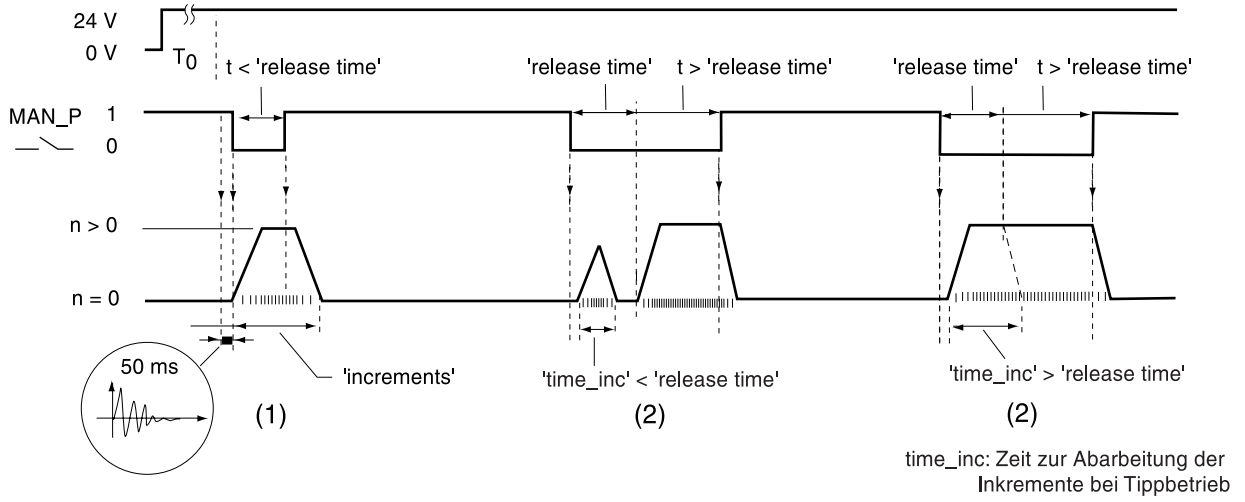

Fig. 5.6 Inching mode (1) and Continuous mode (2), T_0 : Drive start time after switch-on, approx. 6 seconds

Settings

The parameters in manual mode are set via the *manual mode settings* object (2011h). The following can be set:

object 2011h, sub-index	Explanation
01h: increments	number of increments in inching mode
02h: velocity	final velocity
03h: acceleration	acceleration ramp
04h: deceleration	deceleration ramp
05h: max current	max. current, operating current limitation
06h: release time	inching duration

The drive accelerates in accordance with the adjustable ramp value, „acceleration“, but no higher than the manual movement speed, „velocity“. If the signal is released, the drive decelerates in line with the „deceleration“ braking ramp down to a speed of zero. The maximum torque in manual mode can be reduced by the setting on the „max current“ operating current limitation.

5960 0100 011, -000, 04.01

Final velocity, acceleration and deceleration ramps are limited by the maximum values *Max profile velocity* (607Fh), *Max acceleration* (60C5h) and *max deceleration* (60C6h).

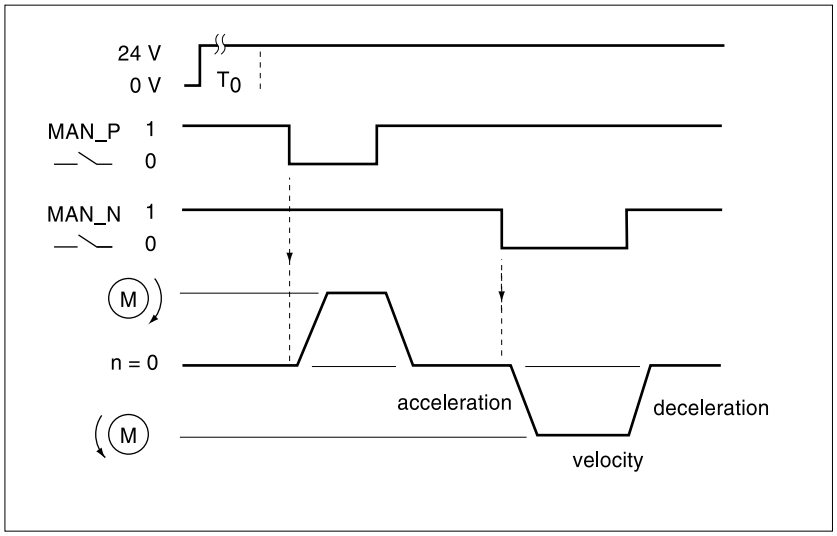

Fig. 5.7 Adjustable values for manual mode

The sense of rotation is determined by the control inputs for manual mode. It is only possible to reverse the sense of rotation when the motor is at a standstill.

5960 0100 011, -000, 04.01

5.5 Positioning mode

Function

In positioning mode, the drive is moved from a starting position to a target position by means of an adjustable movement profile. The value for the target position can be specified as a relative or absolute position, and it must lie within the referenced work area.

Positioning mode can be executed in two operating mode settings:

- Positioning mode with standardized objects conforming to the device profile DSP 402 (profile position mode).

  A movement profile can be set with values for the acceleration and deceleration ramps as well as for the final velocity. The values do not remain stored.

- Positioning mode with manufacturer-specific objects (manufacturer-specific profile)

  A choice can be made between ten pre-defined movement profiles for a positioning process. Nine of the ten movement profiles are adjustable, and one is fixed. The values can be stored.

In both operating modes positioning is carried out via linear acceleration and deceleration ramps.

Relative and absolute positioning

With relative positioning a new target point is measured from the current starting position of the drive, and with absolute positioning from the machine zero point.

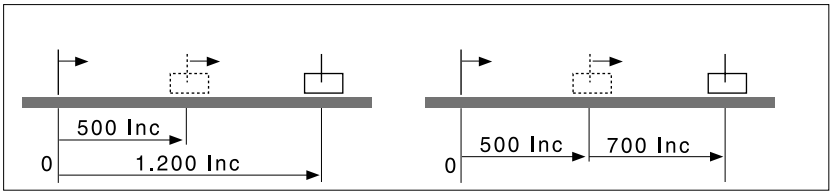

Fig. 5.8 Absolute positioning (left) and relative positioning

You can select the positioning mode – relative or absolute – every time you start a positioning run.

The type of positioning is set on bit 6 in the *controlword* object (6040h). The new target position is shown in the R_PDO2 in bytes 2...5.

The drive stores all position values internally with reference to the machine zero point.

Current Position

The current position of the drive is requested by means of the *position actual value* object (6064h). An absolute value is returned in motor increments.

The current position is transmitted by the drive as a 4 byte value together with the *statusword* object (6041h) in the second T_PDO.

5960 0100 011, -000, 04.01

5.5.1 Starting positioning

Requirement

Positioning mode must be set, and the motor must be at rest. The default settings can be used for the movement profile, but an individual profile can also be transmitted to the drive and activated, see „Settings and Objects“ from page 5-16.

Signals

Object / Signal	Explanation
controlword (6040h)	control word
Bit 6: *new setpoint*	0-> 1: start positioning
Bit 8: *halt*	0: carry out processing 0->1: interrupt processing
Bit 10: *target reached*	1->0: new target position transmitted 0->1: requested target position reached
statusword (6041h)	status word
Bit 12: setpoint acknowledge	0: new position can be accepted 1: new target position received

Before positioning, the new target position is transmitted by SDO by means of the *target position* object (607Ah) or by R_PDO2, bytes 2..5. The motor does not move yet.

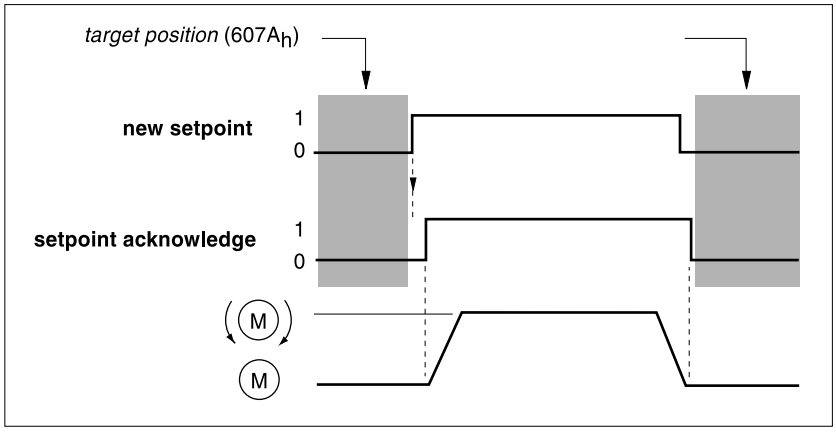

Fig. 5.9 Data transmission for the start of a positioning operation

The process is initiated via bit 4 „new setpoint“ in the *controlword* object (6040h). The transition from ”0” to “1” signals to the drive the transmission of a position and that a run job can be started. The drive confirms via bit 12 „setpoint acknowledge“ = 1, that the movement command has been activated. Bit 4 can be reset. If a movement command cannot be carried out, the drive issues an error message.

The target position is reached as soon as bit 10 in the status word, *target reached*, changes to „1". The drive automatically reports the change of status via T_PDO1.New data can now be processed.

Interrupting positioning

A running movement can be interrupted by bit 8 of the control word *controlword* (6040h).

5960 0100 011, -000, 04.01

5.5.2 Settings and Objects

Positioning mode can be set and executed via the standardized objects of the CANopen „Profile position mode“ in line with DSP 402. In addition manufacturer-specific objects are also available, thereby allowing the use of further freely definable movement profiles for controlling the drive.

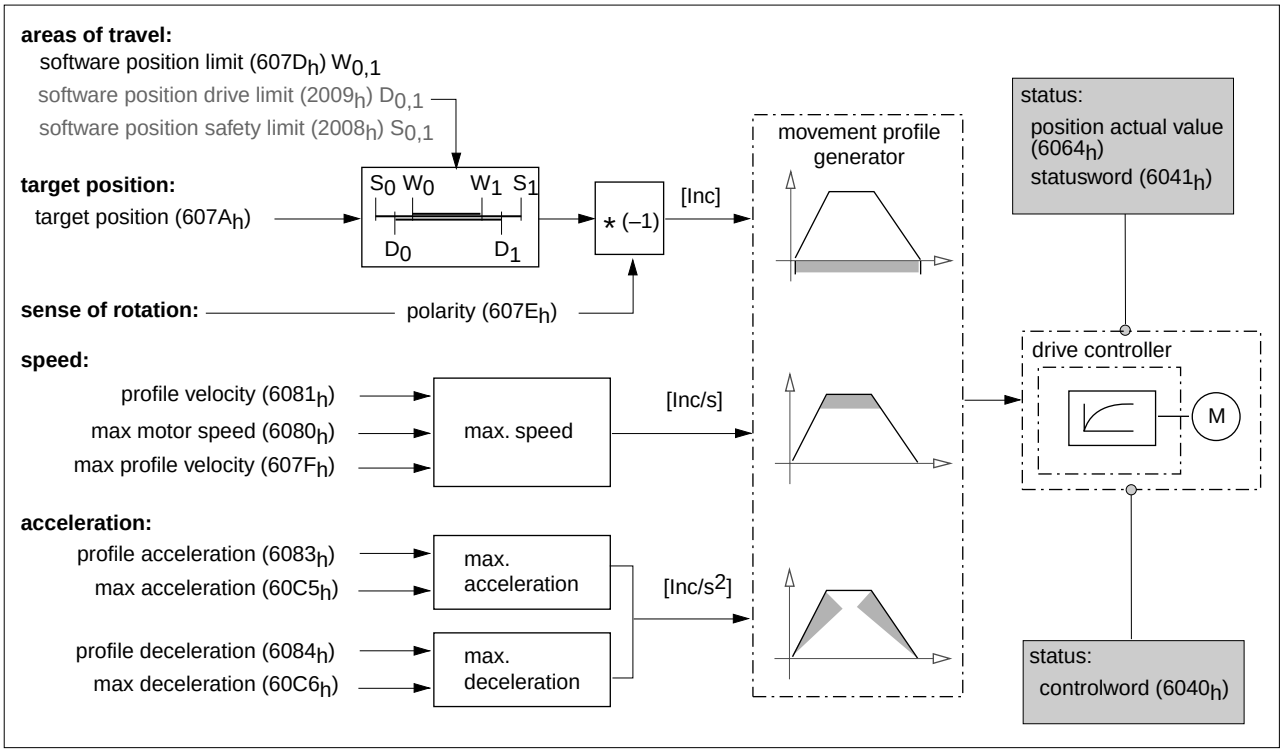

Fig. 5.10 „Profile position mode“ positioning mode and settings

Target position A new position value is transmitted by means of the *target position* object (607Ah) or with the R_PDO2, bytes 2..5. The position is given in motor increments (Inc) and must lie within the defined limits of the work area. If the target position lies outside the limits, the drive issues an error message.

The new target position results from the position value, multiplied by the sense of rotation factor. You will find details on the sense of rotation factor in the Chapter entitled „Direction reversal“ on page 6-5.

Movement range The movement range is limited by software limit switches. In a referencing process, the limit switch values are stored in the objects *software position limit* (607Dh), *software position drive limit* (2009h) and *software position safety limit* (2008h).

Movement profile For the positioning process, the drive responds in accordance with a movement profile, defined by acceleration and deceleration ramps as well as final velocity.

5960 0100 011, -000, 04.01

Movement profile with standardized objects

The standardized positioning mode must be set by means of the *modes of operation* object (6060h). The following objects are then available for defining the movement profile in line with device profile DSP 402.

object	Explanation
profile velocity (6081h)	final speed
profile acceleration (6083h)	acceleration ramp
profile deceleration (6084h)	deceleration ramp

The run profile parameters cannot be stored retentive.

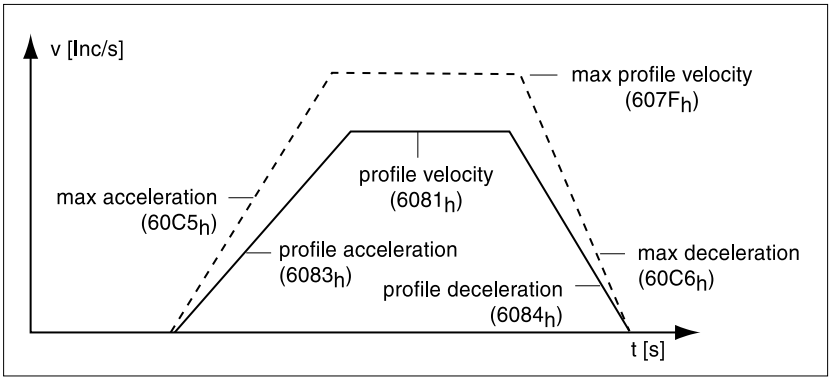

Fig. 5.11 Run profile parameters

Profile settings apply to both directions of movement for the drive.

Maximum values limit the settings which can be made for a movement profile. They are store in the drive's data sheet and cannot be changed. The drive issues an error message if a ramp setting exceeds a maximum limit.

Object	Explanation
max profile velocity (607Fh)	maximum final speed
max motor speed (6080h)	maximum speed of motor
max acceleration (60C5h)	maximum acceleration ramp
max deceleration (60C6h)	maximum deceleration ramp

5960 0100 011, -000, 04.01

Movement profile with manufacturer-specific objects

In order to use the ten pre-configured movement profiles for a positioning process, manufacturer-specific positioning mode must first be set.

The first of the ten movement profiles is factory-set and cannot be changed. It sets the profile speed for the drive's standard rated speed.

For the other nine movement profiles, the acceleration and deceleration ramps as well as the final velocity can be freely set and permanently stored in configuration mode. Each object for setting the parameters contains ten fields for run profile data.

Object	Explanation
user profile velocity (2004h)	final velocity
user acceleration (2005h)	acceleration ramp
user deceleration (2006h)	deceleration ramp
user profile number (2007h)	selection of a movement profile

CANopen example „Positioning with SDO, PDO and SYNC“

The example shows positioning by SDO, change of operating status by PDO1 and start of operation by PDO2 with SYNC reception.

COB-ID	Data	Explanation
601	23 83 60 00 D0 07 00 00	T_SDO: set acceleration ramp: 2000 Inc/s/s
581	60 83 60 00 00 00 00 00	R_SDO: OK
601	23 81 60 00 F4 01 00 00	T_SDO: set speed: 500 Inc/s
581	60 81 60 00 00 00 00 00	R_SDO: OK
601	23 84 60 00 E8 03 00 00	T_SDO: set deceleration ramp: 1000 Inc/s/s
581	60 84 60 00 00 00 00 00	R_SDO: OK
0	01 00	NMT Protocol; Start Remote Node
181	xx xx	reporting the actual state (any)
201	06 00	R_PDO1: Request status change to Ready to switch on
181	21 00	T_PDO1: Data modification in statusword; ready to switch on
201	07 00	R_PDO1: Request status change to switched on
181	23 00	T_PDO1: Data modification in statusword, switched on
201	0F 00	R_PDO1: Request status change to Operation enable
181	27 00	T_PDO1: data modification in statusword; Operation enable
601	2F 60 60 00 01	T_SDO: switch to Profile Position Mode
581	60 60 60 00 00	R_SDO: OK
301	1F 00 DC 05 00 00	R_PDO2: bytes 0,1: control word, bytes 2..5: targ.pos. 5DCh = 1500 Inc
80		SYNC: SYNC object for positioning start
181	27 10	T_PDO1: setpoint acknowledge
201	0F 00	R_PDO1: Bit 4: reset new setpoint
181	27 00	T_PDO1: setpoint acknowledge = 0
181	27 14	T_PDO1: Bit 10, target reached = 1

5960 0100 011, -000, 04.01

5.6 Referencing

Function

Referencing defines the permitted area of travel for the positioning device. The boundaries to the area are stored as absolute values in the positioning device. The positioning device can only be operated when it has been referenced.

Referencing values remain in place after the positioning drive has been switched off and on, as the current position value of the drive is stored as an absolute value in its internal location memory.

If the positioning device detects that the motor shaft has been moved while the power was off, referencing values are cancelled and the process must be carried out again.

In order to carry out referencing, the drive must be located inside the permitted work area to allow it to be moved after referencing has taken place.

5.6.1 Position capture

Overview

For monitoring, processing and storing repeatable position values, the drive continuously updates two pieces of information.

- The rotor position of the motor
- The position counter

The motor works internally with a resolution of twelve increments which are detected with Hall sensors. The actual position is recorded at the same time as an absolute value by means of a 32 bit position counter.

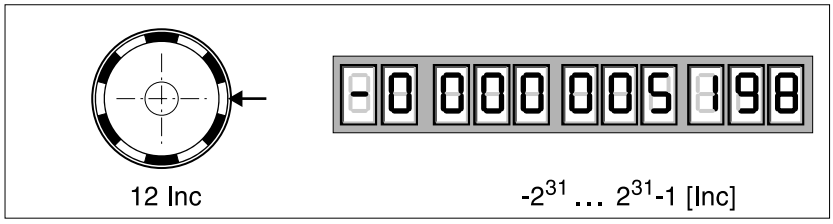

Fig. 5.12 Position capture

Saving position data

When the power supply is switched off, the position counter and rotor position are stored in the unit's memory.If the motor is still turning, it is first slowed down by means of the stator's permanent magnetic field.

Monitoring data storage

An internal safety check detects whether the position data have been correctly saved. If it was not possible, for example, to stop the motor in time due to an external load, the monitoring function causes the reference values to be reset when the power is switched on again.

When the power is switched on again, the drive compares the current position of the rotor with the value in its memory. If it detects that the motor has been moved by more than one increment – with reference to a half turn of the motor – the reference values are deleted. The drive must be referenced again. Any deviation of around ±1 increment is updated in the location memory without deleting the referencing values.

5960 0100 011, -000, 04.01

Quasi absolute value detection

The positioning device uses the position of the rotor to evaluate whether referencing values are valid. The resolution of twelve increments results from the following:

3 Hall sensors ($H_{1,2,3}$) x 2 magnetization states (N, S) x 2 pairs of poles (NS, NS).

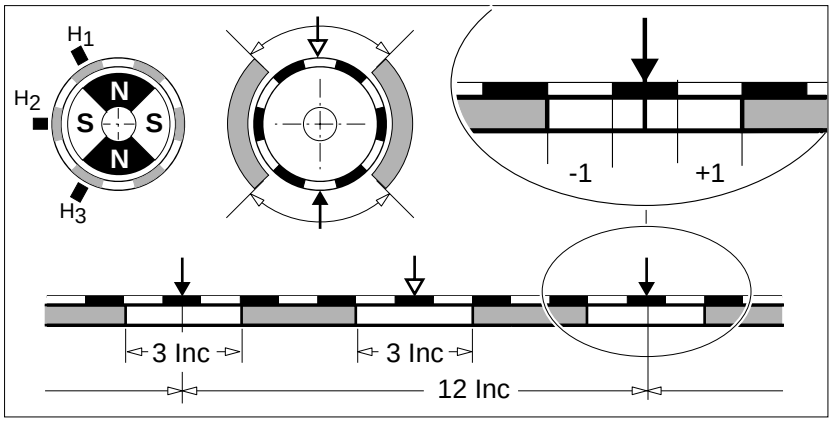

Fig. 5.13 Detection of mechanical displacement of motor (grey)

Detection of motor displacement while power supply is switched off occurs in 50% of cases. A mechanical change in position of six or twelve increments per revolution plus the tolerance range of ±1 increment is not recognized by the quasi absolute value detection system as a mechanical displacement.

5.6.2 Referencing areas

Valid referencing is defined via three limit switch zones which must lie within the possible area of travel of the drive. The limit switches protect the drive and the system from damage.

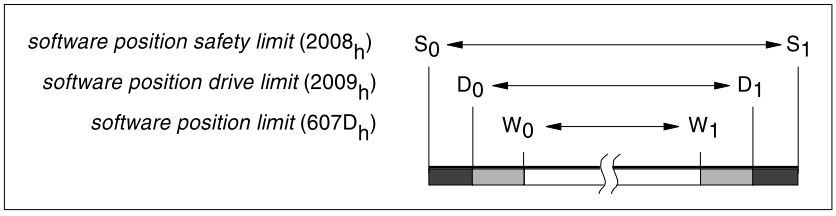

Fig. 5.14 Areas of travel of the drive

- Work area W_0 - W_1 for Positioning mode
- Drive range D0 - D1. From the D0 - W0 and D1 - W1 ranges, the drive can only be returned towards working range.
- Safety area S0 - S1. From the S0 - D0 and S1 - D1 ranges, the drive can only be returned manually.

CANopen objects

Three CANopen objects are used to set up the limit switches, and the position values for the upper and lower limits to the area are entered in them.

- Work area limits in *software position limit* (607Dh)
- Travel area limits in *software position drive limit* (2009h)
- Safety area limits in *software position safety limit* (2008h)

5960 0100 011, -000, 04.01

5.6.3 Referencing methods

The positioning device is referenced by taking current position as zero point.

The method is defined by the *Homing method* object (6098h). If a new position value is assigned, the value must be transferred to the motor control position counter by means of the *Position Assignment value* object (200Bh).

The sequence for carrying out referencing is

- Select Referencing mode, *Modes of operation* object (6060h)
- Set limit switch values and enter them
- The rotary direction can be set via Object 607Eh.
- Specify referencing method: Dimension setting on current position
- Start referencing with control word *controlword*, bit 4=„1“.

The DSP 402 device profile specifies use of the *Homing speed* object (6099h) with which the speed of a reference movement is determined. However it has no relevance for the positioning device, as the drive does not carry out any reference movements.

5.6.4 Checking referencing

The *manufacturer status register* object (1002h) can be used to check on the CAN-Bus whether the drive has already been referenced.

Obj. *manufacturer status register* (1002h),	**Explanation**
Bit 24	0: drive not referenced 1: drive referenced

example of „checking referencing“

The example shows how the referencing status can be checked by SDO. The node address of thepositioning device is set to 01h.

COB-ID	**Data**	**Explanation**
601	40 02 10 00 xx xx xx xx	R_SDO: read manufacturer status register
581	43 02 10 00 00 00 00 00	T_SDO: Referencing OK? Bit 24=0 =>not referenced

5960 0100 011, -000, 04.01

5.6.5 Setting software limit switches

Limit switch positions are given as position values in terms of the resolution of the motor. The reference points of the application are generally available in units of length, and have to be converted to internal motor increments.

Conversion factors

The following data are required from the positioning device for converting the application positions into internal increments:

- gear reduction ratio
- the number of motor increments per revolution

Gear reduction ratio

Gear versions are coded via the type code. The ratio is fixed for each type of gearbox, and can be determined via the *gear ratio* object (6091h). The reduction ratio is given as the ratio of motor revolutions to drive shaft revolutions; the drive shaft is the shaft which leads out of the gearbox.

Motor increments per revolution

The motor turns twelve increments per revolution. Multiplied by the gear reduction ratio, this gives the increments per revolution of the drive shaft. The ratio can be read from the motor data sheet via the *position encoder resolution* object (608Fh).

object **gearbox version**	**6091h** **reduction ratio** **[rev_M / rev_S]**	**608Fh** **motor inc/rev** **[inc_M / rev_S]**
IclA D065 DC024 S115	3675 : 32	44100 : 32
IclA D065 DC024 S054	490 : 9	5880 : 9
IclA D065 DC024 S038	75 : 2	900 : 2
IclA D065 DC024 S018	160 : 9	1920:9

5960 0100 011, -000, 04.01

Application example

The conversion of application data into increments is clarified by the following example. The drive is moving a slide via a spindle drive. In the process, the rotation of the spindle is converted into lateral motion. To prevent the slide from going beyond the permitted area of travel, the limit switch area boundaries have to be defined.

A positioning device *IclA* D065 DC024 S038 is used.

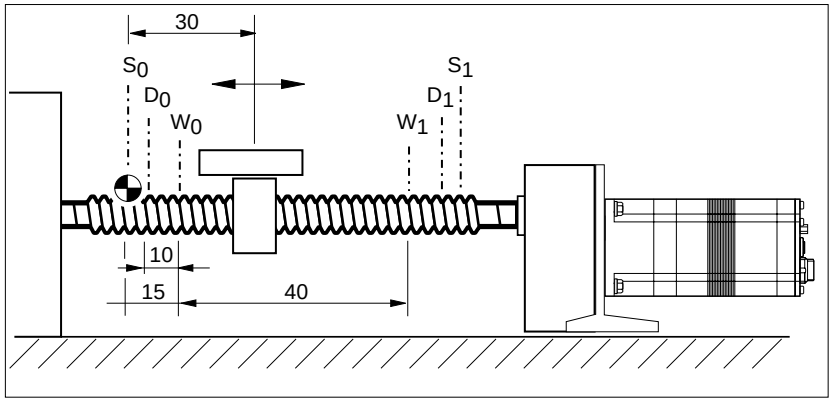

Fig. 5.15 Positioning with spindle drive

Application dimensions

- Work area W_0 - W_1: 40 mm
- Distance of work area W_0 - W_1
  - to the area of travel D_0, D_1 every 10 mm
  - to the safety area S_0, S_1 every 15 mm
- Position of reference pointPosition S_0
- Distance of current position from reference point 30 mm
- Spindle pitch: 1 mm/rev
- Gear reduction ratio: 75: 2
- increments per revolution of the motor: 12 Inc

Step 1 Conversion of rotational movement into distance increments

- 12 Inc/motor revolution * 75/2 = 450 Inc / spindle revolution
- 450 Inc / spindle turn * 1 spindle revolution = 450 Inc / mm

5960 0100 011, -000, 04.01

Step 2 Distance of neighboring limit switches

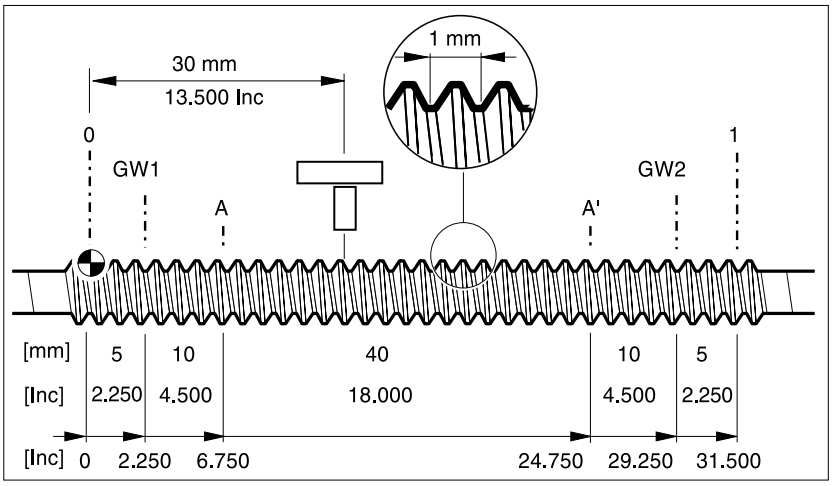

Fig. 5.16 Example calculation

- Distance S_0 - D_0 = 5 mm => 5 * 450 Inc = 2,250 Inc
- Distance D_0 - W_0 = 10 mm => 10 * 450 Inc = 4,500 Inc
- Distance W_0 - W_1 = 40 mm => 40 * 450 Inc = 18,000 Inc
- Distance W_1 - D_1 = 10 mm => 10 * 450 Inc = 4,500 Inc
- Distance D_1 - S_1 = 5 mm => 5 * 450 Inc = 2,250 Inc

Reference point relative to the current position

- Distance S_0 from current position: 30 * 450 Inc = 13,500 Inc

Step 3 Calculate limit switch values on reference point S_0

- Safety area S_0 : 0 Inc
- Area of travel D_0 : 2,250 Inc
- Work area W_0 : 2,250 + 4,500 Inc = 6,750 Inc
- Work area W_1 : 7,750 + 18,000 Inc = 24,750 Inc
- Area of travel D_1 : 25,750 + 4,500 Inc = 29,250 Inc
- Safety area S_1 : 30,250 + 2,250 Inc = 31,500 Inc

Step 4 Input of position values

Object (Index)	Position	Sub-index	Value [Inc], dec (hex)	Software limit switches
software position limit (607Dh)	min.	01h	6,750 (1A5Eh)	Work area W0
	max.	02h	24,750 (60AEh)	Work area W1
software position drive limit (2009h)	min.	01h	2,250 (8CAh)	Area of travel D0
	max.	02h	29,250 (7242h)	Area of travel D1
software position safety limit (2008h)	min.	01h	0	Safety area S0
	max.	02h	31,500 (7B0Ch)	Safety area S1

5960 0100 011, -000, 04.01

CANopen example „Referencing“

The following listing shows the input of referencing values. The node address of the positioning device is set to 01h.

COB-ID	Data	Explanation
601	2F 60 60 00 06	R_SDO: switch into Homing Mode
581	60 60 60 00 xx	T_SDO: OK
601	23 08 20 02 0C 7B 00 00	R_SDO: max. value for safety area S_1: 7B0Ch
581	60 08 20 02 xx xx xx xx	T_SDO: OK
601	23 08 20 01 00 00 00 00	R_SDO: min. value for safety area S_0: 0000h
581	60 08 20 01 xx xx xx xx	T_SDO: OK
601	23 09 20 02 42 72 00 00	R_SDO: max. value for area of travel D_1: 7242h
581	60 09 20 02 xx xx xx xx	T_SDO: OK
601	23 09 20 01 CA 08 00 00	R_SDO: min. value for area of travel D_0: 8CAh
581	60 09 20 01 xx xx xx xx	T_SDO: OK
601	23 7D 60 02 AE 60 00 00	R_SDO: max. value for work area W_1: 60AEh
581	60 7D 60 02 xx xx xx xx	T_SDO: OK
601	23 7D 60 01 5E 1A 00 00	R_SDO: min. value for work area W_0: 1A5Eh
581	60 7D 60 01 xx xx xx xx	T_SDO: OK
601	23 10 10 03 73 61 76 65	R_SDO: save application parameters: ”save”
581	60 10 10 03 xx xx xx xx	T_SDO: OK
601	2F 98 60 00 FF	R_SDO: selection of referencing type
581	60 98 60 00 xx	T_SDO: OK
601	23 0B 20 00 BC 34 00 00	R_SDO: dimension setting, actual position to S0: 34BCh
581	60 0B 20 00 xx xx xx xx	T_SDO: OK
601	2B 40 60 00 1F 00	R_SDO: Homing Operation Start (rising edge, bit 4)
581	60 40 60 00 xx xx	T_SDO: OK

5960 0100 011, -000, 04.01

5960 0100 011, -000, 04.01

6 Operating Functions

6.1 Safety functions

Emergency stop

The positioning device activates the built-in emergency stop safety function via two signal channels which work in different ways and as standbys to each other:

- through the emergency shutdown terminal, pin 2 of the signal interface
- through a field bus command.

You will find details on the connection of the external emergency shutdown signals described in the device manual. The *Manufacturer status register* object (1002h) signals whether the external emergency shutdown signal has been activated by means of bit 28=1.

Quick Stop

The emergency stop command over the field bus is triggered by a change to the operating status Quick Stop. Quick Stop is activated via bit 1 and bit 2 in the *controlword* object (6040h).

The drive decelerates in line with the adjustable Quick Stop ramp.

Object	Explanation
quick stop deceleration (6085h)	Quick Stop braking ramp

The *Manufacturer status register* object (1002h) reports that the drive has come to a stop by means of bit 26=0.

6.2 Monitoring devices

Monitoring and diagnostic functions ensure that operation of the drive is safe and reliable. Operational faults are reported by the drive via the EMCY object.

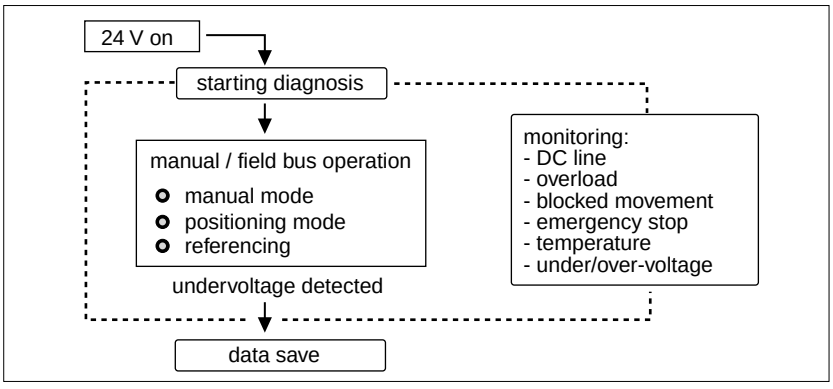

Fig. 6.1 Monitoring functions

Start-up check

After power supply has been switched on, the drive carries out a start-up check in which the important system states are tested. These include memory status, current, voltage, temperature of power amplifier and cable break in the emergency shutdown circuit. The check lasts approx. five to six seconds.

Monitoring during operation

During operation, temperature, current, voltage, speed and commutation status are continuously monitored in the positioning device .

5960 0100 011, -000, 04.01

Data security

If the power supply fails, the positioning device saves the current operating time data and a log-book to the configuration memory.

Device monitoring signals are evaluated through the *Manufacturer status register* object (1002h). You will find details of this in the Chapter entitled „Diagnostics and Error Correction“ from page 7-2.

6.2.1 Temperature monitoring

The temperature of the power amplifier is monitored during the entire operating time. In the event of excess temperature, the motor is stopped, the motor controller disabled and an EMCY message transmitted.

Excess temperature in the power amplifier is displayed in the *Manufacturer status register* object (1002h) through bit 2=1.

The current temperature of the power amplifier can be interrogated through the *Temperature actual value* object (200Dh).

6.2.2 Motor start-up

If the positioning device is not turning after a start-up time of approx. 1 second, the power amplifier switches itself off. The system remains ready for operation.

The drive transmits an EMCY message. The start-up fault is displayed via the *Manufacturer status register* object (1002h), bit 17.

6.2.3 Speed monitoring

After an acceleration phase at the start of a positioning process, the drive switches to steady travel and moves at its actual speed. If actual speed and set speed deviate by more than a threshold value, the drive transmits an EMCY message. Speed deviation can be detected through the *Manufacturer status register* object (1002h), bit 18.

The threshold value for speed deviation can be interrogated and set by means of the *Control parameter set* object (2010h), sub-index 06h.

6.2.4 Voltage monitoring

DC line voltage

During operation, the DC line voltage is continuously monitored. In the event of under or over-voltage, the motor is stopped, the motor controller disabled and an EMCY message transmitted. The *Manufacturer status register* object (1002h) shows an overvoltage fault in the DC line on bit 4, and undervoltage on bit 5.

An undervoltage condition prompts the operating time data to be saved. An overvoltage fault can only be reset by switching the power supply off and on again.

5960 0100 011, -000, 04.01

6.2.5 Current monitoring

The control electronic system uses a current monitor circuit to monitor current. This current monitor is equipped with a motor current detection, current limiting and overload detection circuit.
Parameters and status objects are available for the configuration of current monitoring.

Parameters for current monitoring

The parameters are declared in CANopen objects and can be read out from the object.

Symbol	Parameter	CANopen Object	Unit	Access
$I_{maxoperate}$	Maximum permitted operating current	2010h, Subindex 0Ch *control_parameter_set.max_nominal_current*	mA	rw
$f_{start-up}$	Startup/peak current factor	2010h, Subindex 0Ch *control_parameter_set.max_nominal_factor*	%	rw
t_{Delay}	Time delay after motor start	2010h, Subindex 0Ch *control_parameter_set.deviation_delay*	Milliseconds	rw
t_I	Time interval for overload detection	2010h, Subindex 0Fh *control_parameter_set.current_deviation_events*	–	rw
$\Delta I_{Overcurrent}$	Excess current window	2010h, Subindex 0Ch *control_parameter_set.current_window*	mA	rw
$I_{operation}$	Operating current	6073h *max_current*	1/1,000 of the nominal motor current	rw
$I_{Nominal}$	Nominal motor current	6410h, Subindex 03h *motor_data.nominal_motor_current*	mA	ro
I_{Max}	Maximum current of the electronic circuit	6510h, Subindex 01h *drive_data.max_drive_current*	mA	ro

Status objects for current monitoring

The status object values are required to determine various quantities.

Symbol	Parameter	Unit
$I_{start-up}$	Startup/peak current ($I_{start-up} = f_{start-up} * I_{operation}$)	1/1,000 of the nominal motor current
t_{Ramp}	Time used to accelerate to the set profile speed	–
t_0	Start-up phase : $t_{Ramp} + t_{Delay}$	milliseconds

The following secondary conditions apply:

- $I_{Operate} \leq I_{Operatemax} \leq I_{Nominal} \leq I_{Max}$
- $1 \leq f_{start-up} \leq 1.5$
- $I_{start-up} \leq I_{Max}$

Maximum permitted operating current $I_{Operatemax}$

The parameter Maximum Permitted Operating Current limits the adjustable operating current. The value set in this parameter must not exceed the nominal motor current.
Factory setting: IOperatemax is equal to nominal current.

5960 0100 011, -000, 04.01

Starting current/peak current factor $f_{Start-up}$

The parameter Starting current/peak current factor determines the maximum starting current and the peak current in transient operation. This factor can be varied between 100% and 150%.

Time delay after motor start t_{Delay}

The parameter Time Delay After Motor Start determines the interval that is required after the start-up phase (starting ramp) to ensure that the results of monitoring functions are not aliased by transient operations.

Time interval for overload detection t_I

An error counter is incremented oinstantaneously if the motor current in transient state exceeds the operating current limit. Below the operating current limit the counter is decremented down to zero. The error counter respectively registers one event per control cycle.

The overload error flag (status register bit 21) is set if the counter reaches the count limit for overload and overcurrent indication.
The overcurrent error flag (status register bit 19) is set if the current limiting circuit responds.

Overcurrent window $\Delta I_{Overcurrent}$

Due to different system resolutions of current limiting and current monitoring the measured current value might be smaller than the peak current, while the actual value is higher. The motor runs with current limit.

The parameter Overcurrent Window determines a limit below peak current. If the motor current exceeds this limit, the controller determines overcurrent state. (ILimit = IStart-up - DIOvercurrent).

Operating current $I_{Operation}$

The parameter Operating Current can be used to set the operating current for any run job and, thus, to limit torque. The operating current is limited by the Maximum Operating Current.

Nominal motor current $I_{Nominal}$

The parameter Motor Nominal Current is dependednt on the design of the geared motor. It is listed in the electronic data sheet. The nominal motor current must be smaller than the maximum current of the electronic circuit.

Maximum current of the electronic circuit I_{Max}

The parameter Maximum Current of the Electronic Circuit determines the maximum current the control electronic circuit can process. This value is detemined by hardware. The software ensures that Imax is not exceeded by any other current value.

Starting/peak current $I_{Start-up}$

The status object Starting/Peak Current is the result of a multiplication of the operating current by the Start- up current/Peak current factor. The value of the start-up/peak current must be smaller than the maximum current of the electronic circuit.

If the motor current in transient state exceeds the peak current over a determined period of time (“Time interval for overload detection”) the error “Overcurrent/Peak current” (EMCY 2222h) is triggred. An overcurrent window can be determined in parameter DIOvercurrent to display this event.

Acceleration time t_{Ramp}

The status object Acceleration Time determines the minimum time required by the drive to reahc the set- point rpm.

Start-up phase t_0

This start-up phase is the sum of the start-up ramp and the time delay after motor start. After this time the controller enables the monitoring functions for the detection of overcurrent, overload and rpm offset.

5960 0100 011, -000, 04.01

6.3 Drive functions

6.3.1 Interruption to movement

During a movement command the drive can be stopped over the field bus. If bit 8 „Halt“ in the *controlword* object (6040h) changes to „1“, the motor is stopped using the deceleration ramp, which has been set for the movement command. Movement and position data are retained.

The drive resumes an interrupted movement command as soon as bit 8 has changed back to „0“, and bit 6, „new setpoint“ in the control word, has been set.

6.3.2 Direction reversal

A new target position is calculated by multiplying a position value by the current direction factor.

direction factor

The direction factor is set on bit 7 in the *polarity* object (607Eh), and can take on the value +1 or -1.

Object	Explanation
polarity (607Eh)	Bit 7 = 0: direction factor = +1 Bit 7 = 1: direction factor = -1

Direction and direction factor are defined when looking head on at the gearbox shaft when moving to high position values:

- Factor +1: sense of rotation in a clockwise direction
- Factor - 1: sense of rotation in an anti-clockwise direction.

6.3.3 Target position reached

At the end of a positioning process, minimal discrepancies arise in the target area between set position and actual position, and these can no longer be compensated for by further movements of the motor.

In order to report the target position as having been reached, a symmetrical window is defined through the *Control parameter set* object (2010h), sub-index 05h. If the drive stops within the area of the window, the target position is regarded as reached. Bit 10 in the status word, the *statusword* object (6041h) and bit 25 in the *Manufacturer status register* object (1002h) are set.

If T_PDO1 is transmitted on an event-driven basis, the drive transmits a PDO message on reaching the target position.

6.3.4 Monitoring digital inputs

The signal states of the input signals can be recorded over the field bus. For example this allows the start of a manual movement to be monitored on the field bus through the interface signals.

The *Digital inputs* object (60FDh) evaluates the status of the following digital inputs:

Bit	Signal
16, 17	MAN_N, MAN-P signal
27	emergency shutdown signal

5960 0100 011, -000, 04.01

6.3.5 Log-book

Two electronic log-books record data from the running operation. The following information is saved

- Length of time drive is switched on
- Total operating time of drive
- Number of positioning operations

When power to the drive is switched off, the positioning device uses the voltage from the internal charging capacitor to transfer the operational data to the first log-book. The log-book can be read by means of the *Drive log-book* object (200Eh).

Every twelve hours of operating time, the data are also saved to the second log-book. If the drive has not had time to record the operating data in the first log-book, it is reconstructed using the data from the second log-book.

6.4 Memory locations of the positioning device

The positioning device administers several separate memory locations in order to save operating software, system data and application data. When the drive is started up, the memory locations are checked. If a fault has occurred, the drive transmits an EMCY message. The *Manufacturer status register* object (1002h) displays the fault on bits 13 to 15.

Application memory

All data required for the drive's CAN and CAN-open application are saved in the application memory. These include

- Identification parameters such as serial number and date of manufacture
- CAN interface data such as node address and transmission rate
- CANopen communications parameters
- CANopen application parameters
- CANopen manufacturer-specific parameters
- Log-book and operating time data

The application memory is designed as a power fail safe data memory. Factory settings for all parameter values are stored in the positioning drive.

Changing application parameters

In order to adapt them to the particular application and network operation, parameters which are enabled for write access, can be changed and saved using the *Store parameters* object (1010h). They are then immediately available the next time the drive is switched on.

In order to avoid data being saved unintentionally, the „save“ identifier must be transferred on.

5960 0100 011, -000, 04.01

CANopen example „Saving data“

ASCII-encoded, „save“ is: 65h 76h 61h 73h,

COB-ID	Data	Explanation
601	23 10 10 03 73 61 76 65	R_SDO: Save application parameters
581	60 10 10 03 xx xx xx xx	T_SDO: OK

If the wrong identifier is given, the save command is not executed. T_SDO then supplies an error message by way of response, which can be seen in the value 80h in byte 1 of the SDO response.

Loading factory settings

The parameter values of the factory settings can be restored using the *Restore default parameters* object (1011h). In the same way as for the save command, an identifier must be entered, in this case the word „load“, to avoid unintentional write actions. ASCII-encoded, „load“ is: 6Ch 6Fh 61h 64h.

5960 0100 011, -000, 04.01

5960 0100 011, -000, 04.01

7 Diagnostics and Error Correction

7.1 Error diagnosis field bus communication

In order to be able to evaluate operating messages and error messages, it is essential that field bus operation is working. The positioning device can only exchange signals across the field bus.

connections for field bus operation

If the drive cannot be contacted over the field bus, first check the connections. You will find technical data on the unit as well as information on network and device installation in the manual on the positioning device. Check:

- 24 V power supply
- power supply connections to the unit
- field bus cables and wiring
- network connections to the unit

function test on the field bus

If the positioning device is correctly connected, test field bus operation. To do so, a CAN monitor must be installed which displays CAN messages.

The response from the drive is recorded in the form of a boot-up message:

▶ Switch the power supply to the drive off and back on again.

▶ Watch the network messages shortly after the drive is switched on. The positioning device transmits a 1 byte long boot-up message after bus initialization: 128 (80h)+node-ID.

With the node address at its factory setting of 127 (7Fh), boot-up message 255 (FFh) is transmitted over the bus. The drive can then be started up via LMT and NMT services.

If no boot-up message is received on the network after the drive has been switched on, first check if the CAN monitor baud rate matches the rate set on the network.

If all the network settings are correct and still no message can be received, the drive's network functionality must be checked by SIG Positec. Contact a member of the SIG Positec Automation customer service team.

baud rate and address

If no connection can be established with a network device, check the baud rate and node address.

- The baud rate must be set at the same rate for all network devices.
- The node address for every device must lie between 1 and 127, and must be set differently for every device.

Baud rate and node address for the positioning device can be set via LMT services, see page 3-32.

5960 0100 011, -000, 04.01

7.2 Error diagnosis via field bus

7.2.1 Reporting objects

There are several objects for supplying information on the operating status and error status of the positioning device:

- *Statusword* object (6041h),
  Operating states which are reported via the status word, are described in the Chapter entitled „Operate states and Operating modes“ on page 5-8.
- *EMCY* object (80h+ node-ID),
  Error message from a network device with error status and error code, see Chapter entitled „Emergency service“ from page 3-23
- *Manufacturer status register* object (1002h),
  Operating status and error status of device components, the manufacturer-specific signals on the positioning device are stored here.
- *Error register* object (1001h) error status
- *Error code* object (603Fh) error code of the last error to occur
- *Predefined error field* object (1003h) error memory, contains the last five error messages with error code.
- Network devices use a special SDO error message to report faulty message exchange by SDO.

Confirmed LMT services send acknowledgment information or else an error code. The significance of the response to each LMT service is described in the Chapter entitled „CANopen Communication“ from page 3-1 .

7.2.2 Signals on the device status

The 32-bit object Manufacturer Status Register (1002h) of the motor operating program is the central register of the motor interface.

Bits 24 - 31 give information on the operate state of the motor controls.

Bits 0 -22 give information on the error state of the various monitoring functions.

Errors occurred can be evaluated in detail via the bits 0 - 23.

For the purposes of error evaluation and error handling, a distinction is made between reversible and fatal errors as well as warning signals.

5960 0100 011, -000, 04.01

Reversible errors (Rev) If the drive detects a reversible error, it transmits an EMCY message over the field bus and stops the current operation. After the error has been corrected, the error message must be acknowledged via bit 7, „Reset fault“ in the control word *controlword* (6040h).

With a reversible error, bit 7 in the *Statusword* object (6041h) and bit 30 in the *Manufacturer status register* object (1002h) are set.

Fatal errors (fatal) A fatal error occurs when the internal device test detects hardware or software errors. The motor controls are disabled; this lock can only be released by switching the drive off and on again.

The drive transmits an EMCY message.

Warning notice The „Info“ entry in the column „type of signal“ shows that a warning is being issued with no interruption to the current operation.

The following table shows the bit values of the *Manufacturer status register* object (1002h).

Bit	Designation	Message cause	Message type	Remedy
0	Configuration error control parameters	read/write error when copying the control parameters from configuration memory in the EEPROM	Rev	Check and correct the plausibility of the user configuration or reset to factory default configuration (Bit 30:1)
			Fatal	Pending memory error, replace the drive (Bit 31:1)
1	Configuration error drive data	read/write error when copying the drive data from configuration memory in the EEPROM	Fatal	Replace the drive
2	Self-test error Hardware Emergency Stop	Error in the self-test phase when testing the hardware Emergency Stop circuit	Fatal	Defective Emergency Stop function, replace the drive
3	Low voltage auxiliary voltage	Drop below tolerance or switch-off limit	Rev	Check the 24 V supply; restart the drive; falls If the error persists replace the drive
4	Overvoltage intermediate circuit	Exceeding the tolerance or switch-off limit	Rev	On warning (Bit 30): check the voltage supply, restart the drive On error (Bit 31): Switch off the voltage supply immediately, check the voltage supply
5	Low voltage intermediate circuit	Drop below tolerance or switch-off limit of the intermediate circuit	Rev	check the voltage supply if required
6	Overvoltage voltage supply	Exceeding the tolerance limit of the voltage supply	Rev	check the voltage supply
7	Low voltage voltage supply	Drop below tolerance or switch-off limit	Rev	check the voltage supply
8	Oscillator Watchdog	Excess offsset of the oscillator frequency	Fatal	replace the drive
9	Watchdog Timer Reset	Reset of the microcontroller	Fatal	dispatch the drive for diagnostics
10	Class B Hardware Trap	Error message from internal monitoring function	Fatal	replace the drive
11	Class A Hardware Trap	Error message from internal monitoring function	Fatal	replace the drive
12	Error system timer	Error during the self-test phase when testing the system timer	Fatal	replace the drive

5960 0100 011, -000, 04.01

Bit	Designation	Message cause	Message type	Remedy
13	EEPROM error (Configuration memory)	Error when testing the Konfigurationsspeichers während der Selbsttestphase	Fatal	replace the drive
14	ROM error (Program memory)	Error during the self-test phase when testing the program memory	Fatal	replace the drive
15	RAM error (Data memory)	Error during the self-test phase when testing the data memory	Fatal	replace the drive
16	Unexpected Hall sensor combination	Undefined Hall sensor combination or sequenz during the commutation sequence	Rev/Fatal	On warning (Bit 30): check the mechanical and logical drive position; referencing On error (Bit 31): replace the drive
17	Startup error/motor does not turn	No Hall sensor signals after the startup phase	Rev	Reduce the load; Increase the operating/peak current; Increase the acceleration ramp value
18	Speed offset	Speed offset exceeds the limit value after the acceleration time and the limit of the speed monitoring counter is exceeded	Rev	Reduce the load; Increase the peak current; Increase the acceleration ramp value
19	Overcurrent (current limiting)	The current of the motor in transient state exceeds the permited peak value over a longer period of time.	Info	Reduce the load
20	Running on block	Drive deceleration and overcurrent as a result of running into an obstruction	Rev	Eliminate the cause of running on block (obstruction, mechanical stop) In Fieldbus mode: Execute *Reset Fault* (Object 6040h, Bit 7=1) In manual mode: Switch off the voltage supply; switch on the voltage supply again after fault elimination
21	Overload (operating current)	The measured motor current exceeds the operating current over a longer period of time.	Info	Reduce the load or increase the operating current
22	Excess temperature end amplifier	Temperature tolerance exceeded	Rev	Drive operation can be resumed when the temperature has dropped below the limit.
23	reserved	–	-	–
24	Drive referenced	1: Drive is referenced 0: Drive is not referenced, the drive was probably shifted while the voltage supply was switched off.	Rev	Measure the mechanical dimensions; Reference the drive position again via Fieldbus
25	Positioning run	1: The drive did not reach the target position window after having executed a run job	Info	–
26	Motor is turning	1: Motor rotates (active Hall sensor signals) 0: Motor does not turn	Info	–

5960 0100 011, -000, 04.01

Bit	Designation	Message cause	Message type	Remedy
27	Motorsteuerung freigegeben	1: The motor operating program is ready to carry out run jobs. Prerequisites: - no error occurred - valid operating mode is set - drive is referenced, if positioning mode is to be carried out	Info	–
28	reserviert	–	-	-
29	Hardware Emergency Stop active	Hardware E-Stop signal is active	Rev	If an emergency situation is pending: In Fieldbus mode: – Connect the hardware E-stop signal to the potential of the supply voltage – Execute *Reset Fault* (Object 6040h, bit 7=1) In manual mode without Fieldbus usage: – Switch off the voltage supply – Connect the hardware E-stop signal to the potential of the supply voltage – Switch on the voltage supply again
30	Warning/note	The drive is in a critical state that does not endanger operation. Details on the cause are stored in the bits 2 - 23.	Info	Refer to the information on bits 2 - 23
31	Error/disruption	Error or disruption in a monitoring function of the motor operating program. Details on the cause are stored in the bits 2 - 23.	Info	Refer to the information on bits 2 - 23

5960 0100 011, -000, 04.01

7.3 CANopen error messages

CANopen error messages are displayed via an EMCY message. They are evaluated by means of the *Error register* (1001h) and *Error code* (603Fh) objects. You will find information on the EMCY object in the chapter entitled „Emergency service“ from page 3-23.

Errors occurring in data exchange by SDO, are reported by CANopen by means of a special SDO error message.

7.3.1 Error register

The *Error register* object (1001h) displays the error status of a network device in bit-encoded form. The exact cause of the error must be determined from the error code table. Bit 0 is set as soon as an error occurs.

Bit	Report	Explanation
0	generic error	An error has occurred.
1	current	error in current levels
2	voltage	error in voltage levels
3	temperature	temperature error
4	communication	error in network communication
5	Device profile-specific	error in carrying out device profile
6	reserved	reserved
7	manufacturer-specific	manufacturer-specific error message

5960 0100 011, -000, 04.01

7.3.2 Error code table

The error code is evaluated through the *Error code* object (603Fh), an object belonging to the DSP 402 device profile.

Error code	Name	Explanation
no data	-	Boot-up message
0000h	no error	no error
1000h	generic error	general error coding
2221h	continuous over-current No.1	nominal current over-current (overload)
2222h	continuous over-current No.1	peak current over-current (current limitation)
3110h	mains over-voltage	power supply over-voltage
3120h	mains under-voltage	power supply under-voltage
3210h	DC line over-voltage	motor DC line over-voltage
3220h	DC link under-voltage	motor DC line under-voltage
4310h	excess temperature drive	excess temperature in power amplifier
5000h	device hardware	self test error hardware (here emergency stop circuit)
5110h	supply low voltage	low voltage in VDD auxiliary voltage
5510h	data storage RAM	self test error data memory
5520h	data storage ROM	self test error program memory
5530h	data storage EEPROM	self test error configuration memory
6000h	device software	self test error micro-controller, internal error
6010h	software reset (watchdog)	software reset by watchdog timer
6310h	loss of parameters	loss of parameters (configuration memory)
7120h	motor	gen. motor fault (motor start-up, speed variation)
7121h	motor blocked	movement blocked
7122h	motor error or commutation	Hall error
FF00h	device-specific	additional drive/motor fault
FF01h	emergency stop	hardware emergency stop
FF10h	general application	application: general error
FF11h	illegal mode application	application: invalid mode
FF12h	illegal parameter application	application: invalid parameter value
FF13h	position value application	application: invalid position value
FF14h	usrerr poslim	application: invalid location memory
FF20h	general drive controller	motor controller: general error
FF21h	power drive enabled	motor controller: controller not ready
FF22h	no reference	motor controller: drive not referenced
FF23h	drive moves	motor controller: drive still moving
FF24h	illegal mode drive controller	motor controller: invalid mode
FF25h	illegal parameter drive controller	motor controller: invalid parameter value
FF26h	parameter too low	motor controller: parameter value too low
FF27h	parameter too high	motor controller: parameter value too high
FF28h	position not accepted	motor controller: position not accepted/processed

5960 0100 011, -000, 04.01

7.3.3 SDO error message

An SDO error message is issued in response to an SDO error transmission. The cause of the error is given in the „error code“, byte 4 to byte 7.

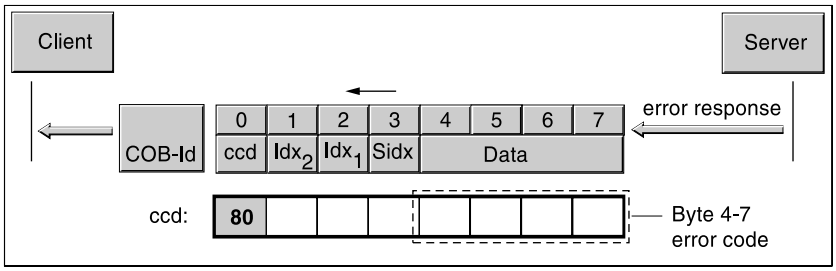

Fig. 7.1 SDO error message in response to SDO message

The following table shows all error messages which can occur in the exchange of data with the positioning device.The byte sequence must be converted before evaluation in accordance with the Intel format.

Example:
Error code 0607 0013h is transmitted as 1300 0706h
byte 7: 06h, byte 6: 07h, byte 5: 00h, byte 4: 13h

Error code	Explanation
0503 0000h	reversal bit not switched
0602 0000h	object not present in object dictionary
0604 0041h	object does not support PDO mapping
0604 0043h	parameters not compatible
0604 0047h	device has detected internal conflict
0606 0000h	hardware fault, access denied
0607 0010h	data type and parameter length do not match
0607 0012h	data type mismatch, parameter too long
0607 0013h	data type mismatch, parameter too short
0609 0011h	sub-index not supported
0609 0031h	parameter values too large
0609 0032h	parameter values too small
0800 0000h	general error
0800 0021h	device monitoring carried out locally, data cannot be transferred or saved.
0800 0022h	device status preventing transfer and storage of data

5960 0100 011, -000, 04.01

8 Service, Maintenance and Warranty

8.1 Service address

If you have queries or problems, please refer them to your SIG Positec contact person or directly to SIG Positec Automation. SIG Positec Automation will be happy to give you the name of their customer service in your area.

SIG Positec Automation GmbH
Breslauer Str. 7
D-77933 Lahr

Tel.: (07821) 946 - 02
Fax: (07821) 946 - 220

http://www.sig-positec.de

Hardware hotline

questions on devices, service, set-up on site

Telephone: +49 (0) 7821 - 946 - 257
Fax: +49 (0) 7821 - 946 - 430

Lotus Notes: Hotline, Hardware
Internet e-mail: hw.hotline@sig-positec.de

Software hotline

questions on software, field bus

Telephone: +49 (0) 7821 - 946 - 360
Fax: +49 (0) 7821 - 946 - 430

Lotus Notes: Hotline, Software
Internet e-mail: sw.hotline@sig-positec.de

RSS office

Repairs and Spare Parts Service

Telephone: +49 (0) 7821 - 946 - 606
Fax: +49 (0) 7821 - 946 - 202

Lotus Notes: RED, Buero
Internet, e-mail: red@sig-positec.de

5960 0100 011, -000, 04.01

5960 0100 011, -000, 04.01

9 Object Directory

9.1 Overview

9.1.1 Object Specifications

Index The index identifies the position of the object in the object directory.The index value is given in hexadecimal.

Object code The object code identifies the data structure of the object.

Object code	Explanation	Coding
VAR	A simple value, which for example is of the Integer8, Unsigned32 or Visible String8 type.	7
ARR (ARRAY)	A data field in which every entry is of the same data type.	8
REC (RECORD)	EA data field which contains entries which are a combination of simple data types.	9

Data type	Value range	Data length
Boolean	0=false, 1=true	1 Byte
Integer8	-128 .. +127	1 Byte
Integer16	-32768 .. +32767	2 Byte
Integer32	-2147483648 .. +2147483647	4 Byte
Unsigned8	0 .. 255	1 Byte
Unsigned16	0 .. 65535	2 Byte
Unsigned32	0 .. 4294967295	4 Byte
Visible String8	ASCII Zeichen	8 Byte
Visible String16	ASCII Zeichen	16 Byte

Access **ro**: read-only, value can only be read

rw: read write: value can be read and written to

wo: write only: value can only be written to

PDO **R_PDO**: Mapping for R_PDO possible

T_PDO: Mapping for T_PDO possible

no information: PDP mapping not possible with the object

Value range Specifies the permissible range in which the object value is defined and valid.

Default value Default values can be adjusted by loading and initializing with the stored factory settings.

storable ✔: values can be saved in the positioning drive's memory, and are available when the unit is switched on again.

–: values are lost when the positioning drive is switched off.

5960 0100 011, -000, 04.01

9.1.2 Objects sorted by object name

Name	Index	Sub-index	Obj. Code	Data type	Access	PDO	Description	Page
1st mapped object	1600h	01h	VAR	Unsigned32	ro		First object for mapping in R_PDO1	9-26
1st mapped object	1601h	01h	VAR	Unsigned32	ro		First object for mapping in R_PDO2	9-27
1st mapped object	1A00h	01h	VAR	Unsigned32	ro		First object for mapping in T_PDO1	9-29
1st mapped object	1A01h	01h	VAR	Unsigned32	ro		First object for mapping in T_PDO2	9-30
1st receive PDO mapping	1600h		REC	PDO Mapping	-		PDO mapping for R_PDO1, settings	9-26
1st receive PDO parameter	1400h		REC	PDO comm. param.	-		First receive PDO R_PDO1, settings	9-23
1st transmit PDO mapping	1A00h		REC	PDO Mapping	-		PDO mapping for T_PDO1, settings	9-29
1st transmit PDO parameter	1800h		REC	PDO comm. param.	-		First transmit PDO T_PDO1, settings	9-27
2nd mapped object	1601h	02h	VAR	Unsigned32	ro		Second object for mapping in R_PDO2	9-27
2nd mapped object	1A01h	02h	VAR	Unsigned32	ro		Second object for mapping in T_PDO2	9-30
2nd receive PDO mapping	1601h		REC	PDO Mapping	-		PDO mapping for R_PDO2, settings	9-27
2nd receive PDO parameter	1401h		REC	PDO comm. param.	-		Second receive PDO R_PDO2, settings	9-25
2nd transmit PDO mapping	1A01h		REC	PDO Mapping	-		PDO mapping for T_PDO2, settings	9-30
2nd transmit PDO parameter	1801h		REC	PDO comm. param.	-		Second transmit PDO T_PDO2, settings	9-28
acceleration	2011h	03h	VAR	Unsigned32	rw		Manual mode, acceleration [Inc/s^2]	9-48
application software	2003h		VAR	Visible String	ro		Software version of application program	9-32
block deceleration	2010h	11h	VAR	Unsigned16	rw		Block movement deceleration in [1/(min×ms)]	9-43
COB-ID Client -> Server (rx)	1200h	01h	VAR	Unsigned32	ro		Identifier for receive SDO R_SDO	9-22
COB-ID emergency message	1014h		VAR	Unsigned32	rw		Identifier for Emergency object	9-20
COB-ID Server -> Client (tx)	1200h	02h	VAR	Unsigned32	ro		Identifier for transmit SDO T_SDO	9-22
COB-ID SYNC-message	1005h		VAR	Unsigned32	rw		Identifier for synchronisation object	9-13
COB-ID used by PDO	1400h	01h	VAR	Unsigned32	rw		Identifier for R_PDO1	9-23
COB-ID used by PDO	1401h	01h	VAR	Unsigned32	rw		Identifier for R_PDO2	9-25
COB-ID used by PDO	1800h	01h	VAR	Unsigned32	rw		Identifier for T_PDO1	9-27
COB-ID used by PDO	1801h	01h	VAR	Unsigned32	rw		Identifier for T_PDO2	9-28
config bits	6510h	07h	VAR	Unsigned8	ro		used internally	9-70
control parameter set	2010h		REC	User Defined	-		Adjustable control parameters	9-43
controlword	6040h		VAR	Integer16	rw	R_PDO	Control word for changing operating status	9-50

5960 0100 011, -000, 04.01

Name	Index	Sub-index	Obj. Code	Data type	Access	PDO	Description	Page
current actual value	6078h		VAR	Integer16	ro		Current motor current [‰], (rated motor current [mA] /1000 mA)	9-56
current deviation events	2010h	0Fh	VAR	Unsigned16	rw		Event counter limit for overload and overcurrent	9-43
current window	2010h	10h	VAR	Unsigned16	rw		Overcurrent window [mA]	9-43
DC link circuit voltage	6079h		VAR	Unsigned32	ro		Voltage in motor DC line [mV]	9-56
deceleration	2011h	04h	VAR	Unsigned32	rw		Manual mode, deceleration [Inc/s^2]	9-48
deviation delay	2010h	0Eh	VAR	Unsigned8	rw		Time delay in measurement of speed variations [ms]	9-43
device type	1000h		VAR	Unsigned32	ro		Device type and profile	9-9
digital inputs	60FDh		VAR	Unsigned32	ro		Signal status of digital inputs	9-65
drive data	6510h		REC	User Defined	-		Electronic data sheet for electronic controls	9-70
drive hardware version	2000h		VAR	Visible String	ro		Hardware version of electronic controls	9-31
drive log book	200Eh		REC	User Defined	-		Drive log-book	9-41
drive manufacturer	6504h		VAR	Visible String	ro		Drive manufacturer	9-69
drive serial-number	200Fh		VAR	Visible String	ro		Serial number of drive	9-43
encoder increments	608Fh	01h	VAR	Unsigned32	ro		Position increments	9-62
enforced commut	6510h	08h	VAR	Unsigned16	ro		used internally	9-70
error code	603Fh		VAR	Integer16	ro		Last error to have occurred	9-50
error register	1001h		VAR	Unsigned8	ro		Error register	9-9
excess temperature	6510h	02h	VAR	Unsigned8	ro		Maximum operating temperature [°C]	9-70
gear detent torque	6410h	0Ah	VAR	Unsigned16	ro		Gear self-holding torque [1/1000 Nm]	9-66
gear efficiency	6410h	08h	VAR	Unsigned16	ro		Gear efficiency [‰]	9-66
gear motor revolutions	6410h	06h	VAR	Unsigned32	ro		Revolutions of motor gear shaft	9-66
gear ratio	6091h		ARR	Unsigned32	-		Gear ratio	9-63
gear shaft revolutions	6410h	05h	VAR	Unsigned32	ro		Revolutions at gear output	9-66
gear stages	6410h	07h	VAR	Unsigned16	ro		Number of gear stages	9-66
guard time	100Ch		VAR	Unsigned16	rw		Time span for node guarding [ms]	9-14
hall sensors	6510h	06h	VAR	Unsigned8	ro		Number of Hall sensors	9-70
holding torque time	2010h	0Bh	VAR	Unsigned8	rw		Duration of holding torque [ms]	9-43
homing method	6098h		VAR	Integer8	rw		Homing method	9-63
increments	2011h	01h	VAR	Integer32	rw		Manual mode, number of motor increments	9-48
k p cmd	2010h	03h	VAR	Unsigned16	rw		Reference variable speed controller [‰]	9-43
k p pos	2010h	01h	VAR	Unsigned16	rw		P-amplification position controller [‰]	9-43
k p rpm	2010h	02h	VAR	Unsigned16	rw		P-amplification speed controller [‰]	9-43
life time factor	100Dh		VAR	Unsigned8	rw		Repeat factor for node guarding protocol.	9-15

5960 0100 011, -000, 04.01

Name	Index	Sub-index	Obj. Code	Data type	Access	PDO	Description	Page
lp1 time const	2010h	0Ah	VAR	Unsigned8	rw		Time constant speed low-pass filter [ms]	9-43
manual mode settings	2011h		REC	User Defined	-		Settings for inching mode	9-48
manufacturer device name	1008h		VAR	Visible String	ro		Device name	9-13
manufacturer status register	1002h		VAR	Unsigned32	ro		Status of motor and configuration	9-10
max acceleration	60C5h		VAR	Unsigned32	ro		Maximum permissible profile acceleration [Inc/s^2]	9-64
max current	2011h	05h	VAR	Unsigned16	rw		Rated current for manual mode, [‰], (rated motor current[mA] /1000 mA)	9-48
max current	6073h		VAR	Unsigned16	rw		Rated current for bus operation [‰], (rated motor current [mA] /1000 mA)	9-55
max deceleration	60C6h		VAR	Unsigned32	ro		Maximum permissible profile deceleration [Inc/s^2]	9-65
max drive current	6510h	01h	VAR	Unsigned16	ro		Maximum current [mA]	9-70
max drive speed	6510h	09h	VAR	Unsigned16	ro		Maximum idle speed of motor [r.p.m.]	9-70
max motor speed	6080h		VAR	Unsigned16	ro		Maximum motor speed [r.p.m.]	9-60
max position limit	2008h	02h	VAR	Integer32	rw		Upper software limit switch S1 [Inc]	9-38
max position limit	2009h	02h	VAR	Integer32	rw		Upper software limit switch D1 [Inc]	9-39
max position limit	607Dh	02h	VAR	Integer32	rw		Upper software limit switch W1 [Inc]	9-58
max profile velocity	607Fh		VAR	Unsigned32	ro		Maximum permissible profile speed [Inc/s]	9-60
max speed	6410h	01h	VAR	Unsigned16	ro		Maximum speed at gear output [r.p.m.]	9-66
max steady current	2010h	0Ch	VAR	Unsigned16	rw		Rated current for application [mA]	9-43
min position limit	2008h	01h	VAR	Integer32	rw		Lower software limit switch S_0 [Inc]	9-38
min position limit	2009h	01h	VAR	Integer32	rw		Lower software limit switch D_0 [Inc]	9-39
min position limit	607Dh	01h	VAR	Integer32	rw		Lower software limit switch W_0 [Inc]	9-58
min ramp acceleration	6510h	04h	VAR	Unsigned16	ro		Minimum acceleration ramp [ms]	9-70
min ramp deceleration	6510h	05h	VAR	Unsigned16	ro		Minimum deceleration ramp [ms]	9-70
modes of operation	6060h		VAR	Integer8	wo		Set operating mode	9-53
modes of operation display	6061h		VAR	Integer8	ro		Display current operating mode	9-54
motor calibration date	6406h		VAR	Date	ro		Manufacturing date or last repair date	9-66
motor data	6410h		REC	User Defined	-		Electronic data sheet for motor	9-66
motor hardware version	2001h		VAR	Visible String	ro		Hardware version of motor mechanics	9-31
motor manufacturer	6404h		VAR	Visible String	ro		Motor manufacturer	9-65
motor operating software	2002h		VAR	Visible String	ro		Software version of motor operating program	9-31
motor rated current	6075h		VAR	Unsigned32	ro		Rated motor current [mA]	9-56
motor revolutions	608Fh	02h	VAR	Unsigned32	ro		Motor revolutions	9-62

5960 0100 011, -000, 04.01

Name	Index	Sub-index	Obj. Code	Data type	Access	PDO	Description	Page
motor revolutions	6091h	01h	VAR	Unsigned32	ro		Motor revolutions	9-63
motor torque constant	6410h	04h	VAR	Unsigned16	ro		Motor torque constant [1/1000 Ncm/A]	9-66
node guarding identifier	100Eh		VAR	Unsigned32	rw		Identifier for node guarding protocol	9-16
node-ID	100Bh		VAR	Unsigned32	ro		7 bit node address	9-14
nominal gear torque	6410h	09h	VAR	Unsigned16	ro		Nominal gear torque [1/1000 Nm]	9-66
nominal motor current	6410h	03h	VAR	Unsigned16	ro		Rated current of motor [mA]	9-66
nominal speed	6410h	02h	VAR	Unsigned16	ro		Rated speed at gear output [r.p.m.]	9-66
number of asynchronous PDOs	1004h	02h	VAR	Unsigned32	ro		Number of asynchronous R_PDOs, T_PDOs	9-12
number of errors	1003h	00h	VAR	Unsigned8	rw		Number of error entries	9-10
number of log book loss	200Eh	05h	VAR	Unsigned16	ro		Number of log-book losses	9-41
number of PDOs supported	1004h		ARR	Unsigned32	-		Number of PDOs supported	9-12
number of PDOs supported	1004h	00h	VAR	Unsigned32	ro		Number of PDOs supported	9-12
number of positionings	200Eh	03h	VAR	Unsigned32	ro		Number of positioning operations	9-41
number of reference loss	200Eh	04h	VAR	Unsigned16	ro		Number of reference losses	9-41
number of SDOs supported	100Fh		VAR	Unsigned32	ro		Number of SDOs supported	9-16
number of synchonous PDOs	1004h	01h	VAR	Unsigned32	ro		Number of synchronous R_PDOs, T_PDOs	9-12
overcurrent average value	200Ch		VAR	Integer16	ro		Average overload current [‰], (rated motor current [mA] /1000 mA)	9-40
polarity	607Eh		VAR	Unsigned8	rw		Sense of rotation factor	9-59
pole pairs	6410h	0Bh	VAR	Unsigned8	ro		Number of pole pairs	9-66
position actual value	6064h		VAR	Integer32	ro	T_PDO	Current position of drive [Inc]	9-54
position assignment value	200Bh		VAR	Integer32	wo		Position memory initialization	9-40
position encoder resolution	608Fh		ARR	Unsigned32	-		Resolution of position capture	9-62
position window	2010h	05h	VAR	Unsigned32	rw		Tolerance window for target position [ms]	9-43
predefined error field	1003h		ARR	Unsigned32	-		Error history, memory for error messages	9-10
profile acceleration	6083h		VAR	Unsigned32	rw		Acceleration preset for a positioning profile [Inc/s^2]	9-61
profile deceleration	6084h		VAR	Unsigned32	rw		Deceleration preset for a positioning profile [Inc/s^2]	9-61
profile velocity	6081h		VAR	Unsigned32	rw		Speed for a positioning profile [Inc/s]	9-60
r i lag	2010h	04h	VAR	Unsigned16	rw		Reset time for PI controller [ms]	9-43
release time	2011h	06h	VAR	Unsigned16	rw		Manual mode, maximum pulse duration [ms]	9-48

5960 0100 011, -000, 04.01

Name	Index	Sub-index	Obj. Code	Data type	Access	PDO	Description	Page
restore all default parameters	1011h	01h	VAR	Unsigned32	rw		Load all parameter values from the configuration memory	9-19
restore application default parameters	1011h	03h	VAR	Unsigned32	rw		Load applications parameters from the configuration memory	9-19
restore communication default parameters	1011h	02h	VAR	Unsigned32	rw		Load communications parameters from the configuration memory	9-19
restore default parameters	1011h		ARR	Unsigned32	-		Restore parameter values	9-19
rpm deviation events	2010h	07h	VAR	Unsigned16	rw		Event counter limit for speed monitoring	9-43
rpm start timeout	2010h	08h	VAR	Unsigned16	rw		Speed monitoring for motor start-up [ms]	9-43
rpm stop	2010h	09h	VAR	Unsigned16	rw		Speed monitoring when motor stop to 0 [r.p.m.]	9-43
rpm window	2010h	06h	VAR	Unsigned16	rw		Speed deviations when setpoint reached [r.p.m.]	9-43
save all parameters	1010h	01h	VAR	Unsigned32	rw		Write all parameters to configuration memory	9-17
save application parameters	1010h	03h	VAR	Unsigned32	rw		Write applications parameters to configuration memory	9-17
save communication parameters	1010h	02h	VAR	Unsigned32	rw		Write communications parameters to configuration memory	9-17
serial number	1018h	04h	VAR	Unsigned32	ro		Serial ID of the drive	9-21
server SDO parameter	1200h		REC	SDO Parameter	-		Server SDO, settings	9-22
shaft revolutions	6091h	02h	VAR	Unsigned32	ro		Revolutions, gear shaft	9-63
software position drive limit	2009h		ARR	Integer32	-		Software limit switches D_0-D_1 for travel area	9-39
software position limit	607Dh		ARR	Integer32	-		Software limit switches W_0-W1 for work area	9-58
software position safety limit	2008h		ARR	Integer32	-		Software limit switches S_0-S_1 for safety area	9-38
standard error field 1	1003h	01h	VAR	Unsigned32	ro		Last error which occurred	9-10
standard error field 2	1003h	02h	VAR	Unsigned32	ro		Older error message (2nd place in history)	9-10
standard error field 3	1003h	03h	VAR	Unsigned32	ro		Older error message (3rd place in history)	9-10
standard error field 4	1003h	04h	VAR	Unsigned32	ro		Older error message (4th place in history)	9-10
standard error field 5	1003h	05h	VAR	Unsigned32	ro		Older error message (5th place in history)	9-10
starting current factor	2010h	0Dh	VAR	Unsigned8	rw		Start-up factor [%] (maximum overload current)	9-43
statusword	6041h		VAR	Unsigned16	ro	T_PDO	Status word for evaluating operating status	9-51
store parameter	1010h		ARR	Unsigned32	-		Store parameter	9-17

5960 0100 011, -000, 04.01

Name	Index	Sub-index	Obj. Code	Data type	Access	PDO	Description	Page
supported drive modes	6502h		VAR	Unsigned32	ro		Operating modes supported	9-68
target position	607Ah		VAR	Integer32	rw	R_PDO	Target position (setpoint) [Inc]	9-57
temperature actual value	200Dh		VAR	Integer16	ro		Temperature of power amplifier in [°C]	9-41
temperature threshold	6510h	03h	VAR	Unsigned8	ro		Temperatur Schwellwert [°C]	9-70
total power-on time	200Eh	01h	VAR	Unsigned32	ro		Power-on time of drive [s]	9-41
total turn-on time	200Eh	02h	VAR	Unsigned32	ro		Turn-on time of drive [s]	9-41
transmission type	1400h	02h	VAR	Unsigned8	rw		Transmission type for R_PDO1	9-23
transmission type	1401h	02h	VAR	Unsigned8	rw		Transmission type for R_PDO2	9-25
transmission type	1800h	02h	VAR	Unsigned8	rw		Transmission type for T_PDO1	9-27
transmission type	1801h	02h	VAR	Unsigned8	rw		Transmission type for T_PDO2	9-28
user profile acceleration	2005h		ARR	Unsigned32	-		Acceleration for the 10 pre-configured movement profiles	9-34
user profile acceleration 1	2005h	01h	VAR	Unsigned32	ro		Movement profile 1: acceleration in [Inc/s^2]	9-34
user profile acceleration 10	2005h	0Ah	VAR	Unsigned32	rw		Movement profile 10: acceleration in [Inc/s^2]	9-34
user profile acceleration 2	2005h	02h	VAR	Unsigned32	rw		Movement profile 2: acceleration in [Inc/s^2]	9-34
user profile acceleration 3	2005h	03h	VAR	Unsigned32	rw		Movement profile 3: acceleration in [Inc/s^2]	9-34
user profile acceleration 4	2005h	04h	VAR	Unsigned32	rw		Movement profile 4: acceleration in [Inc/s^2]	9-34
user profile acceleration 5	2005h	05h	VAR	Unsigned32	rw		Movement profile 5: acceleration in [Inc/s^2]	9-34
user profile acceleration 6	2005h	06h	VAR	Unsigned32	rw		Movement profile 6: acceleration in [Inc/s^2]	9-34
user profile acceleration 7	2005h	07h	VAR	Unsigned32	rw		Movement profile 7: acceleration in [Inc/s^2]	9-34
user profile acceleration 8	2005h	08h	VAR	Unsigned32	rw		Movement profile 8: acceleration in [Inc/s^2]	9-34
user profile acceleration 9	2005h	09h	VAR	Unsigned32	rw		Movement profile 9: acceleration in [Inc/s^2]	9-34
user profile deceleration	2006h		ARR	Unsigned32	-		Deceleration for the 10 pre-configured movement profiles	9-36
user profile deceleration 1	2006h	01h	VAR	Unsigned32	ro		Movement profile 1: deceleration in [Inc/s^2]	9-36
user profile deceleration 10	2006h	0Ah	VAR	Unsigned32	rw		Movement profile 10: deceleration in [Inc/s^2]	9-36
user profile deceleration 2	2006h	02h	VAR	Unsigned32	rw		Movement profile 2: deceleration in [Inc/s^2]	9-36
user profile deceleration 3	2006h	03h	VAR	Unsigned32	rw		Movement profile 3: deceleration in [Inc/s^2]	9-36
user profile deceleration 4	2006h	04h	VAR	Unsigned32	rw		Movement profile 4: deceleration in [Inc/s^2]	9-36

5960 0100 011, -000, 04.01

Name	Index	Sub-index	Obj. Code	Data type	Access	PDO	Description	Page
user profile deceleration 5	2006h	05h	VAR	Unsigned32	rw		Movement profile 5: deceleration in [Inc/s^2]	9-36
user profile deceleration 6	2006h	06h	VAR	Unsigned32	rw		Movement profile 6: deceleration in [Inc/s^2]	9-36
user profile deceleration 7	2006h	07h	VAR	Unsigned32	rw		Movement profile 7: deceleration in [Inc/s^2]	9-36
user profile deceleration 8	2006h	08h	VAR	Unsigned32	rw		Movement profile 8: deceleration in [Inc/s^2]	9-36
user profile deceleration 9	2006h	09h	VAR	Unsigned32	rw		Movement profile 9: deceleration in [Inc/s^2]	9-36
user profile number	2007h		VAR	Unsigned8	rw		Selection of movement profile 1 to 10	9-38
user profile velocity	2004h		ARR	Unsigned32	-		Speeds for the 10 pre-configured movement profiles	9-32
user profile velocity 1	2004h	01h	VAR	Unsigned32	ro		Movement profile 1: speed in [Inc/s]	9-32
user profile velocity 10	2004h	0Ah	VAR	Unsigned32	rw		Movement profile 10: speed in [Inc/s]	9-32
user profile velocity 2	2004h	02h	VAR	Unsigned32	rw		Movement profile 2: speed in [Inc/s]	9-32
user profile velocity 3	2004h	03h	VAR	Unsigned32	rw		Movement profile 3: speed in [Inc/s]	9-32
user profile velocity 4	2004h	04h	VAR	Unsigned32	rw		Movement profile 4: speed in [Inc/s]	9-32
user profile velocity 5	2004h	05h	VAR	Unsigned32	rw		Movement profile 5: speed in [Inc/s]	9-32
user profile velocity 6	2004h	06h	VAR	Unsigned32	rw		Movement profile 6: speed in [Inc/s]	9-32
user profile velocity 7	2004h	07h	VAR	Unsigned32	rw		Movement profile 7: speed in [Inc/s]	9-32
user profile velocity 8	2004h	08h	VAR	Unsigned32	rw		Movement profile 8: speed in [Inc/s]	9-32
user profile velocity 9	2004h	09h	VAR	Unsigned32	rw		Movement profile 9: speed in [Inc/s]	9-32
velocity	2011h	02h	VAR	Unsigned32	rw		Manual mode, speed [Inc/s]	9-48
velocity actual value	606Ch		VAR	Integer32	ro		Current speed of drive [Inc/s]	9-55

5960 0100 011, -000, 04.01

9.2 Objects of the positioning drive

1000h *Device type*

The object gives details of the device profile and device type used.

Object description

Index	1000h
Object name	device type
PDO mapping	VAR
Object code	Unsigned32

Value description

Sub-index	00h, device type
Explanation	Device type and profile
Access	read-only
PDO mapping	–
Value range	–
Default value	0x00020192
Storable	✔

Bit coding, sub-index 00h

Bit	Access	Value	Explanation
31-24	ro	00_h	not used
23-16	ro	02_h	Bit17=1: actuator
15-0	ro	0192_h	Device profile DSP-402 (402=192h)

1001h *Error register*

The object displays the error status of the device. The exact cause of the error can be determined via the *error code* object (603Fh). The *pre-defined error field* object (1003h) displays the error history of the drive.

Errors are signalled when they occur by means of an EMCY message.

Object description

Index	1001h
Object name	error register
PDO mapping	VAR
Object code	Unsigned8

Value description

Sub-index	00h, error register
Explanation	Error register
Access	read-only
PDO mapping	–
Value range	–
Default value	–
Storable	–

5960 0100 011, -000, 04.01

Bit coding, sub-index 00h

Bit	Access	Value	Explanation, if bit = 1
0	ro	–	Error (generic error)
1	ro	–	Current
2	ro	–	Voltage
3	ro	–	Temperature
4	ro	–	Communication error
5	ro	–	Device profile error
6	ro	–	reserved
7	ro	–	manufacturer-specific)

If a bit is set, it displays an error message. If sub-index 00h=0, there are no errors.

1002h *Manufacturer Status Register*

The object displays the operating status and error status of the positioning device in bit coded form. The significance of each bit status is set by the manufacturer. An operating status can also be requested via the *statusword* object (6041h).

Object description

Index	1002h
Object name	manufacturer status register
PDO mapping	VAR
Object code	Unsigned32

Value description

Sub-index	00h, manufacturer status register
Explanation	Status of motor and configuration
Access	read-only
PDO mapping	–
Value range	–
Default value	–
Storable	–

The list of operating messages and error messages available in bit coded form is given in the chapter on diagnostics and error correction.

1003h *Pre-defined error field*

The object stores the last error messages to be displayed as an EMCY message.

- The entry in sub-index 00h contains the number of error messages stored.
- The current error message is filed in sub-index 01h and older messages are moved to higher sub-index entries.
- Writing a '0‘ to sub-index 00h resets the error list.

5960 0100 011, -000, 04.01

Object description

Index	1003h
Object name	predefined error field
PDO mapping	ARRAY
Object code	Unsigned32

Value description

Sub-index	00h, number of errors
Explanation	Number of error entries
Access	
PDO mapping	–
Value range	0 (rw), 1...5 (ro)
Default value	–
Storable	–

Sub-index	01h, standard error field 1
Explanation	Last error which occurred
Access	read-only
PDO mapping	–
Value range	–
Default value	–
Storable	–

Sub-index	02h, standard error field 2
Explanation	Older error message (2nd place in history)
Access	read-only
PDO mapping	–
Value range	–
Default value	–
Storable	–

Sub-index	03h, standard error field 3
Explanation	Older error message (3rd place in history)
Access	read-only
PDO mapping	–
Value range	–
Default value	–
Storable	–

Sub-index	04h, standard error field 4
Explanation	Older error message (4th place in history)
Access	read-only
PDO mapping	–
Value range	–
Default value	–
Storable	–

Sub-index	05h, standard error field 5
Explanation	Older error message (5th place in history)
Access	read-only
PDO mapping	–
Value range	–
Default value	–
Storable	–

The positioning device stores the last five error messages. The list is deleted when the drive is switched off.

5960 0100 011, -000, 04.01

Bit coding, sub-index 01h..05h

Bit	Explanation
31-16	Additional error information, not assigned in the positioning device.
15-0	Error code, see error code table in the chapter on diagnostics and error correction, and the *error code* object (603Fh).

1004h *Number of PDOs supported*

The object gives the number of PDOs supported by the device.

Object description

Index	1004h
Object name	number of PDOs supported
PDO mapping	ARRAY
Object code	Unsigned32

Value description

Sub-index	00h, number of PDOs supported
Explanation	Number of PDOs supported
Access	read-only
PDO mapping	–
Value range	–
Default value	0x00020002
Storable	–

Sub-index	01h, number of synchonous PDOs
Explanation	Number of synchronous R_PDOs, T_PDOs
Access	read-only
PDO mapping	–
Value range	–
Default value	0x00010001
Storable	–

Sub-index	02h, number of asynchronous PDOs
Explanation	Number of asynchronous R_PDOs, T_PDOs
Access	read-only
PDO mapping	–
Value range	–
Default value	0x00010001
Storable	–

Bit coding, sub-index 00h

Bit	Access	Value	Explanation
31-16	ro	0002_h	Number of R_PDOs
15-0	ro	0002_h	Number of T_PDOs

Bit coding, sub-index 01h

Bit	Access	Value	Explanation
31-16	ro	0001_h	Number of synchronous R_PDOs
15-0	ro	0001_h	Number of synchronous T_PDOs

Bit coding, sub-index 02h

Bit	Access	Value	Explanation
31-16	ro	0001_h	Number of asynchronous R_PDOs
15-0	ro	0001_h	Number of asynchronous T_PDOs

5960 0100 011, -000, 04.01

1005h *COB-Id SYNC message*

The object gives the COB-Id of the SYNC object, and defines whether a device sends or receives SYNC messages.

The positioning device can only receive SYNC messages.

Object description

Index	1005h
Object name	COB-ID SYNC-message
PDO mapping	VAR
Object code	Unsigned32

Value description

Sub-index	00h, COB-ID SYNC-message
Explanation	Identifier for synchronisation object
Access	
PDO mapping	–
Value range	–
Default value	80000080h
Storable	–

Bit coding, sub-index 00h

Bit	Access	Value	Explanation
31	ro	1_b	Device can receive SYNC messages (SYNC-consumer)
30	ro	0_b	Device cannot send SYNC messages (SYNC-producer)
29	ro	0_b	0: 11-bit identifier (CAN 2.0A) 1: 29-bit identifier (CAN 2.0B)
28-11	ro	0000_h	only relevant if bit 29=1, not used by the positioning device.
10-7	rw	0001_b	Function code, bits 10..7 of the COB-Id
6-0	ro	$7F_h$	Node address, bits 6..0 of the COB-Id

A network device must transmit SYNC objects for the purpose of synchronization.

Saving the COB-Id

The COB-Id can be changed in the „pre-operational“ NMT status. The changed setting is then saved in the positioning device by means of the *Store parameters* object (1010h), sub-index=02h.

1008h *Manufacturer device name*

This object identifies the manufacturer's device name.

Object description

Index	1008h
Object name	manufacturer device name
PDO mapping	VAR
Object code	Visible String

5960 0100 011, -000, 04.01

Value description

Sub-index	00h, manufacturer device name
Explanation	Device name
Access	read-only
PDO mapping	–
Value range	–
Default value	IclA D065
Storable	–

The following objects contain further information about the device:
- Objects 2000h to 2003h: hardware and software versions
- Objects 6404h, 6406h, 6410h: motor data

100Bh *Node ID*

This object supplies the node address (node ID) of the device.

Object description

Index	100Bh
Object name	node-ID
PDO mapping	VAR
Object code	Unsigned32

Value description

Sub-index	00h, node-ID
Explanation	7 bit node address
Access	read-only
PDO mapping	–
Value range	–
Default value	127
Storable	–

The node number of the positioning drive is set and saved via LMT services. If the node number is changed, the COB IDs of the following objects automatically change:

Objects	**COB ID stored in the object**
EMCY	1014h
R_PDO1/R_PDO2	1400h/1401h
T_PDO1/T_PDO2	1800h/1801h
SDO	1200h
NMT service for monitoring connections	100Eh

100Ch *Guard time*

This object gives the time for monitoring the connections (node guarding) of an NMT slave.

Object description

Index	100Ch
Object name	guard time
PDO mapping	VAR
Object code	Unsigned16

5960 0100 011, -000, 04.01

Value description

Sub-index	00h, guard time
Explanation	Time span for node guarding [ms]
Access	
PDO mapping	–
Value range	0…65535
Default value	0
Storable	✔

The node guarding time for an NMT master is defined as the „guard time“ multiplied by the „life time“ factor, the *Life time factor* object (100Dh).

The time can be changed in the „pre-operational“ NMT status. The changed setting is then saved in the positioning device by means of the *Store parameters* object (1010h), sub-index=02h.

100Dh *Life time factor*

This object gives the factor which together with the „guard time“ makes up the interval for monitoring the connections of an NMT master. Within this time span the NMT slave expects to receive a node guarding enquiry from the NMT master.

life time = guard time * life time factor

The value „0“ deactivates monitoring of the NMT master.

Object description

Index	100Dh
Object name	life time factor
PDO mapping	VAR
Object code	Unsigned8

Value description

Sub-index	00h, life time factor
Explanation	Repeat factor for node guarding protocol.
Access	
PDO mapping	–
Value range	0…255
Default value	0
Storable	–

If the NMT master does not carry out a node guarding operation during the „life time“ time span, any current positioning operation is halted. The positioning device then changes to manual mode.

The time factor can be changed in the „pre-operational“ NMT status. The changed setting is then saved in the positioning device by means of the *Store parameters* object (1010h), sub-index=02h.

The „guard time“ is set by means of the *Guard time* object (100Ch).

5960 0100 011, -000, 04.01

100Eh *Node Guarding Identifier*

This object gives the COB ID for the NMT monitoring service „Node-guarding“.

Object description

Index	100Eh
Object name	node guarding identifier
PDO mapping	VAR
Object code	Unsigned32

Value description

Sub-index	00h, node guarding identifier
Explanation	Identifier for node guarding protocol
Access	
PDO mapping	–
Value range	–
Default value	0x0000077F
Storable	–

Bit coding, sub-index 00h

Bit	Access	Value	Explanation
31-30	ro	00_b	reserved
29	ro	0_b	0: 11-bit identifier (CAN 2.0A) 1: 29-bit identifier (CAN 2.0B)
28-11	ro	0000_h	only relevant if bit 29=1, not used by the positioning device.
10-7	rw	1110_b	Function code, bits 10..7 of the COB-Id
6-0	ro	7Fh	Node address, bits 6..0 of the COB-Id

The COB-Id can be changed in the „pre-operational“ NMT status. The changed setting is then saved in the positioning device by means of the *Store parameters* object (1010h), sub-index=02h.

100Fh *Number of SDOs supported*

The object gives the number of server and client SDOs supported by the device.

Object description

Index	100Fh
Object name	number of SDOs supported
PDO mapping	VAR
Object code	Unsigned32

Value description

Sub-index	00h, number of SDOs supported
Explanation	Number of SDOs supported
Access	read-only
PDO mapping	–
Value range	–
Default value	0x00000001
Storable	–

5960 0100 011, -000, 04.01

Bit coding, sub-index 00h

Bit	Access	Value	Explanation
31-16	ro	0000_h	Number of client SDOs supported
15-0	ro	0001_h	Number of server SDOs supported

1010h *Store parameters*

This allows changable parameters to be stored in the device's permanent memory. The data saved are available again when the device is switched on. Read access to the object gives information about the device's memory functions.

Parameter ranges for read and write access are specified by selecting the relevant sub-index:

- Sub-index 01h: All storable parameters
- Sub-index 02h: Communications parameters
- Sub-index 03h: Application parameters

Object description

Index	1010h
Object name	store parameter
PDO mapping	ARRAY
Object code	Unsigned32

Value description

Sub-index	00h, number of elements
Explanation	Entries per object, number
Access	read-only
PDO mapping	–
Value range	–
Default value	3
Storable	–

Sub-index	01h, save all parameters
Explanation	Write all parameters to configuration memory
Access	
PDO mapping	–
Value range	1 (rd), 0x65766173 (wr)
Default value	0x00000001
Storable	–

Sub-index	02h, save communication parameters
Explanation	Write communications parameters to configuration memory
Access	
PDO mapping	–
Value range	1 (rd), 0x65766173 (wr)
Default value	0x00000001
Storable	–

5960 0100 011, -000, 04.01

Sub-index	03h, save application parameters
Explanation	Write applications parameters to configuration memory
Access	
PDO mapping	–
Value range	1 (rd), 0x65766173 (wr)
Default value	0x00000001
Storable	–

Bit coding, sub-index 01h..03h

To prevent data being saved unintentionally, the „save“ signature must be sent when saving data:

Bits	**3124**	**2316**	**15..8**	**7..0**
ASCII	e	v	a	s
hex value	65h	76h	61h	73h

The data in the following objects are saved for the selected range of parameters.

Communications parameters

Index	**Sub-index**	**Object name**
100Ch	00h	Guard time
100Dh	00h	Life time factor

Application parameters, device profile

Index	**Sub-index**	**Object name**
607Dh	01h:02	Software position limit
607Eh	00h	Polarity

Application parameters, manufacturer-specific

Index	**Sub-index**	**Object name**
2004h	02h..0Ah	User profile velocity, profile 2..10
2005h	02h..0Ah	User profile acceleration, profile 2..10
2006h	02h..0Ah	User profile deceleration, profile 2..10
2008h	01h, 02h	Software position drive limit D_0, D_1
2009h	01h, 02h	Software position safety limit S_0, S_1
2010h	01h..11h	Control parameter set
2011h	01h..06h	Manual mode settings

During read access, the positioning device returns the value „1“ for every sub-index.
1: device saves parameters or parameter ranges on request

5960 0100 011, -000, 04.01

1011h *Restore default parameters*

This object resets all parameters or certain parameter ranges to given settings during write access. Read access to the object checks whether the given values can be set.

Parameter ranges for read and write access are specified by selecting the relevant sub-index:

- Sub-index 01h: All storable parameters
- Sub-index 02h: Communications parameters
- Sub-index 03h: Application parameters

Object description

Index	1011h
Object name	restore default parameters
PDO mapping	ARRAY
Object code	Unsigned32

Value description

Sub-index	00h, number of elements
Explanation	Entries per object, number
Access	read-only
PDO mapping	–
Value range	–
Default value	3
Storable	–

Sub-index	01h, restore all default parameters
Explanation	Load all parameter values from the configuration memory
Access	
PDO mapping	–
Value range	1(rd), 0x64616F6C (wr)
Default value	0x00000001
Storable	–

Sub-index	02h, restore communication default parameters
Explanation	Load communications parameters from the configuration memory
Access	
PDO mapping	–
Value range	1(rd), 0x64616F6C (wr)
Default value	0x00000001
Storable	–

Sub-index	03h, restore application default parameters
Explanation	Load applications parameters from the configuration memory
Access	
PDO mapping	–
Value range	1(rd), 0x64616F6C (wr)
Default value	0x00000001
Storable	–

5960 0100 011, -000, 04.01

Bit coding, sub-index 01h..03h

To prevent unintentional write operations, the „load“ signature must be sent when requesting write access to an object:

Bits	**3124**	**2316**	**15..8**	**7..0**
ASCII	d	a	o	l
hex value	64h	61h	6Fh	6Ch

When the communications parameters are restored, the default values are loaded, and when the application parameters are restored, the application values stored in the positioning device. The object values which can be loaded are described in the section on the *Store parameters* object (1010h).

Restoring control parameters

In addition the factory settings can be loaded for the control parameters which are stored in the *Control parameter set* object (2010h), by setting the configuration mode - *Modes of operation* object (6060h) by means of sub-index 00h=80h - before the application parameters are loaded.

To ensure that the factory settings data are available when the device is switched on again, the NMT service „Reset node“ must be carried out after the data are loaded. Later the data can be saved using the *Store parameters* object (1010h), sub-index 03h.

During read access, the positioning device returns for every sub-index the value „1“.
1: device can restore default parameters.

1014h *COB-ID emergency message*

This object gives the COB-ID of the Emergency object „EMCY“.

Object description

Index	1014h
Object name	COB-ID emergency message
PDO mapping	VAR
Object code	Unsigned32

Value description

Sub-index	00h, COB-ID emergency message
Explanation	Identifier for Emergency object
Access	
PDO mapping	–
Value range	–
Default value	0x400000FF
Storable	–

5960 0100 011, -000, 04.01

Bit coding, sub-index 00h

Bit	Access	Value	Explanation
31	ro	0_b	0: device will not evaluate EMCY messages 1: device will evaluate EMCY messages
30	ro	1_b	0: device will not generate EMCY messages 1: device will send EMCY messages
29	ro	0_b	0: 11-bit identifier (CAN 2.0A) 1: 29-bit identifier (CAN 2.0B)
28-11	ro	0000_h	only relevant if bit 29=1, not used by the positioning device.
10-7	rw	0001_b	Function code, bits 10..-7 of the COB-Id
6-0	ro	7Fh	Node address, bits 6..-0 of the COB-Id

Saving the COB-Id

The COB-Id can be changed in the „pre-operational“ NMT status. The changed setting is then saved in the positioning device by means of the *Store parameters* object (1010h), sub-index=02h.

1018h *Identity Object*

This object contains general drive data.

Object description

Index	1018Fh
Object name	identity object
PDO mapping	RECORD
Object code	Visible String

Value description

Sub-index	00h, number of entries
Explanation	Number of entries
Access	read-only
PDO mapping	–
Value range	1 .. 4
Default value	–
Storable	–

Sub-index	01h, vendor ID
Explanation	Manufacturer name
Access	read-only
PDO mapping	–
Value range	Unsigned32
Default value	–
Storable	–

Sub-index	02h, product code
Explanation	Product name
Access	read-only
PDO mapping	–
Value range	Unsigned32
Default value	–
Storable	–

5960 0100 011, -000, 04.01

Sub-index	03h, revision number
Explanation	Serial number of the drive
Access	read-only
PDO mapping	–
Value range	1 .. 4
Default value	–
Storable	–

Sub-index	04h, serial number
Explanation	Serial number of the drive
Access	read-only
PDO mapping	–
Value range	Unsigned32
Default value	–
Storable	–

1200h *Server SDO parameters*

The object gives the number of server SDOs used and the COB-IDs of R_SDO and T_SDO of the first server SDO.

The server is the network device to whose object directory the index and sub-index of an SDO message refer. The SDO data exchange is initiated and controlled via the client SDO of a network client.

Object description

Index	1200h
Object name	server SDO parameter
PDO mapping	RECORD
Object code	SDO Parameter

Value description

Sub-index	00h, number of elements
Explanation	Entries per object, number
Access	read-only
PDO mapping	–
Value range	–
Default value	2
Storable	–

Sub-index	01h, COB-ID Client -> Server (rx)
Explanation	Identifier for receive SDO R_SDO
Access	read-only
PDO mapping	–
Value range	–
Default value	0x0000067F
Storable	–

Sub-index	02h, COB-ID Server -> Client (tx)
Explanation	Identifier for transmit SDO T_SDO
Access	read-only
PDO mapping	–
Value range	–
Default value	0x000005FF
Storable	–

5960 0100 011, -000, 04.01

Bit coding, sub-index 01h

Bit	Access	Value	Explanation
31	ro	0_b	0: SDO is valid 1: SDO is invalid
30	ro	0_b	reserved
29	ro	0_b	0: 11-bit identifier (CAN 2.0A) 1: 29-bit identifier (CAN 2.0B)
28-11	ro	0000_h	only relevant if bit 29=1, not used by the positioning device.
10-7	ro	1100_b	Function code, bits 10..-7 of the COB-Id
6-0	ro	$7F_h$	Node address, bits 6..-0 of the COB-Id

Bit coding, sub-index 02h

Bits 11 to 31 are assigned as for sub-index 01h. The function code of the T_SDOs (bits 7-10) is set to 1011_b.

1400h *1st receive PDO parameter*

The object stores the COB-ID and transmission type of the first receive PDO R_PDO1.

Object description

Index	1400h
Object name	1st receive PDO parameter
PDO mapping	RECORD
Object code	PDO Communication Parameter

Value description

Sub-index	00h, number of elements
Explanation	Entries per object, number
Access	read-only
PDO mapping	–
Value range	–
Default value	2
Storable	–

Sub-index	01h, COB-ID used by PDO
Explanation	Identifier for R_PDO1
Access	
PDO mapping	–
Value range	–
Default value	0x0000027F
Storable	–

Sub-index	02h, transmission type
Explanation	Transmission type for R_PDO1
Access	
PDO mapping	–
Value range	0 ... 240, 252...255
Default value	255
Storable	–

5960 0100 011, -000, 04.01

Bit assignment, sub-index 01h

Bit	Access	Value	Explanation
31	rw	0_b	0: PDO is valid 1: PDO is invalid
30	ro	0_b	0: RTR is possible (see below) 1: RTR not permitted
29	ro	0_b	0: 11-bit identifier (CAN 2.0A) 1: 29-bit identifier (CAN 2.0B)
28-11	ro	0000_h	only relevant if bit 29=1, not used by the positioning device.
10-7	rw	0100_b	Function code, bits 10..-7 of the COB-Id
6-0	ro	7Fh	Node address, bits 6..-0 of the COB-Id

Bit 31 An R_PDO can only be used if bit 31=„0“.

Bit 30: RTR-Bit If a network device supports R_PDOs with RTR (remote transmission request), it can request a PDO from a PDO producer by means of RTR = „0“ in accordance with the producer-consumer relationship.

The positioning device cannot request PDOs, but it can respond to a request for a PDO, see RTR bit for T_PDO1 settings (1800h).

Bit coding, sub-index 02h The control for evaluating R_PDO data is defined via sub-index 02h. The values 241..251 are reserved.

Type of transmission	cyclical	acyclical	synchronous	asynchronous	RTR controlled
0	–	X	X	–	–
1-240	X	–	X	–	–
252	–	–	X	–	X
253	–	–	–	X	X
254	–	–	–	X	–
255	–	–	–	X	–

If an R_PDO is transmitted synchronously (transmission type=0..252), the device evaluates the data received according to the SYNC object.

- When transmission is acyclical (transmission type=0), evaluation is linked to the SYNC object, while transmission of the PDO is not. When a PDO message is received, it is evaluated with the following SYNC.

  A value between 1 and 240 gives the number of SYNC cycles after which a received PDO is evaluated.

The values 252 to 254 are relevant for updating - but not for transmitting - T_PDOs.

- 252: Updating of transmission data with reception of the next SYNC
- 253: Updating of transmission data with the reception of a request from a PDO consumer.
- 254: Updating of data event-controlled, the trigger event defined by the manufacturer

5960 0100 011, -000, 04.01

R_PDOs with the value 255 are updated immediately upon receipt of the PDO. The data which are transmitted in the PDO in accordance with the definition of the device profile DSP 402, constitute the trigger event.

Settings

R_PDO1 is processed asynchronously as soon as the data are received.

The byte assignment of the R_PDO1 is defined via the PDO mapping with the *1st Receive PDO mapping* object (1600h). The following assignment is permanently set for the positioning device:

- bytes 0..1: *controlword* (6040h)

The “transmission type” can be modified independent of the NMT status.

Setting the COB ID

The COB ID can only be modified in NMT status “pre-operational”.

1401h *2nd receive PDO parameter*

The object stores the COB ID and transmission type of the second receive PDO, R_PDO2.

Object description

Index	1401h
Object name	2nd receive PDO parameter
PDO mapping	RECORD
Object code	PDO Communication Parameter

Value description

Sub-index	00h, number of elements
Explanation	Entries per object, number
Access	read-only
PDO mapping	–
Value range	–
Default value	2
Storable	–

Sub-index	01h, COB-ID used by PDO
Explanation	Identifier for R_PDO2
Access	
PDO mapping	–
Value range	–
Default value	0x0000037F
Storable	✔

Sub-index	02h, transmission type
Explanation	Transmission type for R_PDO2
Access	
PDO mapping	–
Value range	0 ... 240, 252...255
Default value	0
Storable	✔

The signficance of the bit states is described by means of the *1st receive PDO-parameters* object (1400h). The function code, bits 7-10, is 0110_b.

5960 0100 011, -000, 04.01

Settings

R_PDO2 is processed synchronously after the data are received.

The byte assignment of the R_PDO2 is defined via the PDO mapping with the *2nd Receive PDO mapping* object (1601h). The following assignment is permanently set for the positioning device:

- bytes 0..1: *controlword* (6040h)
- bytes 2..5: target position of the movement command *Target position* (607Ah)

The “transmission type” can be modified independent of the NMT status.

Setting the COB ID

The COB ID can only be modified in NMT status “pre-operational”.

1600h *1st receive PDO mapping*

This object states which objects are shown in the R_PDO1 and transmitted with the PDO. When sub-index 00h of the object is read, the number of objects shown is supplied.

Object description

Index	1600h
Object name	1st receive PDO mapping
PDO mapping	RECORD
Object code	PDO Mapping

Value description

Sub-index	00h, number of elements
Explanation	Entries per object, number
Access	read-only
PDO mapping	–
Value range	–
Default value	1
Storable	–

Sub-index	01h, 1st mapped object
Explanation	First object for mapping in R_PDO1
Access	read-only
PDO mapping	–
Value range	–
Default value	0x60400010
Storable	–

Bit coding from sub-index 01h..h

Every sub-index entry from sub-index 01h specifies the object and the its byte length. The object is identified via the index and sub-index which refer to the device's object directory.

Bit	Explanation
3116	Index
15..8	Sub-index
7..0	object length in bytes

Settings

The positioning device supports static PDO mapping. This means that the PDO assignment cannot be changed. The following assignment is permanently set for R_PDO1:

- Sub-index 01h: PDO mapping of the control word object, *controlword* (6040h).

5960 0100 011, -000, 04.01

1601h *2nd receive PDO mapping*

This object states which objects are shown in the R_PDO2 and transmitted with the PDO. When sub-index 00h of the object is read, the number of objects shown is supplied.

Object description

Index	1601h
Object name	2nd receive PDO mapping
PDO mapping	RECORD
Object code	PDO Mapping

Value description

Sub-index	00h, number of elements
Explanation	Entries per object, number
Access	read-only
PDO mapping	–
Value range	–
Default value	2
Storable	–

Sub-index	01h, 1st mapped object
Explanation	First object for mapping in R_PDO2
Access	read-only
PDO mapping	–
Value range	–
Default value	0x60400010
Storable	–

Sub-index	02h, 2nd mapped object
Explanation	Second object for mapping in R_PDO2
Access	read-only
PDO mapping	–
Value range	–
Default value	0x607A0020
Storable	–

The signficance of the bit states is described by means of the *1st receive PDO-mapping* object (1600h).

Settings

The positioning device supports static PDO mapping. This means that the PDO assignment cannot be changed. The following assignment is permanently set for R_PDO2:

- Sub-index 01h: PDO mapping of the control word object, *control-word* (6040h)
- Sub-index 02h: Target position des movement command, *target position* object (607Ah).

1800h *1st transmit PDO parameter*

The object stores the COB-ID and transmission type of the first transmit PDO, T_PD01. R_PDO1.

Object description

Index	1800h
Object name	1st transmit PDO parameter
PDO mapping	RECORD
Object code	PDO Communication Parameter

5960 0100 011, -000, 04.01

Value description

Sub-index	00h, number of elements
Explanation	Entries per object, number
Access	read-only
PDO mapping	–
Value range	–
Default value	2
Storable	–

Sub-index	01h, COB-ID used by PDO
Explanation	Identifier for T_PDO1
Access	
PDO mapping	–
Value range	–
Default value	0x000001FF
Storable	–

Sub-index	02h, transmission type
Explanation	Transmission type for T_PDO1
Access	
PDO mapping	–
Value range	0 ... 240, 252...255
Default value	255
Storable	–

The signficance of the bit states is described by means of the *1st receive PDO-parameters* object (1400h).

Settings

T_PDO1 is communicated on an event-driven basis after every change to the PDO data.

The byte assignment of the R_PDO1 is defined via the PDO mapping with the *1st transmit PDO mapping* object (1A00h). The following assignment is permanently set for the positioning device:

- bytes 0..1: *statusword* (6041h)

Setting the COB ID

The COB ID can only be modified in NMT status “pre-operational”.

1801h *2nd transmit PDO parameter*

The object stores the COB-ID and transmission type of the second transmit PDO, T_PD02.

Object description

Index	1801h
Object name	2nd transmit PDO parameter
PDO mapping	RECORD
Object code	PDO Communication Parameter

Value description

Sub-index	00h, number of elements
Explanation	Entries per object, number
Access	read-only
PDO mapping	–
Value range	–
Default value	2
Storable	–

5960 0100 011, -000, 04.01

Sub-index	01h, COB-ID used by PDO
Explanation	Identifier for T_PDO2
Access	
PDO mapping	–
Value range	–
Default value	0x000002FF
Storable	–

Sub-index	02h, transmission type
Explanation	Transmission type for T_PDO2
Access	
PDO mapping	–
Value range	0...255
Default value	0
Storable	–

The signficance of the bit states is described by means of the *1st receive PDO-parameters* object (1400h).

Settings T_PDO2 must be requested from a PDO consumer.

The byte assignment of the R_PDO1 is defined via the PDO mapping with the *2nd transmit PDO mapping* object (1A01h). The following assignment is permanently set for the positioning device:

- bytes 0..1: *statusword* (6041h)
- Bytes 2..5: *actual position* (6064h).

Setting the COB ID The COB ID can only be modified in NMT status “pre-operational”.

1A00h *1st transmit PDO mapping*

This object states which objects are shown in the T_PDO1 and transmitted with the PDO. When sub-index 00h of the object is read, the number of objects shown is supplied.

Object description

Index	1A00h
Object name	1st transmit PDO mapping
PDO mapping	RECORD
Object code	PDO Mapping

Value description

Sub-index	00h, number of elements
Explanation	Entries per object, number
Access	read-only
PDO mapping	–
Value range	–
Default value	1
Storable	–

Sub-index	01h, 1st mapped object
Explanation	First object for mapping in T_PDO1
Access	read-only
PDO mapping	–
Value range	–
Default value	0x60410010
Storable	–

5960 0100 011, -000, 04.01

The signficance of the bit states is described by means of the *1st receive PDO-mapping* object (1600h).

Settings The positioning device supports static PDO mapping. This means that the PDO assignment cannot be changed. The following assignment is permanently set for T_PDO1:

- Sub-index 1: PDO mapping of the status word *statusword* object (6041h)

1A01h *2nd transmit PDO mapping*

This object states which objects are shown in the T_PDO2 and transmitted with the PDO. When sub-index 0 of the object is read, the number of objects shown is supplied.

Object description

Index	1A01h
Object name	2nd transmit PDO mapping
PDO mapping	RECORD
Object code	PDO Mapping

Value description

Sub-index	00h, number of elements
Explanation	Entries per object, number
Access	read-only
PDO mapping	–
Value range	–
Default value	2
Storable	–

Sub-index	01h, 1st mapped object
Explanation	First object for mapping in T_PDO2
Access	read-only
PDO mapping	–
Value range	–
Default value	0x60410010
Storable	–

Sub-index	02h, 2nd mapped object
Explanation	Second object for mapping in T_PDO2
Access	read-only
PDO mapping	–
Value range	–
Default value	0x60640020
Storable	–

The signficance of the bit states is described by means of the *1st receive PDO-mapping* object (1600h).

Settings The positioning device supports static PDO mapping. This means that the PDO assignment cannot be changed. The following assignment is permanently set for T_PDO2:

- Sub-index 1: PDO mapping of the status word *statusword* object (6041h)
- Sub-index 2: PDO mapping of the actual position, object *actual position* (6064h).

5960 0100 011, -000, 04.01

2000h *Drive hardware version*

The object gives information about the electronic specifications of the device. Service staff use the data to locate the cause of a malfunction quickly.

Object description

Index	2000h
Object name	drive hardware version
PDO mapping	VAR
Object code	Visible String

Value description

Sub-index	00h, drive hardware version
Explanation	Hardware version of electronic controls
Access	read-only
PDO mapping	–
Value range	–
Default value	–
Storable	–

2001h *Motor hardware version*

The object gives information about the motor mechanics.

Object description

Index	2001h
Object name	motor hardware version
PDO mapping	VAR
Object code	Visible String

Value description

Sub-index	00h, motor hardware version
Explanation	Hardware version of motor mechanics
Access	read-only
PDO mapping	–
Value range	–
Default value	–
Storable	–

2002h *Motor operating software*

The object gives information about the software in the motor operation program.

Object description

Index	2002h
Object name	motor operating software
PDO mapping	VAR
Object code	Visible String

Value description

Sub-index	00h, motor operating software
Explanation	Software version of motor operating program
Access	read-only
PDO mapping	–
Value range	–
Default value	774.00 x.xxx
Storable	–

5960 0100 011, -000, 04.01

2003h *Application software*

The object gives information about the applications software.

Object description

Index	2003h
Object name	application software
PDO mapping	VAR
Object code	Visible String

Value description

Sub-index	00h, application software
Explanation	Software version of application program
Access	read-only
PDO mapping	–
Value range	–
Default value	775.00 x.xxx
Storable	–

2004h *User profile velocity*

The object defines the speed values for the ten pre-configured movement profiles. A movement profile is selected by means of the *User profile number* object (2007h). The value of this movement profile is transferred to the *Profile velocity* object (6081h) as the profile speed during manufacturer-specific positioning mode.

The object value may not exceed the maximum profile speed which is defined via the *max profile velocity* object (607Fh).

Object description

Index	2004h
Object name	user profile velocity
PDO mapping	ARRAY
Object code	Unsigned32

Value description

Sub-index	00h, number of elements
Explanation	Entries per object, number
Access	read-only
PDO mapping	–
Value range	–
Default value	10
Storable	–

Sub-index	01h, user profile velocity 1
Explanation	Movement profile 1: speed in [Inc/s]
Access	read-only
PDO mapping	–
Value range	–
Default value	S18: 782; S38: 750; S54: 770; S115: 750
Storable	–

5960 0100 011, -000, 04.01

Sub-index	02h, user profile velocity 2
Explanation	Movement profile 2: speed in [Inc/s]
Access	
PDO mapping	–
Value range	2...S18: ..999; S38: ..999; S54: ..914; S115: ..987
Default value	S18: 782; S38: 750; S54: 770; S115: 750
Storable	✔

Sub-index	03h, user profile velocity 3
Explanation	Movement profile 3: speed in [Inc/s]
Access	
PDO mapping	–
Value range	2...S18: ..999; S38: ..999; S54: ..914; S115: ..987
Default value	S18: 782; S38: 750; S54: 770; S115: 750
Storable	✔

Sub-index	04h, user profile velocity 4
Explanation	Movement profile 4: speed in [Inc/s]
Access	
PDO mapping	–
Value range	2...S18: ..999; S38: ..999; S54: ..914; S115: ..987
Default value	S18: 782; S38: 750; S54: 770; S115: 750
Storable	✔

Sub-index	05h, user profile velocity 5
Explanation	Movement profile 5: speed in [Inc/s]
Access	
PDO mapping	–
Value range	2...S18: ..999; S38: ..999; S54: ..914; S115: ..987
Default value	S18: 782; S38: 750; S54: 770; S115: 750
Storable	✔

Sub-index	06h, user profile velocity 6
Explanation	Movement profile 6: speed in [Inc/s]
Access	
PDO mapping	–
Value range	2...S18: ..999; S38: ..999; S54: ..914; S115: ..987
Default value	S18: 782; S38: 750; S54: 770; S115: 750
Storable	✔

Sub-index	07h, user profile velocity 7
Explanation	Movement profile 7: speed in [Inc/s]
Access	
PDO mapping	–
Value range	2...S18: ..999; S38: ..999; S54: ..914; S115: ..987
Default value	S18: 782; S38: 750; S54: 770; S115: 750
Storable	✔

Sub-index	08h, user profile velocity 8
Explanation	Movement profile 8: speed in [Inc/s]
Access	
PDO mapping	–
Value range	2...S18: ..999; S38: ..999; S54: ..914; S115: ..987
Default value	S18: 782; S38: 750; S54: 770; S115: 750
Storable	✔

5960 0100 011, -000, 04.01

Sub-index	09h, user profile velocity 9
Explanation	Movement profile 9: speed in [Inc/s]
Access	
PDO mapping	–
Value range	2...S18: ..999; S38: ..999; S54: ..914; S115: ..987
Default value	S18: 782; S38: 750; S54: 770; S115: 750
Storable	✔

Sub-index	0Ah, user profile velocity 10
Explanation	Movement profile 10: speed in [Inc/s]
Access	
PDO mapping	–
Value range	2...S18: ..999; S38: ..999; S54: ..914; S115: ..987
Default value	S18: 782; S38: 750; S54: 770; S115: 750
Storable	✔

Sub-index 01h contains movement profile values which are set for rated load and speed, and which cannot be changed. The other nine movement profile values can be set in configuration mode. Configuration mode is selected by means of the *Modes of operation* object (6060h), sub-index 00h=80h.

Saving values

Object settings can be saved in the positioning device using the *Store parameters* object (1010h), sub-index=03h.

2005h *User profile acceleration*

The object defines the acceleration values for ten pre-configured movement profiles. A movement profile is selected by means of the *User profile number* object (2007h). The value of this movement profile is transferred to the *Profile acceleration* object (6083h) as the profile acceleration during manufacturer-specific positioning mode.

The object value may not exceed the maximum profile acceleration defined via the *max profile acceleration* object (60C5h) .

Object description

Index	2005h
Object name	user profile acceleration
PDO mapping	ARRAY
Object code	Unsigned32

Value description

Sub-index	00h, number of elements
Explanation	Entries per object, number
Access	read-only
PDO mapping	–
Value range	–
Default value	10
Storable	–

Sub-index	01h, user profile acceleration 1
Explanation	Movement profile 1: acceleration in [Inc/s^2]
Access	read-only
PDO mapping	–
Value range	–
Default value	5000
Storable	–

5960 0100 011, -000, 04.01

Sub-index	02h, user profile acceleration 2
Explanation	Movement profile 2: acceleration in [Inc/s^2]
Access	
PDO mapping	–
Value range	1...5000
Default value	5000
Storable	✔

Sub-index	03h, user profile acceleration 3
Explanation	Movement profile 3: acceleration in [Inc/s^2]
Access	
PDO mapping	–
Value range	1...5000
Default value	5000
Storable	✔

Sub-index	04h, user profile acceleration 4
Explanation	Movement profile 4: acceleration in [Inc/s^2]
Access	
PDO mapping	–
Value range	1...5000
Default value	5000
Storable	✔

Sub-index	05h, user profile acceleration 5
Explanation	Movement profile 5: acceleration in [Inc/s^2]
Access	
PDO mapping	–
Value range	1...5000
Default value	5000
Storable	✔

Sub-index	06h, user profile acceleration 6
Explanation	Movement profile 6: acceleration in [Inc/s^2]
Access	
PDO mapping	–
Value range	1...5000
Default value	5000
Storable	✔

Sub-index	07h, user profile acceleration 7
Explanation	Movement profile 7: acceleration in [Inc/s^2]
Access	
PDO mapping	–
Value range	1...5000
Default value	5000
Storable	✔

Sub-index	08h, user profile acceleration 8
Explanation	Movement profile 8: acceleration in [Inc/s^2]
Access	
PDO mapping	–
Value range	1...5000
Default value	5000
Storable	✔

5960 0100 011, -000, 04.01

Sub-index	09h, user profile acceleration 9
Explanation	Movement profile 9: acceleration in [Inc/s^2]
Access	
PDO mapping	–
Value range	1...5000
Default value	5000
Storable	✔

Sub-index	0Ah, user profile acceleration 10
Explanation	Movement profile 10: acceleration in [Inc/s^2]
Access	
PDO mapping	–
Value range	1...5000
Default value	5000
Storable	✔

Sub-index 01h contains movement profile values which are set for rated load and speed, and which cannot be changed. The other nine movement profile values can be set in configuration mode. Configuration mode is selected by means of the *Modes of operation* object (6060h), sub-index 00h=80h.

Saving values

Object settings can be saved in the positioning device using the *Store parameters* object (1010h), sub-index=03h.

2006h *User profile deceleration*

The object defines the deceleration values for ten pre-configured movement profiles. A movement profile is selected by means of the *User profile number* object (2007h). The value of this movement profile is transferred to the *Profile deceleration* object (6084h) as the profile deceleration during manufacturer-specific positioning mode.

The object value may not exceed the maximum profile deceleration defined via the *max profile deceleration* object (60C5h) .

Object description

Index	2006h
Object name	user profile deceleration
PDO mapping	ARRAY
Object code	Unsigned32

Value description

Sub-index	00h, number of elements
Explanation	Entries per object, number
Access	read-only
PDO mapping	–
Value range	–
Default value	10
Storable	–

Sub-index	01h, user profile deceleration 1
Explanation	Movement profile 1: deceleration in [Inc/s^2]
Access	read-only
PDO mapping	–
Value range	–
Default value	2000
Storable	✔

5960 0100 011, -000, 04.01

Sub-index	02h, user profile deceleration 2
Explanation	Movement profile 2: deceleration in [Inc/s^2]
Access	
PDO mapping	–
Value range	1...2000
Default value	2000
Storable	✔

Sub-index	03h, user profile deceleration 3
Explanation	Movement profile 3: deceleration in [Inc/s^2]
Access	
PDO mapping	–
Value range	1...2000
Default value	2000
Storable	✔

Sub-index	04h, user profile deceleration 4
Explanation	Movement profile 4: deceleration in [Inc/s^2]
Access	
PDO mapping	–
Value range	1...2000
Default value	2000
Storable	✔

Sub-index	05h, user profile deceleration 5
Explanation	Movement profile 5: deceleration in [Inc/s^2]
Access	
PDO mapping	–
Value range	1...2000
Default value	2000
Storable	✔

Sub-index	06h, user profile deceleration 6
Explanation	Movement profile 6: deceleration in [Inc/s^2]
Access	
PDO mapping	–
Value range	1...2000
Default value	2000
Storable	✔

Sub-index	07h, user profile deceleration 7
Explanation	Movement profile 7: deceleration in [Inc/s^2]
Access	
PDO mapping	–
Value range	1...2000
Default value	2000
Storable	✔

Sub-index	08h, user profile deceleration 8
Explanation	Movement profile 8: deceleration in [Inc/s^2]
Access	
PDO mapping	–
Value range	1...2000
Default value	2000
Storable	✔

5960 0100 011, -000, 04.01

Sub-index	09h, user profile deceleration 9
Explanation	Movement profile 9: deceleration in [Inc/s^2]
Access	
PDO mapping	–
Value range	1...2000
Default value	2000
Storable	✔

Sub-index	0Ah, user profile deceleration 10
Explanation	Movement profile 10: deceleration in [Inc/s^2]
Access	
PDO mapping	–
Value range	1...2000
Default value	2000
Storable	✔

Sub-index 01h contains movement profile values which are set for rated load and speed, and which cannot be changed. The other nine movement profile values can be set in configuration mode. Configuration mode is selected by means of the *Modes of operation* object (6060h), sub-index 00h=80h.

Saving values

Object settings can be saved in the positioning device using the *Store parameters* object (1010h), sub-index=03h.

2007h *User profile number*

This object is used to select one of ten pre-configured movement profiles in manufacturer-specific positioning mode. Profile values are set by means of the objects *User profile velocity* (2004h), *User profile acceleration* (2005h) and *User profile deceleration* (2006h).

Object description

Index	2007h
Object name	user profile number
PDO mapping	VAR
Object code	Unsigned8

Value description

Sub-index	00h, user profile number
Explanation	Selection of movement profile 1 to 10
Access	
PDO mapping	–
Value range	1...10
Default value	1
Storable	–

2008h *Software position safety limit*

This object stores the limit switch values for the safety area S_0 to S_1. If the positioning device crosses area limit S_0 or S1, it must be moved back to the movement area or work area manually.

Object description

Index	2008h
Object name	software position safety limit
PDO mapping	ARRAY
Object code	Integer32

5960 0100 011, -000, 04.01

Value description

Sub-index	00h, number of elements
Explanation	Entries per object, number
Access	read-only
PDO mapping	–
Value range	–
Default value	2
Storable	–

Sub-index	01h, min position limit
Explanation	Lower software limit switch S_0 [Inc]
Access	
PDO mapping	–
Value range	-2147483648...2147483647
Default value	-2147483648
Storable	✔

Sub-index	02h, max position limit
Explanation	Upper software limit switch S_1 [Inc]
Access	
PDO mapping	–
Value range	-2147483648...2147483647
Default value	2147483647
Storable	✔

Saving values

Object values are set in referencing mode, and can be saved in the positioning device using the *Store parameters* object (1010h), sub-index=03h.

Three limit switch pairs must be set for a valid referencing operation:
The work area via the *Software position limit* object (607Dh)
the movement area via the *Software position drive limit* object (2009h)
the safety area via the *Software position safety limit* object (2008h).

2009h *Software position drive limit*

This object stores the limit switch values for the travel area D_0 to D_1. The values can be set in referencing mode. If the positioning device crosses area limit D_0 or D_1, it can be moved back into the work area.

Object description

Index	2009h
Object name	software position drive limit
PDO mapping	ARRAY
Object code	Integer32

Value description

Sub-index	00h, number of elements
Explanation	Entries per object, number
Access	read-only
PDO mapping	–
Value range	–
Default value	2
Storable	–

5960 0100 011, -000, 04.01

Sub-index	01h, min position limit
Explanation	Lower software limit switch D0 [Inc]
Access	
PDO mapping	–
Value range	-2147483648...2147483647
Default value	-2147483648
Storable	✔

Sub-index	02h, max position limit
Explanation	Upper software limit switch D_1 [Inc]
Access	
PDO mapping	–
Value range	-2147483648...2147483647
Default value	2147483647
Storable	✔

Saving values

Object values are set in referencing mode, and can be saved in the positioning device using the *Store parameters* object (1010h), sub-index=03h.

Three limit switch pairs must be set for a valid referencing operation:
The work area via the *Software position limit* object (607Dh)
the movement area via the *Software position drive limit* object (2009h)
the safety area via the *Software position safety limit* object (2008h).

200Bh *Position assignment value*

This object uses the position value which is transferred to initialize the positioning device's position memory. Referencing mode must be set and activated, and the position value transferred must lie within the work area W_0 to W1. the work area is defined by means of the *Software position limit* object (607Dh).

Object description

Index	200Bh
Object name	position assignment value
PDO mapping	VAR
Object code	Integer32

Value description

Sub-index	00h, position assignment value
Explanation	Position memory initialization
Access	write-only
PDO mapping	–
Value range	-2147483647...2147483646
Default value	0
Storable	–

200Ch *Overcurrent average value*

This object shows the average motor current which is measured during overload or overcurrent for a defined period of time. It is shown as a per mil value with reference to the motor's rated current: value = measured value / (1000 * rated current of motor).

The time period is set using the *Control parameter set* object (2010h), sub-index 0fh. The motor's rated current can be established via the *motor rated current* object (6075h).

5960 0100 011, -000, 04.01

Object description

Index	200Ch
Object name	overcurrent average value
PDO mapping	VAR
Object code	Integer16

Value description

Sub-index	00h, overcurrent average value
Explanation	Average overload current [‰], (rated motor current [mA] / 1000 mA)
Access	read-only
PDO mapping	–
Value range	–
Default value	–
Storable	–

200Dh *Temperature actual value*

This object gives the current temperature of the power amplifier.

Object description

Index	200Dh
Object name	temperature actual value
PDO mapping	VAR
Object code	Integer16

Value description

Sub-index	00h, temperature actual value
Explanation	Temperature of power amplifier in [°C]
Access	read-only
PDO mapping	–
Value range	–
Default value	–
Storable	–

Power amplifier temperature thresholds can be established via the *drive data* object (6510h), sub-index 02h and 03h.

200Eh *Drive log-book*

This object stores the positioning device's.log-book data. The log-book is continuously updated during operating time and saved to the configuration memory when the positioning device is switched off. The following data are saved:

- length of time drive was switched on
- total operating time of drive
- number of positioning operations

Object description

Index	200Eh
Object name	drive log book
PDO mapping	RECORD
Object code	User Defined

5960 0100 011, -000, 04.01

Value description

Sub-index	00h, number of elements
Explanation	Entries per object, number
Access	read-only
PDO mapping	–
Value range	–
Default value	5
Storable	–

Sub-index	01h, total power-on time
Explanation	Power-on time of drive [s]
Access	read-only
PDO mapping	–
Value range	–
Default value	–
Storable	–

Sub-index	02h, total turn-on time
Explanation	Turn-on time of drive [s]
Access	read-only
PDO mapping	–
Value range	–
Default value	–
Storable	–

Sub-index	03h, number of positionings
Explanation	Number of positioning operations
Access	read-only
PDO mapping	–
Value range	–
Default value	–
Storable	–

Sub-index	04h, number of reference loss
Explanation	Number of reference losses
Access	read-only
PDO mapping	–
Value range	–
Default value	0
Storable	–

Sub-index	05h, number of log book loss
Explanation	Number of log-book losses
Access	read-only
PDO mapping	–
Value range	–
Default value	0
Storable	–

5960 0100 011, -000, 04.01

200Fh *Drive serial number*

This object gives the serial number of the positioning device which is printed on the device's name plate.

Object description

Index	200Fh
Object name	drive serial-number
PDO mapping	VAR
Object code	Visible String

Value description

Sub-index	00h, drive serial-number
Explanation	Serial number of drive
Access	read-only
PDO mapping	–
Value range	–
Default value	–
Storable	–

The serial number is part of the LMT address and is required for the assignment of a node ID by the LMT service. It is shown in the LMT address as a 14-digit, seven byte character sequence. Serial numbers with fewer than 14 characters are completed by using leading zeros.

Examplel: 10-digit serial number „31 23 45 67 89“ is transmitted as the byte sequence „00h 00h 00h 31h 23h 45h 67h 89h“.

2010h *Control parameter set*

This object contains all the adjustable control parameters in the motor controls.

Object description

Index	2010h
Object name	control parameter set
PDO mapping	RECORD
Object code	User Defined

Value description

Sub-index	00h, number of elements
Explanation	Entries per object, number
Access	read-only
PDO mapping	–
Value range	–
Default value	17
Storable	–

Sub-index	01h, k p pos
Explanation	P-amplification position controller [‰]
Access	
PDO mapping	–
Value range	100...65535
Default value	1000
Storable	✔

5960 0100 011, -000, 04.01

Sub-index	02h, k p rpm
Explanation	P-amplification speed controller [‰]
Access	
PDO mapping	–
Value range	100...15999
Default value	1000
Storable	✔

Sub-index	03h, k p cmd
Explanation	Reference variable speed controller [‰]
Access	
PDO mapping	–
Value range	0...15999
Default value	0
Storable	✔

Sub-index	04h, r i lag
Explanation	Reset time for PI controller [ms]
Access	
PDO mapping	–
Value range	1...4100
Default value	34
Storable	✔

Sub-index	05h, position window
Explanation	Tolerance window for target position [ms]
Access	
PDO mapping	–
Value range	0...65535
Default value	1
Storable	✔

Sub-index	06h, rpm window
Explanation	Speed deviations when setpoint reached [r.p.m.]
Access	
PDO mapping	–
Value range	0...5000
Default value	500
Storable	✔

Sub-index	07h, rpm deviation events
Explanation	Event counter limit for speed monitoring
Access	
PDO mapping	–
Value range	10...32000
Default value	400
Storable	✔

Sub-index	08h, rpm start timeout
Explanation	Speed monitoring for motor start-up [ms]
Access	
PDO mapping	–
Value range	50...8000
Default value	650
Storable	✔

5960 0100 011, -000, 04.01

Sub-index	09h, rpm stop
Explanation	Speed monitoring when motor stop to 0 [r.p.m.]
Access	
PDO mapping	–
Value range	100...1000
Default value	300
Storable	✔

Sub-index	0Ah, lp1 time const
Explanation	Time constant speed low-pass filter [ms]
Access	
PDO mapping	–
Value range	1...255
Default value	30
Storable	✔

Sub-index	0Bh, holding torque time
Explanation	Duration of holding torque [ms]
Access	
PDO mapping	–
Value range	0...230
Default value	100
Storable	✔

Sub-index	0Ch, max steady current
Explanation	Rated current for application [mA]
Access	
PDO mapping	–
Value range	0...6000
Default value	S18, S38: 6000; S54: 5800; S115: 3300
Storable	✔

Sub-index	0Dh, starting current factor
Explanation	Start-up factor [%] (maximum overload current)
Access	
PDO mapping	–
Value range	100...150
Default value	150
Storable	✔

Sub-index	0Eh, deviation delay
Explanation	Time delay in measurement of speed variations [ms]
Access	
PDO mapping	–
Value range	0...255
Default value	100
Storable	✔

Sub-index	0Fh, current deviation events
Explanation	Event counter limit for overload and overcurrent
Access	
PDO mapping	–
Value range	50...30000
Default value	500
Storable	✔

5960 0100 011, -000, 04.01

Sub-index	10h, current window
Explanation	Overcurrent window [mA]
Access	
PDO mapping	–
Value range	0...400
Default value	160
Storable	✔

Sub-index	11h, block deceleration
Explanation	Block movement deceleration in [1/(min×ms)]
Access	
PDO mapping	–
Value range	200...5000
Default value	1000
Storable	✔

Control parameters The positioning device is operated via a PI control system with reference variable input. The control parameters can be optimized via sub-index 01h to 04h in case the device is tending to oscillate.

Sub-index 01h The P amplification changes the ramp time of the deceleration ramp. Low amplification reduces the ramp time.

Sub-index 02h The P amplification of the speed control influences the proportional share of the speed control loop. Higher values make total control more dynamic.

Sub-index 03h The reference variable correction value makes the speed control loop more dynamic by adding in a variable which is proportional to the set value. The reference variable correction value is only of significance in very dynamic situations and normally remains set to zero.

Sub-index 04h The reset time changes the integral proportion of the speed controller. Higher values mean more dynamic performance but at the same time an increased tendency for the system to oscillate.

Sub-index 05h The tolerance window for the target position describes a symmetrical area in which the set position is considered to have been reached, and in which bit 10 in *statusword* (6041h) is set to „position_reached“. The position controller attempts to arrive at the exact set position irrespective of the size of the tolerance window. The system only checks to see whether the position reached is within the positioning tolerance window when the motor is at a standstill.

Sub-index 06h The speed window describes the symmetrical area on either side of the set speed in its steady state in which the set sped is considered to have been reached.

Sub-index 07h The counter is used to monitor deviations in speed once the motor has reached its steady state.

The error counter is incremented at speeds outside the speed window, adjustable via sub-index 06h, and decremented at speeds inside the window. Counting takes place per control cycle or system cycle. If the error counter reaches the count limit which can be set via sub-index 07h, bit 18 „rpm_fail“ of the status register is set.

Sub-index 08h If the motor does not start within the speed monitoring time window, bit 17 „start_fail“ of the status register is set.

Sub-index 09h If the speed of the motor falls below the "stop motor" speed monitoring value, it is stopped by switching in holding torque.

5960 0100 011, -000, 04.01

Sub-index 0Ah Speed measurement signals are cleaned up using a first level filter. Higher values for the low-pass filter time constant have a stronger smoothing and delaying effect on speed values. The higher the values, the greater the delay between the actual speed and the speed which the controller processes. If the value is changed, the time constant can have a negative effect on the positioning accuracy.

Sub-index 0Bh The motor's resting torque does not stop the motor in an exactly defined position. A defined motor stop can be achieved by adding in holding torque. The length of the holding torque is set via the holding torque duration parameter. The amount of holding torque cannot be adjusted.

Sub-index 0Ch The drive torque is set via the operating current. To prevent the system from mechanical damage, the operating current can be used to set a maximum torque. The rated current is pre-set as the threshold for the maximum operating current. If the motor current measured in a steady state exceeds the set operating current for a defined time period, the positioning device signals overload via bit 21, „overcurrent", of the status word.

During operation, the average overload current over a period of time is registered and stored.

Sub-index 0Dh A maximum overload current can be set in order to limit motor torque. The overload current is the operating current set for a movement of the drive (sub-index 0Ch) multiplied by the peak current factor (sub-index 0Dh). The maximum overload current is limited by the hardware current limitation of the electronic components. Overload current settings apply to the start-up phase and to steady state operation. If the motor current measured exceeds the overload current during a drive movement in steady state for a defined period of time, the positioning device signals that the overcurrent limit has been reached via bit 19, „current_limit",.

During operation, the overload current over a period of time is registered and stored cyclically. The time interval for measuring the overcurrent can be changed via sub-index 10h, „ Stromfenster".

Sub-index 0Eh The functions for monitoring deviations in speed and overcurrent are activated when the positioning device is changed from acceleration to constant speed. To prevent pick-up events from falsifying the results of the monitoring functions, the starting point for activating the monitoring functions can be delayed:

starting point for monitoring = time to end of acceleration + delay (sub-index 0Eh)

Sub-index 0Fh The counter is used for monitoring operating current in a steady state.

If the motor current measured exceeds the pre-set operating current, adjustable via sub-index 0Ch, an error counter is incremented and below the operating current threshold it is decremented. Counting takes place per control cycle or system cycle. If the error counter reaches the count limit which can be set via sub-index 0Fh, bit 19 „current_limit" of the status register is set. If the current value reaches the current limit, bit 21, „overcurrent" in the status register is set.

Sub-index 10h The overcurrent window corrects the display sequence for current limitation and overcurrent which can occur as a result of the different system resolutions of the hardware current limitation system and the motor current measuring loop. Bit 19, „current_limit" must first show overcurrent or overload, then bit 21, „overcurrent" the fact that the current limit has been reached.

The overcurrent window can be used to reduce the operating current threshold to the point where overcurrent is displayed at the same time as or just before the current limit is reached.

5960 0100 011, -000, 04.01

Sub-index 11h

The block movement delay stops the drive as soon as it hits an obstacle. The delay is set for a fixed time window of ten milliseconds.

If the reduction in speed within a time window (10ms) exceeds the set delay value with simultaneous overcurrent, bit 20 „block_fail“ in the status word is set and the drive is stopped immediately.

The value for block movement delay should not be set too low in order to prevent premature triggering. The control unit works internally with unfiltered speed values in order to be able to recognize a drop in speed without any delay. Incorrect measurements can result especially in the start-up phase and at low currents.

Saving values

Control parameters can be set in configuration mode (object 6060h, sub-index 00h=80h) and saved in the positioning device using the *Store parameters* object (1010h), sub-index=03h.

Factory settings can be restored by loading the applications parameters in configuration mode. Details on restoring factory settings are described in the communications object, *Load parmeters* (1011h).

2011h *Manual mode settings*

This object is used to set the movement parameters for manual mode. These include:

- number of motor steps in manual mode
- final speed
- acceleration ramp
- deceleration ramp
- max. current, operating current limit
- inching duration

You will find details on the parameters and on manual mode in the chapter on operating modes in the section on manual mode.

Object description

Index	2011h
Object name	manual mode settings
PDO mapping	RECORD
Object code	User Defined

Value description

Sub-index	00h, number of elements
Explanation	Entries per object, number
Access	read-only
PDO mapping	–
Value range	–
Default value	6
Storable	–

Sub-index	01h, increments
Explanation	Manual mode, number of motor increments
Access	
PDO mapping	–
Value range	0, 2...65535
Default value	2
Storable	✔

5960 0100 011, -000, 04.01

Sub-index	02h, velocity
Explanation	Manual mode, speed [Inc/s]
Access	
PDO mapping	–
Value range	0...S18: ..999; S38: ..999; S54: ..914; S115: ..987
Default value	200
Storable	✔

Sub-index	03h, acceleration
Explanation	Manual mode, acceleration [Inc/s^2]
Access	
PDO mapping	–
Value range	0...5000
Default value	500
Storable	✔

Sub-index	04h, deceleration
Explanation	Manual mode, deceleration [Inc/s^2]
Access	
PDO mapping	–
Value range	0...2000
Default value	2000
Storable	✔

Sub-index	05h, max current
Explanation	Rated current for manual mode, [‰], (rated motor current[mA] /1000 mA)
Access	
PDO mapping	–
Value range	0...1000
Default value	1000
Storable	✔

Sub-index	06h, release time
Explanation	Manual mode, maximum pulse duration [ms]
Access	
PDO mapping	–
Value range	0...65535
Default value	500
Storable	✔

The maximum current is limited by the object value *Max current*(6073h) and the object values for describing the ramp by the following profile values:

- Terminal speed by the *max profile velocity* object (607Fh)
- the acceleration ramp by the *max profile acceleration* object (60C5h)
- the deceleration ramp by the *max profile deceleration* object (60C6h) .

Parameters for manual mode can be set in configuration mode (object 6060h, sub-index 00h=80h) and stored in the positioning device using the *Store parameters* object (1010h), sub-index=03h.

5960 0100 011, -000, 04.01

603Fh *Error code*

This object gives the error code of the last error. The value corresponds to the lower 16 bits in the *predefined error field* object (1003h).

Object description

Index	603Fh
Object name	error code
PDO mapping	VAR
Object code	Integer16

Value description

Sub-index	00h, error code
Explanation	Last error to have occurred
Access	read-only
PDO mapping	–
Value range	–
Default value	–
Storable	–

6040h *Controlword*

The object represents the controlword of the device. This *controlword* is used to realize a number of control jobs:

- Transition between the various operate states. For information on possible states and transitions refer to the index entry “default profile ”. Relevant for a status transition are the bits 0 to 3 and bit 7.
- Starting and interrupting specific operating mode functions, for example, starting a run job via bit 4. Bits 4 to 6 are used for specific operating mode settings. You can find details under the keywords “Operating mode, starting”, “Operating modes, monitoring” and in the description of the respective operate states in the Chapters “Manual Mode” and “Positioning Mode”.
- Stopping the positioning drive during a run operation. “Halt” bit 8 is used to stop the drive. For details refer to the keywords “Operating mode, starting” and “operating mode, monitoring”.

The image of the controlword can be found in the first two bytes of the T_PDOs.

Object description

Index	6040h
Object name	controlword
PDO mapping	VAR
Object code	Integer16

Value description

Sub-index	00h, controlword
Explanation	Control word for changing operating status
Access	
PDO mapping	R_PDO
Value range	–
Default value	–
Storable	–

5960 0100 011, -000, 04.01

Bit coding, sub-index 00h Change of mode

Bit	designation	explanation
11..15		not used
9, 10		reserved
..8	Stop	stop motor
7	Reset fault	reset fault
4..6	–	depends on mode, see below
3	Enable operation	carry out mode
2	Quick Stop	brake using Quick Stop ramp
1	Disable voltage	switch off voltage
0	Switch on	switch on ready for operation

The significance of bits 4 and 6 depends on the operating mode currently set. Bit 5 is not assigned.

operating mode	Bit 4	Bit 6
Positioning mode (Profile position mode)	new_setpoint, 0-> 1: start positioning	positioning 0: absolute 1: relative
homing (Homing mode)	0->1: start homing	–
manual mode (manufacturer-specific)	–	–

You can find information on the bit states, procedures for operating mode transitions and the default profile under “CANopen default profile”, “Changing and monitoring operating states”, “Setting and monitoring operating states”, “Manual mode” and “Positioning mode”.

6041h *Statusword*

The object describes the actual operating state of the device. Using statusword, carry out the following monitoring functions:

- Checking the operate state of the positioning controls. Bits 0 - 3, 5 and 6 are here relevant.
- Bit 4 indicates whether the final stage is ready to carry out a run job.
- Bits 7 to 15 are used to monitor run operation and the status of device specific states. You can find details under the keywords “Operating mode, starting”, “Operating modes, monitoring” and in the description of the respective operate states in the Chapters “Manual Mode” and “Positioning Mode”. The device status monitoring bits are described in the Chapter “Diagnostics and fault elimination”.

The image of the controlword can be found in the first two bytes of the R_PDOs.

5960 0100 011, -000, 04.01

Object description

Index	6041h
Object name	statusword
PDO mapping	VAR
Object code	Unsigned16

Value description

Sub-index	00h, statusword
Explanation	Status word for evaluating operating status
Access	read-only
PDO mapping	T_PDO
Value range	–
Default value	–
Storable	–

Bit coding, sub-index 00h

Operating states:

Bit	Designation	Explanation
15	Out of security area	safety area has been left 0->1: limit switch position S_0 or S_1 crossed
14	Out of drive area	area of travel has been left 0->1: limit switch position D_0 or D_1 crossed
12..13	–	significance dependent on operating mode, see below
11	Internal limit active	work area has been left 0->1: limit switch position W_0 or W_1 crossed
10	Target reached	target has been reached 1->0: new target position communicated 0->1: requested target position reached or motor at standstill after stop request
9	Remote	- 0: manual mode switched on - 1: manual mode not switched on
8	Right out of drive area	only valid if bit 11 = 1 - 0: limit switch position W_1 crossed - 1: limit switch position W_0 crossed
7	Warning	warning
6	Switch on disabled	not ready for operation
5	Quick Stop	Quick Stop active
4	Voltage disabled	voltage switched off
3	Fault	fault has occurred
2	Operation enabled	operating mode activated
1	Switched on	ready for operation
0	Ready to switch on	ready to be switched on

The significance of bits 12 and 13 depends on the operating mode currently set.

5960 0100 011, -000, 04.01

operating mode	Bit 12	Bit 13
Positioning mode	setpoint acknowledge 0: new position can be accepted 1: new target position received	–
homing	Homing attained 0: : homing not yet carried out 1: homing carried out	Homing error 0: no fault 1: fault when homing

You can find information on the bit states, procedures for operating mode transitions and the default profile under “CANopen default profile”, “Changing and monitoring operating states”, “Setting and monitoring operating states”, “Manual mode” and “Positioning mode”.

6060h *Modes of operation*

The object switches to the specified operating mode.

Object description

Index	6060h
Object name	modes of operation
PDO mapping	VAR
Object code	Integer8

Value description

Sub-index	00h, modes of operation
Explanation	Set operating mode
Access	write-only
PDO mapping	–
Value range	-128, -2, -1, 1, 6
Default value	-1
Storable	–

Bit coding, sub-index 00h

Value	Explanation
-128 (80h)	configuration mode (manufacturer-specific)
-2 (FEh)	Positioning mode with pre-set movement profile (manufacturer-specific)
-1 (FFh)	manual mode (manufacturer-specific)
1	Positioning mode (Profile position mode)
6	homing (Homing mode)

The current operating mode is shown by the *Modes of operation display* object (6061h).

The operating mode can be changed as soon as the drive has come to a stop. It is only possible to switch into configuration mode (-128) from the NMT status „pre-operational“.

After the power supply has been switched on and the NMT status changed via „Reset node“, the positioning device switches into manual mode (-1).

5960 0100 011, -000, 04.01

6061h *Modes of operation display*

The object displays the current operating mode.

Object description

Index	6061h
Object name	modes of operation display
PDO mapping	VAR
Object code	Integer8

Value description

Sub-index	00h, modes of operation display
Explanation	Display current operating mode
Access	read-only
PDO mapping	–
Value range	–
Default value	-1
Storable	–

Bit coding, sub-index 00h

Value	Explanation
-128 (80h)	configuration mode (manufacturer-specific)
-2 (FEh)	Positioning mode with pre-set movement profile (manufacturer-specific)
-1 (FFh)	manual mode (manufacturer-specific)
1	Positioning mode (Profile position mode)
6	homing (Homing mode)

The current operating mode can be changed via the *Modes of operation* object (6060h).

6064h *Position actual value*

The object gives the current position in user-defined units. For the positioning device this corresponds to motor increments. The direction of motion is established by multiplying by the sense of rotation factor *polarity* (607Eh).

Object description

Index	6064h
Object name	position actual value
PDO mapping	VAR
Object code	Integer32

Value description

Sub-index	00h, position actual value
Explanation	Current position of drive [Inc]
Access	read-only
PDO mapping	T_PDO
Value range	–
Default value	0
Storable	–

The positioning drive transmits the current position in bytes 2..5 of the T_PDO2.

A target position is communicated by means of the *target position* object (607Ah).

5960 0100 011, -000, 04.01

606Ch *Velocity actual value*

The object gives information about the current speed of the drive.

Object description

Index	606Ch
Object name	velocity actual value
PDO mapping	VAR
Object code	Integer32

Value description

Sub-index	00h, velocity actual value
Explanation	Current speed of drive [Inc/s]
Access	read-only
PDO mapping	–
Value range	–
Default value	–
Storable	–

6073h *Max current*

The object is used to set the maximum operating current for the movement of the drive in positioning mode. The value is limited by the two current values

- rated motor current, object*Motor rated current* (6075h)
- permissible operating current, object *Control parameter set* (2010h), sub-index 0Ch.

The operating current is given as a per mil value with respect to the rated current of the motor (6075h).

Object description

Index	6073h
Object name	max current
PDO mapping	VAR
Object code	Unsigned16

Value description

Sub-index	00h, max current
Explanation	Rated current for bus operation [‰], (rated motor current [mA] /1000 mA)
Access	
PDO mapping	–
Value range	0...1000
Default value	1000
Storable	–

The start-up current and peak current are defined at the same time as the operating current. Start-up current and peak current are defined by means of the *Control parameter set* object (2010h), sub-index 0Dh as a factor of the operating current.

5960 0100 011, -000, 04.01

6075h *Motor rated current*

This object specifies the rated current of the motor being used in milli-amps.

Object description

Index	6075h
Object name	motor rated current
PDO mapping	VAR
Object code	Unsigned32

Value description

Sub-index	00h, motor rated current
Explanation	Rated motor current [mA]
Access	read-only
PDO mapping	–
Value range	–
Default value	S18: 6200; S38: 6200; S54: 5800; S115: 3300
Storable	–

6078h *Current actual value*

The motor current being measured can be read from this object.

The value is given as a per mil value with respect to the rated current of the motor (6075h).

Object description

Index	6078h
Object name	current actual value
PDO mapping	VAR
Object code	Integer16

Value description

Sub-index	00h, current actual value
Explanation	Current motor current [‰], (rated motor current [mA] /1000 mA)
Access	read-only
PDO mapping	–
Value range	–
Default value	–
Storable	–

In the case of overload and overcurrent, the average motor current can be established via the *Overcurrent average value* object (200Ch).

6079h *DC link circuit voltage*

The object gives the current voltage on the DC line.

Object description

Index	6079h
Object name	DC link circuit voltage
PDO mapping	VAR
Object code	Unsigned32

5960 0100 011, -000, 04.01

Value description

Sub-index	00h, DC link circuit voltage
Explanation	Voltage in motor DC line [mV]
Access	read-only
PDO mapping	–
Value range	–
Default value	–
Storable	–

The device monitors for undervoltage and overvoltage and sends off an EMCY message when a fault occurs.

607Ah *Taget position*

The object specifies a new position value which is approached using the set values in the movement profile. The position value is used for positioning mode to DSP 402 and for manufacturer-specific positioning mode.

The target position is determined by multiplying by the sense of rotation factor *polarity* (607Eh) and either as an absolute or relative position depending on bit 4 of the *controlword* (6040h).

The positioning operation is initiated via bit 6 =„1“ in the control word, when positioning mode is activated.

The position value is shown in bytes 2..5 of the R_PDO2.

Object description

Index	607Ah
Object name	target position
PDO mapping	VAR
Object code	Integer32

Value description

Sub-index	00h, target position
Explanation	Target position (setpoint) [Inc]
Access	
PDO mapping	R_PDO
Value range	-2147483647...2147483646
Default value	0
Storable	–

When the start command is given for a positioning operation, the drive checks to see whether the specified target position lies within the referenced work area W_0 to W_1. If this is not the case, the drive remains stationary and signals EMCY fault code FF22h: invalid position value.

The work area is defined by means of the *Software position limit* object (607Dh).

5960 0100 011, -000, 04.01

607Dh *Software position limit*

The object gives the absolute limit switch positions for the work area W_0-W_1. The drive can be operated within these limits. The limit switch positions refer to the point which is defined as the zero point by a homing process.

Object description

Index	607Dh
Object name	software position limit
PDO mapping	ARRAY
Object code	Integer32

Value description

Sub-index	00h, number of elements
Explanation	Entries per object, number
Access	read-only
PDO mapping	–
Value range	–
Default value	2
Storable	–

Sub-index	01h, min position limit
Explanation	Lower software limit switch W_0 [Inc]
Access	
PDO mapping	–
Value range	-2147483647...2147483646
Default value	-2147483647
Storable	✔

Sub-index	02h, max position limit
Explanation	Upper software limit switch W1 [Inc]
Access	
PDO mapping	–
Value range	-2147483647...2147483646
Default value	2147483646
Storable	✔

Limit switch areas

The positioning device recognizes three limit switch areas which must be defined by homing before operation commences. homing is initiated as an operating mode. The type of homing is specified by the *Homing method* object (6098h). The limit switch areas to be defined are:

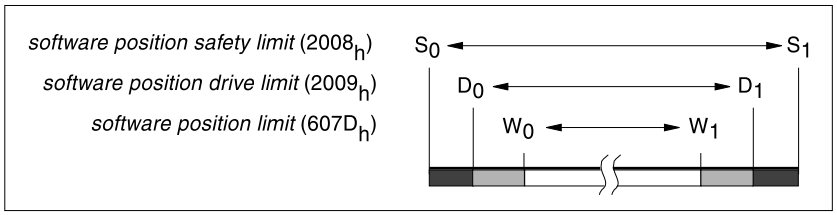

Bild 9.1 Limit switch areas and objects

- Work area W_0-W_1, in which the drive moves to different positions
- Area of travel D_0-D_1, adjustable via the *software position drive limit* object (2009h). If the drive leaves the work area, it can be moved back into the work area by means of a manual movement or positioning command.

5960 0100 011, -000, 04.01

- Safety area S_0-S_1, adjustable via the *software position safety limit* object (2008h). If the current position is outside the area of travel D_0-D_1, the drive must be moved back into the work area mechanically.

Because of the resting torque of the motor, the work area W_0-W_1 must be set to be at least one increment smaller than the area of travel D_0-D_1 on both sides.

Object values are set in homing mode, and can be saved in the positioning device using the *Store parameters* object (1010h), sub-index=03h.

You will find examples of and procedures on homing in the chapter on operating modes.

Acknowledgement information

Three limit switch pairs must be set in total. After each input, the drive checks whether the values are valid. If for example there is a temporary overlap between two limit switch areas after the first input, the drive sends out the SDO error message 0604 0043h. However, when the last limit switch has been set, the drive must give a positive response.

607Eh *Polarity*

This object specifies the sense of rotation factor with which the motor's direction of rotation can be reversed. Changing the sense of rotation is only possible when the positioning device is in homing mode.

Object description

Index	607Eh
Object name	polarity
PDO mapping	VAR
Object code	Unsigned8

Value description

Sub-index	00h, polarity
Explanation	Sense of rotation factor
Access	
PDO mapping	–
Value range	0, 128
Default value	0
Storable	✔

Bit coding sub-index 00h

Bit	Access	Value	Explanation
7	rw	0_b	0: sense of rotation factor = 1 1: sense of rotation factor = -1
6	ro	0_b	not used
5..0	ro	00_h	reserved

The sense of rotation of the positioning device is defined when looking at the driven shaft head on. If the shaft turns clockwise, the position values of the position memory increase to high values with a positive sense of rotation factor.

The sense of rotation factor can be saved in the positioning device using the *Store parameters* object (1010h), sub-index=03h.

5960 0100 011, -000, 04.01

607Fh *Max profile velocity*

This object specifies the maximum permissible speed of the drive.

Object description

Index	607Fh
Object name	max profile velocity
PDO mapping	VAR
Object code	Unsigned32

Value description

Sub-index	00h, max profile velocity
Explanation	Maximum permissible profile speed [Inc/s]
Access	read-only
PDO mapping	–
Value range	–
Default value	S18, S38: 999; S54: 914; S115: 987
Storable	–

6080h *Max motor speed*

The object gives the maximum achievable speed of the motor shaft.

Object description

Index	6080h
Object name	max motor speed
PDO mapping	VAR
Object code	Unsigned16

Value description

Sub-index	00h, max motor speed
Explanation	Maximum motor speed [r.p.m.]
Access	read-only
PDO mapping	–
Value range	–
Default value	5000
Storable	–

6081h *Profile velocity*

The object gives the speed of the drive after the acceleration phase has been completed. The value may not exceed the following thresholds:

- maximum speed, *Max profile velocity* object (607Fh)
- maximum motor speed, *Max motor speed* object (6080h)

Object description

Index	6081h
Object name	profile velocity
PDO mapping	VAR
Object code	Unsigned32

5960 0100 011, -000, 04.01

Value description

Sub-index	00h, profile velocity
Explanation	Speed for a positioning profile [Inc/s]
Access	
PDO mapping	–
Value range	0...S18: ..999; S38: ..999; S54: ..914; S115: ..987
Default value	200
Storable	–

sub-index 00h, default value

As manual mode is the mode into which the drive automatically changes after being switched on (default operating mode), the default value corresponds to the setting for manual movement speed, see *Manuell mode settings* object (2011h), sub-index 02h on page 9-48.

6083h *Profile acceleration*

The object specifies the acceleration after a positioning operation has been commenced. If the value exceeds the threshold for the maximum profile acceleration contained in the *Max acceleration* object (60C5h), the drive limits the value to the maximum value and issues an SDO error message.

Object description

Index	6083h
Object name	profile acceleration
PDO mapping	VAR
Object code	Unsigned32

Value description

Sub-index	00h, profile acceleration
Explanation	Acceleration preset for a positioning profile [Inc/s^2]
Access	
PDO mapping	–
Value range	0...5000
Default value	500
Storable	–

Sub-index 00h, default value

As manual mode is the mode into which the drive automatically switches after being switched on (default operating mode), the default value corresponds to the setting for manual movement acceleration, see the *Manuell mode settings* object (2011h), sub-index 03h on page 9-48.

6084h *Profile deceleration*

The object specifies the deceleration required for reaching the target position. If the value exceeds the threshold for the maximum profile acceleration contained in the *Max deceleration* object (60C5h), the drive limits the value to the maximum value and issues an SDO error message.

Object description

Index	6084h
Object name	profile deceleration
PDO mapping	VAR
Object code	Unsigned32

5960 0100 011, -000, 04.01

Value description

Sub-index	00h, profile deceleration
Explanation	Deceleration preset for a positioning profile [Inc/s^2]
Access	
PDO mapping	–
Value range	0...2000
Default value	–
Storable	–

Sub-index 00h, default value

As manual mode is the mode into which the drive automatically switches after being switched on (default operating mode), the default value corresponds to the setting for manual movement deceleration, see the *Manuell mode settings* object (2011h), sub-index 04h on page 9-48.

608Fh *Position encoder resolution*

The object gives the resolution of the position capture with reference to one revolution of the gear output shaft.

Object description

Index	608Fh
Object name	position encoder resolution
PDO mapping	ARRAY
Object code	Unsigned32

Value description

Sub-index	00h, number of elements
Explanation	Entries per object, number
Access	read-only
PDO mapping	–
Value range	–
Default value	2
Storable	–

Sub-index	01h, encoder increments
Explanation	Position increments
Access	read-only
PDO mapping	–
Value range	–
Default value	S18: 1920; S38: 900; S54: 5880; S115: 4410
Storable	–

Sub-index	02h, motor revolutions
Explanation	Motor revolutions
Access	read-only
PDO mapping	–
Value range	–
Default value	S18: 9; S38: 2; S54: 9; S115: 32
Storable	–

5960 0100 011, -000, 04.01

The resolution is given by the motor data:
resolution = number of Hall sensors × 2 × motor pole pairs × gear ratio

- number of Hall sensors = 3, see *Drive data* object (6510h), sub-index 06h
- number of motor pole pairs = 2, see *Motor data* objedt (6410h), sub-index 0Bh
- Gear ratio depends on the type of drive, see *Gear ratio* object (6091h)

6091h *Gear ratio*

The object specifies the gear ratio of the positioning device. It is calculated from the ratio of motor revolutions to gear shaft revolutions.

Object description

Index	6091h
Object name	gear ratio
PDO mapping	ARRAY
Object code	Unsigned32

Value description

Sub-index	00h, number of elements
Explanation	Entries per object, number
Access	read-only
PDO mapping	–
Value range	–
Default value	2
Storable	–

Sub-index	01h, motor revolutions
Explanation	Motor revolutions
Access	read-only
PDO mapping	–
Value range	–
Default value	S18: 160; S38: 75; S54: 490; S115: 3675
Storable	–

Sub-index	02h, shaft revolutions
Explanation	Revolutions, gear shaft
Access	read-only
PDO mapping	–
Value range	–
Default value	S18: 9; S38: 2; S54: 9; S115: 32
Storable	–

Gear ratio = value (sub-index 02h) / value (sub-index 01h)

6098h *Homing method*

The object specifies the method for carrying out homing operations. The positioning device supports a manufacturer-specific method: the reference point is defined by allocating an absolute position value.

5960 0100 011, -000, 04.01

Object description

Index	6098h
Object name	homing method
PDO mapping	VAR
Object code	Integer8

Value description

Sub-index	00h, homing method
Explanation	Homing method
Access	
PDO mapping	–
Value range	-1
Default value	-1
Storable	–

Values

Value	Explanation
-128..-2	not used
-1	Homing by value allocation
1..34	homing methods 1 to 19 to DSP 402, not supported by the positioning device.
35..127	reserved

The reference point is defined via the *Position assignment value* object (200Bh).

Position values for the reference point refer to the current position of the drive. When the homing operation is initiated, the value of the *Position assignment value* object (200Bh) is adopted as the position value.

Homing is initiated via bit 4 of *Controlword* (6040h). Bits 12 and 13 in *Statusword* (6041h) give information about the operating status of the homing operation. You will find details on homing procedures in the chapter on operating modes.

60C5h *Max acceleration*

The object gives the maximum permissible acceleration.

Object description

Index	60C5h
Object name	max acceleration
PDO mapping	VAR
Object code	Unsigned32

Value description

Sub-index	00h, max acceleration
Explanation	Maximum permissible profile acceleration [Inc/s^2]
Access	read-only
PDO mapping	–
Value range	–
Default value	5000
Storable	–

The ramp value is used in manual and positioning modes.

5960 0100 011, -000, 04.01

60C5h *Max acceleration*

The object gives the maximum permissible deceleration

Object description

Index	60C6h
Object name	max deceleration
PDO mapping	VAR
Object code	Unsigned32

Value description

Sub-index	00h, max deceleration
Explanation	Maximum permissible profile deceleration [Inc/s^2]
Access	read-only
PDO mapping	–
Value range	–
Default value	2000
Storable	–

The ramp value is used in manual and positioning modes.

60FDh *Digital inputs*

The object gives the signal states of the device's digital inputs. The value „1" corresponds to the signal state „High" and signifies: input „triggered".

Object description

Index	60FDh
Object name	digital inputs
PDO mapping	VAR
Object code	Unsigned32

Value description

Sub-index	00h, digital inputs
Explanation	Signal status of digital inputs
Access	read-only
PDO mapping	–
Value range	–
Default value	–
Storable	–

Bit coding, sub-index 00h

The positioning device represents the following signal states:

Bit	Explanation
16, 17	MAN_N, MAN_P-Signal
27	Emergency shutdown signal

6404h *Motor manufacturer*

This object identifies the manufacturer of the motor.

Object description

Index	6404h
Object name	motor manufacturer
PDO mapping	VAR
Object code	Visible String

5960 0100 011, -000, 04.01

Value description

Sub-index	00h, motor manufacturer
Explanation	Motor manufacturer
Access	read-only
PDO mapping	–
Value range	–
Default value	SIG Positec
Storable	–

6406h *Motor calibration date*

The object gives the date of manufacture or the date of the last service.

Object description

Index	6406h
Object name	motor calibration date
PDO mapping	VAR
Object code	Date

Value description

Sub-index	00h, motor calibration date
Explanation	Manufacturing date or last repair date
Access	read-only
PDO mapping	–
Value range	–
Default value	–
Storable	–

Bit coding

Bit	Designation	Explanation	Value range
54..48	year	year	0..99
45..40	month	month	1..12
39..37	day of week	day of week	1: Mon..7:Sun
36..32	day of month	Day	1..31
31	su	summer time	0: no, 1: yes
28..24	hour	hour	0..23
21..16	min	minute	0..59
15..0	ms	millisecond	0..5999

6410h *Motor data*

The object gives the technical data and settings for the connected motor and gear. Settings are dependent on the device type.

Object description

Index	6410h
Object name	motor data
PDO mapping	RECORD
Object code	User Defined

5960 0100 011, -000, 04.01

Value description

Sub-index	00h, number of elements
Explanation	Entries per object, number
Access	read-only
PDO mapping	–
Value range	–
Default value	11
Storable	–

Sub-index	01h, max speed
Explanation	Maximum speed at gear output [r.p.m.]
Access	read-only
PDO mapping	–
Value range	–
Default value	S18: 281; S38: 133; S54: 84; S115: 43
Storable	–

Sub-index	02h, nominal speed
Explanation	Rated speed at gear output [r.p.m.]
Access	read-only
PDO mapping	–
Value range	–
Default value	S18: 220; S38: 100; S54: 65; S115: 30
Storable	–

Sub-index	03h, nominal motor current
Explanation	Rated current of motor [mA]
Access	read-only
PDO mapping	–
Value range	–
Default value	S18, S38: 6000; S54: 5800; S115: 3300
Storable	–

Sub-index	04h, motor torque constant
Explanation	Motor torque constant [1/1000 Ncm/A]
Access	read-only
PDO mapping	–
Value range	–
Default value	4063
Storable	–

Sub-index	05h, gear shaft revolutions
Explanation	Revolutions at gear output
Access	read-only
PDO mapping	–
Value range	–
Default value	S18: 9; S38: 2; S54: 9; S115: 32
Storable	–

Sub-index	06h, gear motor revolutions
Explanation	Revolutions of motor gear shaft
Access	read-only
PDO mapping	–
Value range	–
Default value	S18: 160; S38: 75; S54: 490; S115 3675
Storable	–

5960 0100 011, -000, 04.01

Sub-index	07h, gear stages
Explanation	Number of gear stages
Access	read-only
PDO mapping	–
Value range	–
Default value	S18, S38: 3; S54, S115: 4
Storable	–

Sub-index	08h, gear efficiency
Explanation	Gear efficiency [‰]
Access	read-only
PDO mapping	–
Value range	–
Default value	S18, S38: 779; S54, S115: 716
Storable	–

Sub-index	09h, nominal gear torque
Explanation	Nominal gear torque [1/1000 Nm]
Access	read-only
PDO mapping	–
Value range	–
Default value	S18: 3500; S38: 7300; S54: 10000; S115: 12000
Storable	–

Sub-index	0Ah, gear detent torque
Explanation	Gear self-holding torque [1/1000 Nm]
Access	read-only
PDO mapping	–
Value range	–
Default value	S18: 700; S38: 1400; S54: 2600; S115: 5000
Storable	–

Sub-index	0Bh, pole pairs
Explanation	Number of pole pairs
Access	read-only
PDO mapping	–
Value range	–
Default value	2
Storable	–

Motor and gear in the positioning device cannot be swapped. The data can therefore not be changed.

6502h *Supported drive modes*

The object specifies the operating modes which can be carried out with the device.

Object description

Index	6502h
Object name	supported drive modes
PDO mapping	VAR
Object code	Unsigned32

5960 0100 011, -000, 04.01

Value description

Sub-index	00h, supported drive modes
Explanation	Operating modes supported
Access	read-only
PDO mapping	–
Value range	–
Default value	0x80030021
Storable	–

Bit significance, sub-index 00h

Bit	Access	Value	Explanation
31	ro	1_b	configuration mode
30..18	ro	000_h	not used
17	ro	1_b	Positioning mode with pre-set movement profile, manufacturer-specific
16	ro	1_b	manual mode, manufacturer-specific
15-7	–	00_h	reserved
6	ro	0_b	Interpolated position mode
5	ro	1_b	Homing mode
4	–	0_b	reserved
3	ro	0_b	Profile torque mode
2	ro	0_b	Profile velocity mode
1	ro	0_b	Velocity mode
0	ro	1_b	Profile position mode

Bit value = 1_b: operating mode is supported.
Bit value = 0_b: operating mode not supported.

6504h *Drive manufacturer*

This object specifies the manufacturer of the device.

Object description

Index	6504h
Object name	drive manufacturer
PDO mapping	VAR
Object code	Visible String

Value description

Sub-index	00h, drive manufacturer
Explanation	Drive manufacturer
Access	read-only
PDO mapping	–
Value range	–
Default value	SIG Positec
Storable	–

5960 0100 011, -000, 04.01

6510h *Drive data*

This object gives technical data and settings for the device.

Object description

Index	6510h
Object name	drive data
PDO mapping	RECORD
Object code	User Defined

Value description

Sub-index	00h, number of elements
Explanation	Entries per object, number
Access	read-only
PDO mapping	–
Value range	–
Default value	9
Storable	–

Sub-index	01h, max drive current
Explanation	Maximum current [mA]
Access	read-only
PDO mapping	–
Value range	–
Default value	6000
Storable	–

Sub-index	02h, excess temperature
Explanation	Maximum operating temperature [°C]
Access	read-only
PDO mapping	–
Value range	–
Default value	80
Storable	–

Sub-index	03h, temperature threshold
Explanation	Temperatur threshold [°C]
Access	read-only
PDO mapping	–
Value range	–
Default value	70
Storable	–

Sub-index	04h, min ramp acceleration
Explanation	Minimum acceleration ramp [ms]
Access	read-only
PDO mapping	–
Value range	–
Default value	200
Storable	–

Sub-index	05h, min ramp deceleration
Explanation	Minimum deceleration ramp [ms]
Access	read-only
PDO mapping	–
Value range	–
Default value	500
Storable	–

5960 0100 011, -000, 04.01

Sub-index	06h, hall sensors
Explanation	Number of Hall sensors
Access	read-only
PDO mapping	–
Value range	–
Default value	3
Storable	–

Sub-index	07h, config bits
Explanation	used internally
Access	read-only
PDO mapping	–
Value range	–
Default value	128
Storable	–

Sub-index	08h, enforced commut
Explanation	used internally
Access	read-only
PDO mapping	–
Value range	–
Default value	300
Storable	–

Sub-index	09h, max drive speed
Explanation	Maximum idle speed of motor [r.p.m.]
Access	read-only
PDO mapping	–
Value range	–
Default value	5000
Storable	–

Sub-index 01h The maximum permissible operating current for the control electronics is stored under sub-index 01h.

Sub-index 02h The value gives the maximum temperature up to which the drive is still operable. If the positioning device exceeds the threshold temperature, the electronic controls will switch off the power amplifier and report excess temperature via bit 22 of the status word.

Sub-index 03h The parameter specifies from what temperature the drive is in a critical but not life-threatening condition.

Sub-index 04h The parameter gives the minimum amount of time needed by the drive to accelerate from a standing start to maximum speed.

Sub-index 05h The parameter specifies the minimum time for decelerating the drive from maximum speed to standstill.

Sub-index 06h Three Hall sensors capture position and speed in the positioning device.

Suböindex 09h The parameter gives the maximum idle speed of the motor shaft at the motor - gear connection point. The integrated controller adopts this value as a threshold parameter.

Sub-index 07h and 08h are only implemented for servicing purposes in the factory.

5960 0100 011, -000, 04.01

5960 0100 011, -000, 04.01

Index

5960 0100 011, -000, 04.01

5960 0100 011, -000, 04.01

5960 0100 011, -000, 04.01

5960 0100 011, -000, 04.01

5960 0100 011, -000, 04.01

5960 0100 011, -000, 04.01