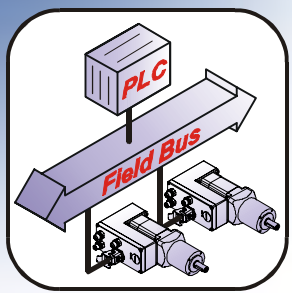
CANopen®

MD-300-CO-CXXX



encoDrive CANopen Handbuch / *Manual*

Dezentrale Stellantriebe / *Decentralized positioning drives*
MD-300-CO-CXXX Series with CANopen interface

- Zusätzliche Sicherheitshinweise
- CANopen-Kommunikation
- Konfiguration / Parametrierung
- Störungsbeseitigung / Diagnose

- *Additional safety instructions*
- *CANopen communication*
- *Configuration / Parameterization*
- *Troubleshooting / Diagnosis*

TR electronic

TR - EMO - BA - DGB - 0010 - 02 / 01/09/2008

User Manual

Decentralized positioning drives MD-300-CO-CXXX

TR-Electronic GmbH
D-78647 Trossingen
Eglishalde 6
Tel.: (0049) 07425/228-0
Fax: (0049) 07425/228-33
E-mail: info@tr-electronic.de
http://www.tr-electronic.de

Copyright protection

Subject to amendments

Any technical changes that serve the purpose of technical progress, reserved.

Document information

Release date/Rev. date:	01/09/2008
Document rev. no.:	TR - EMO - BA - DGB - 0010 - 02
File name:	TR-EMO-BA-DGB-0010-02.DOC
Author:	MÜJ

Font styles

Italic or **bold** font styles are used for the title of a document or are used for highlighting.

`Courier` font displays text, which is visible on the display or screen and software menu selections.

″ < > ″ indicates keys on your computer keyboard (such as <RETURN>).

Trademarks

CANopen® and ***CiA***® are registered community trademarks of CAN in Automation e.V.

CoDeSys is a registered trademark of 3S – Smart Software Solutions GmbH

encoTRive is a registered trademark of TR-Electronic GmbH

Number representations

Prefix "0x" shows that a hexadecimal number is following.
"0x0012" is the hexadecimal representation of the decimal number 18.

Suffix "bin" is the binary representation of a decimal number. Bit 2^0 is on the right.
"0001 0010" is the binary representation of the decimal number 18.

Literature

CiA [2001]: CiA Draft Standard Proposal 306. Electronic Data Sheet Specification for CANopen. Version 1.1. Jun. 2001.

CiA [2002a]: CiA Draft Standard 301. CANopen Application Layer and Communication Profile. Version 4.02. Feb. 2002.

CiA [2002b]: CiA Draft Standard Proposal 402. CANopen Device Profile Drives and Motion Control. Version 2.0. July 2002.

Contents

List of tables

List of figures

List of formulas

Revision index

Revision	Date	Index
First release	02/21/06	00
Modifications – Chapter „Units “, page 172 – English part added	11/27/07	01
Modifications – New error messages: 6345, 6348, 8612 – Parameter modifications, Default values	01/09/08	02

1 General information

This Manual contains the following topics:

- Safety instructions in addition to the basic safety instructions defined in the Assembly Instructions
- Drive profile DSP 402
- CANopen communication
- Configuration / Parameterization
- Troubleshooting and diagnosis options

As the documentation is arranged in a modular structure, this User Manual is supplementary to other documentation, such as customer-specific user manuals, assembly / installation instructions, dimensional drawings, brochures etc..

The User Manual may be included in the customer's specific delivery package or it may be requested separately.

1.1 Target group

This documentation is directed towards

- Commissioning, operating and maintenance personnel, who are instructed to carry out activities on the positioning drive.

The respective qualifications of the personnel are defined in the assembly/commissioning manual in the chapter entitled "Choice and qualifications of personnel; basic obligations".

1.2 Applicability

This Manual applies exclusively for the following decentralized positioning drive types with CANopen interface:

- MD-300-CO-CXXX

The products are labeled with affixed nameplates and are components of a system.

The following documentation therefore also applies:

- operator's operating instructions specific to the system,
- this encoTRive CANopen manual,
- the assembly/installation instructions **TR-EMO-BA-DGB-0015**,
- the customer-specific user manual (optional),
- commissioning instructions for CoDeSys/PLCopen/Function modules/Hand-held unit (optional)

1.3 Abbreviations used / Terminology

A	Ampere
ASCII	American Standard Code for Information Interchange
CAN	Controller Area Network
CCD	Command Code
CiA	CAN in Automation e.V.
COB	Communication Object
COB-ID	COB Identifier
CPU	Central Processing Unit
DIP switch	Dual in-line package switch
DS	Draft Standard
DSP	Draft Standard Proposal
EDS	Electronic Data Sheet
EMCY	Emergency
encoTRive	TR-specific term for the drive
HW	Hardware
inc	Increments
mA	Milliampere
mm	Millimeter
mNm	Millinewton meter
mV	Millivolt
Nm	Newton meter
NMT	Network Management
OV	Object directory
PC	Personal Computer
PDO	Process Data Object
PI	Proportional-Integral
PID	Proportional-Integral-Derivative
PZD	Process data
ro	read only
RPDO	Receive PDO
rph	Revolutions per hour
rpm	Revolutions per minute
rps	Revolutions per second
RTR	Remote Transmission Request
rw	read/write
SDO	Service Data Object
sec	second, Sekunde
STW	Control word
STW.x	Bit x of the control word
PLC	Programmable Logic Controller

SW	Software
SYNC	Synchronization
TPDO	Transmit PDO
V	Volt
wo	write only
ZSW	Status word
ZSW.x	Bit x of the status word

2 Additional safety instructions

2.1 Definition of symbols and instructions

 ***WARNING!***	means that death, serious injury or major damage to property could occur if the required precautions are not met.
 ***CAUTION !***	means that minor injuries or damage to property can occur if the stated precautions are not met.
	indicates important information or features and application tips for the product used.

2.2 Organizational measures

- This Manual must be kept ready to hand at all times at the place of use of the encoTRive.
- Prior to commencing work, the personnel working with the encoTRive must have read and understood
  - the Assembly/Installation Instructions, particularly the chapter "Basic Safety Instructions",
  - and this Manual, particularly the chapter "Additional safety instructions".

This particularly applies for personnel who are only deployed occasionally, e.g. in the parameterization of the encoTRive.

3 DSP 402 drive profile

The linguistic devices for controlling the drive are extensively independent of the manufacturer. For this reason, communication between the drive and the superimposed control system has been standardized in so-called ***drive profiles***.

A ***drive profile*** specifies how an electrical drive is controlled via a field bus. It defines the behavior of the device and the method of accessing the drive data. The following sub areas in particular are controlled:

- Control and status monitoring
- Standardized parameterization
- Changing operating modes

encoTRive supports the DSP 402 profile (CiA [2002b]) as CAN node

The following information is typically exchanged between a master (e.g. control system) and a drive, which assumes a "slave" function:

The drive provides information on its current status (e.g. "*Drive running"*) and possibly additional information such as the current position, current speed etc. In the other direction, the control system assigns positioning orders, for example, (*"Move at speed x to position y"*). Without the DSP 402 profile, each manufacturer would have to specify its own protocols for the transmission of commands and status messages, and there would consequently be many applications performing the same task in different ways.

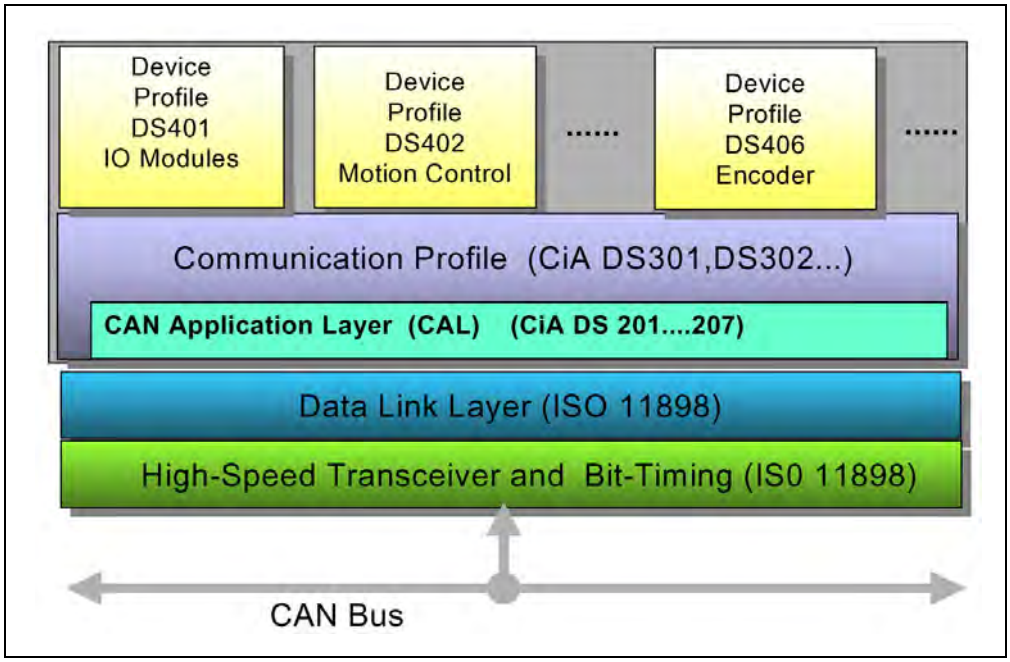


Figure 1: CANopen application profiles

3.1 The object directory

A basic feature of drive profiles is the ***object directory*** (OV). All the information (parameters) relevant to a device is brought together in the object directory. A parameter is identified by its ***parameter number*** (16 Bit). Certain ranges of parameter numbers are occupied or reserved; others are available for so-called manufacturer-specific parameters.

Included in the pre-defined parameters are optional parameters and those, which must be supported by every slave that conforms to the profile ("mandatory parameters").

3.2 CANopen object directory

In CiA, the object directory is seen as an application-neutral concept. Accordingly, the structure of the object directory and access to the objects contained in it are specified in a separate standard, namely ***Draft Standard 301 (DS 301; CANopen , Application Layer and Communication Profile***, CiA [2002a]). Application-specific standards such as DSP 402 or the standard for encoders (DS 406) are added to this. Defined parameters have defined functions within a defined profile. Hexadecimal notation is generally used for parameter numbers. The designation **index** is also normal for the parameter number in the context of CANopen. The parameter range 0x0001-0x1FFF is assigned by DS 301 for parameters that are common to all CANopen devices. The range **0x6000-0x9FFF** is available for the ***standardized device profiles***. In the case of DSP 402, for example, the target position has parameter number 0x607A. In error situations general error causes are defined within DS 301, and each profile specifies further profile-specific error numbers.

Index	Object	
0000_h	not used	Common to all devices
0001_h - $025F_h$	Data type definitions	
0260_h - $0FFF_h$	Reserved	
1000_h - $1FFF_h$	Communication profile area	
2000_h - $5FFF_h$	Manufacturer specific profile area	Device specific
6000_h - $9FFF_h$	Standardized device profile area	
$A000_h$ - $BFFF_h$	Standardized interface profile area	
$C000_h$ - $FFFF_h$	Reserved	

Figure 2: Structure of the object directory

3.3 State machine, status and control word

The state machine is a central element in the drive profile. This is where the operating states and the state transitions are defined. The states that the device goes through after switch-on and how it is transferred into the ***"Ready"*** state are defined so that a positioning movement, for example, can be carried out.
Most state transitions are initiated sequentially by the master transmitting certain commands in the control word in the form of bit patterns.

4 CANopen communication

All signals and information that are required for controlling the electrical drive are transmitted via the field bus.

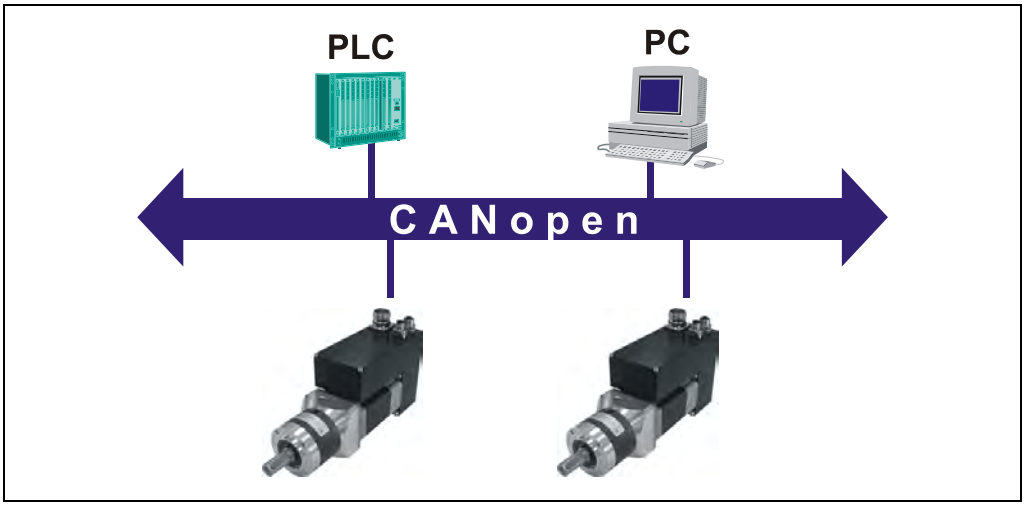


Figure 3: Drives on the field bus

4.1 CANopen communication profile

The CANopen communication profile (documented in CiA DS 301) regulates how devices exchange data with each other. A distinction is made here between real time data and parameter data. CANopen assigns appropriate communication elements to these data types, which are completely different in character.

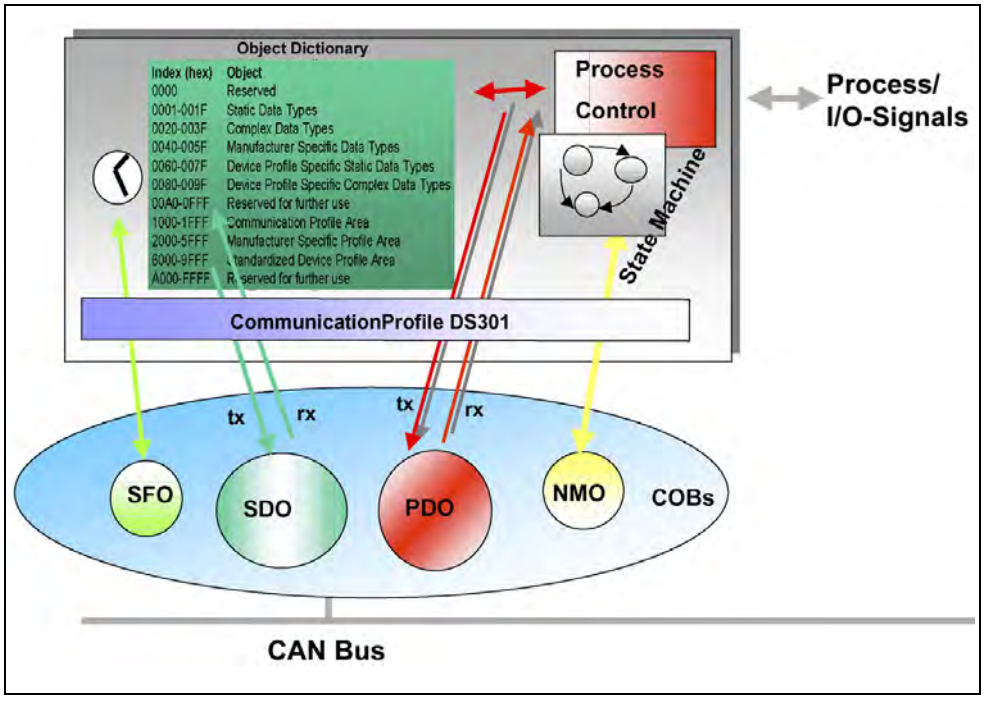


Figure 4: Communication profile

Special **F**unction **O**bject **(SFO)**

- Synchronization (SYNC)
- Emergency (EMCY) protocol

Network **M**anagement **O**bject **(NMO)**

e.g.

- Life / Node-Guarding
- Boot-up, …
- Error Control protocol

4.2 CANopen

4.2.1 General information on CAN and CANopen

The CAN bus is a multi-master system in which each node can send data independently. It is possible that several nodes may send simultaneously. In this case, **the message** with the higher priority takes precedence in CAN. The priority of a message is defined by the so-called ***identifier***, which is transmitted at the beginning of the message. In CAN 2.0A the identifier has a length of 11 bits, and in CAN 2.0B a length of 29 bits.

encoTRive uses the 11-bit identifier in accordance with CAN 2.0A.

The smaller the identifier, the higher the priority of a message.
A node can use several identifiers and thus send different types of messages. However, it must be ensured that the identifiers of different nodes are different.

In order to prevent a high priority message from permanently occupying the bus and thus preventing the transmission of messages with lower priority, an identifier can be assigned a so-called ***Inhibit Time***. This defines the minimum permissible interval between two transmissions of a message with the same identifier.

In order to also be able to address **nodes** in this message-based system, which is particularly necessary when starting a positioning movement for a drive, the identifier is separated into two parts **in CANopen**:

The low value 7 bits contain the ***node address*** (***Node ID***), while the remaining bits contain the so-called function code, which specifies the **purpose** of the message:

Function code				Node ID						
Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Table 1: Identifier with Node ID and function code

In CANopen this identifier, subdivided as above, is referred to as ***COB-ID (Communication Object Identifier)***. As the function code uses the higher value bits of the COB-ID, the function code controls the transmission priorities:
The smaller the function code, the higher the priority.

The Node ID identifies a node in a CANopen network.

In encoTRive, the Node ID and transmission speed are permanently set via DIP switches, defined in the device-specific pin configurations.

CANopen always uses ***Little Endian Format*** for the transmission of numeric values:
The low value byte is stored in the message first of all.

4.2.2 Roles and communication relationships

8 bytes of user data can be transported in a CAN frame. CANopen defines various linguistic devices for the transmission of process data and demand data. So-called ***PDO***s (***Process Data Objects***) are used for process data, and ***SDO***s (***Service Data Objects***) are used for demand data.

Although the CAN bus in itself represents a system with equality of rights (***Multi-Master System***), in CANopen there are different roles, of which some are typically performed by a control, and others typically by a node such as a drive.

Important features of SDO and PDO

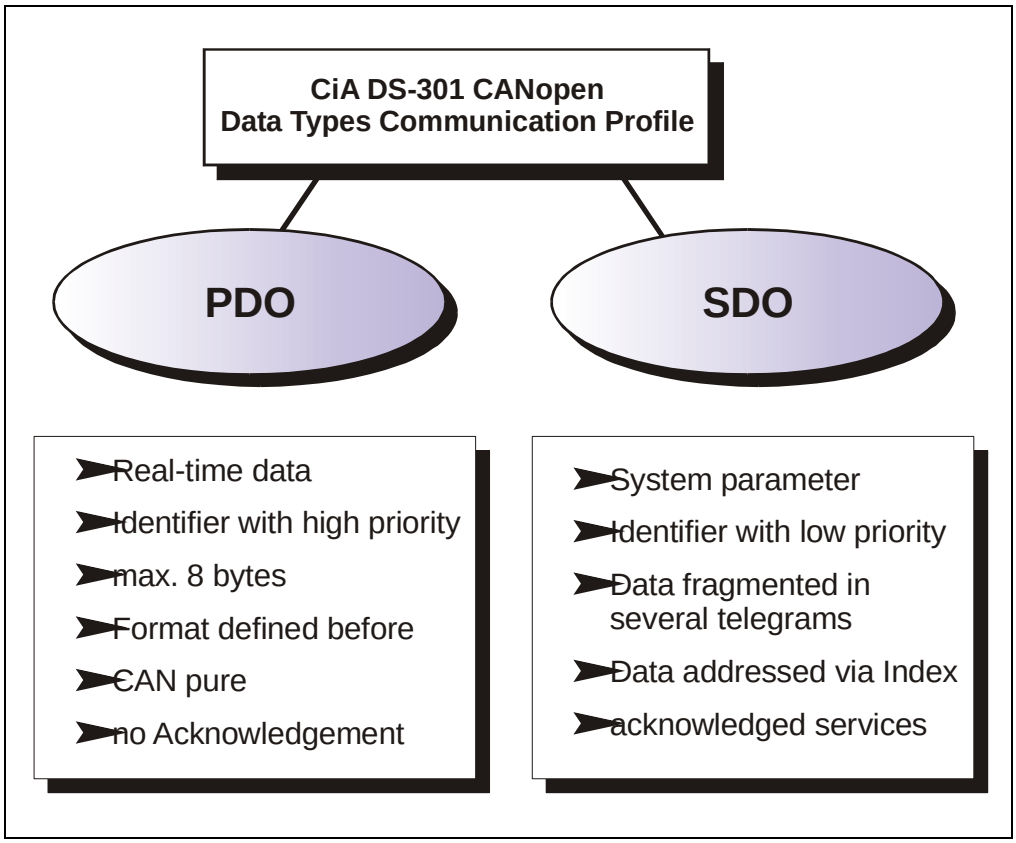


Figure 5: Comparison of PDO/SDO characteristics

4.2.2.1 SDO (Service Data Object)

Objects can be read or written with SDOs. This is a confirmed service. The so-called ***SDO Client*** specifies in its "Request" the parameter, the access type (read/write) and the value if applicable. The so-called ***SDO Server*** executes the write or read access and answers the request with a "Response". In the case of error, an error code provides information on the cause of the error. Transmit SDO and receive SDO are differentiated by their function codes.

encoTRive represents an SDO server and uses the following function codes for SDOs:

Function code	COB-ID	Meaning
`11 (1011 bin)`	`0x580 + Node ID`	**encoTRive → SDO Client**
`12 (1100 bin)`	`0x600 + Node ID`	**SDO Client → encoTRive**

Table 2: COB-IDs for Service Data Object (SDO)

Example:

Node ID encoTRive: 112 (0x70)

- A message with COB-ID 0x5F0 (= 0x580+0x70) corresponds to an SDO from encoTRive to the SDO client.
- A message with COB-ID 0x670 (=0x600+0x70) corresponds to an SDO from the SDO Client to the encoTRive.

4.2.2.1.1 SDO message format

The maximum 8 byte long data range of a CAN message is configured by an SDO as follows:

CCD	Index		Subindex	Data			
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7

Table 3: SDO message

The ***command code*** (***CCD***) identifies whether the SDO request is of the read or write type. With a write command, the number of bytes to be written are also encoded in the CCD.
In the SDO response the CCD indicates whether the request was successful. In the case of a read command, the CCD also provides information on the number of bytes read:

CCD	Meaning	Valid for
0x23	Write 4 bytes	SDO request
0x27	Write 3 bytes	SDO request
0x2B	Write 2 bytes	SDO request
0x2F	Write 1 byte	SDO request
0x60	Write successful	SDO response
0x80	Error	SDO response
0x40	Read request	SDO request
0x43	4 bytes of data	SDO response to read request
0x47	3 bytes of data	SDO response to read request
0x4B	2 bytes of data	SDO response to read request
0x40	1 byte of data	SDO response to read request

Table 4: Command codes for SDO

In the case of an error (SDO response CCD = 0x80), the data range contains a 4-byte-error code, which provides information on the cause of the error. The meaning of the error codes can be found in Table 32, page 247.

Examples:

1. In the following record, first of all the COB-ID, then the message length and finally the data range of the CAN message are shown.

```
670       8   40 18 10 04 00 00 00 00
5F0       8   43 18 10 04 01 00 00 00
```

The first message has COB-ID 0x670 and is thus an SDO from the client to the node with address 0x70. This is a read request (CCD = 0x40) for parameter 0x1018, subindex 4 (Little Endian Format).
The second message has COB-ID 0x5F0 (=0x580+0x70). This is therefore an SDO response from the node with address 0x70 to the SDO client. 4 bytes of data have been read and the parameter value is 0x0000 0001.

2.
```
670       8   40 60 60 00 00 00 00 00
5F0       8   80 60 60 00 01 00 01 06
```

The first message is a read request from the SDO client to node 0x70 for parameter 0x6060, subindex 0. The second message is the node's response. CCD 0x80 indicates that the request could not be executed. 0x0601 0001 is specified as error code. According to Table 32, an attempt was made to read an object for which only write access is permitted.

encoTRive also supports the SDO transmission of data more than 4 bytes in length. With this segmented transmission, several messages are necessary for the data transfer.

4.2.2.2 PDO (Process Data Object), SYNC (synchronization)

With a special message, the ***SYNC message***, a so-called ***SYNC Producer*** can reach all nodes and thus synchronize the message exchange.

encoTRive is a SYNC Consumer, so can only receive SYNC messages and react to them.

A CANopen slave goes into **"PRE-OPERATIONAL"** state after switch-on. SDOs, but not PDOs, may be transmitted in this status, and it is possible to configure PDOs using SDOs. In the case of PDOs, the entire user data range of the CAN message (8 bytes) can be used for the transmission of **parameter values**.

encoTRive supports up to six PDOs in each transmission direction.

The structure and contents that are transported with a PDO are defined with special parameters, the ***PDO Mapping Parameters*** (Figure 6).

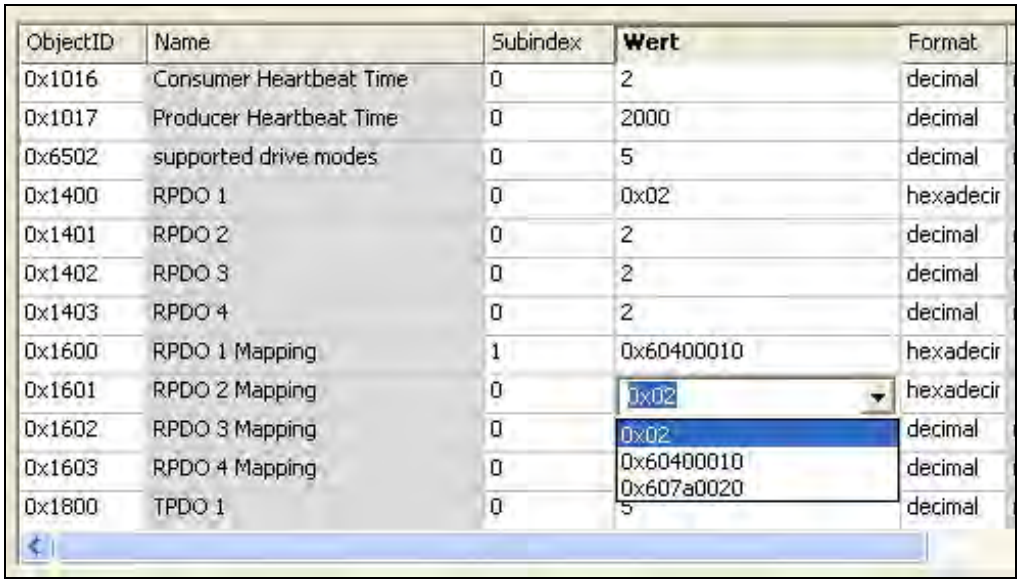


Figure 6: PDO Mapping Parameters

This so-called ***PDO Mapping*** specifies which parameters are transmitted in a PDO, in which sequence. Figure 6 shows the mapping for Receive PDO 1 (RPDO1) and Receive PDO 2 (RPDO2) from the viewpoint of the drive: Only object 0x6040 (control word) is transmitted in RPDO1. RPDO2 consists of object `0x6040` and also of object `0x607A` (target position).

In addition, the ***PDO Communication Parameters*** allow you to define under which conditions a PDO is to be transmitted and when a transmitted value is to be processed internally:

- ***Synchronous transmission***:
  In this case a node, the ***SYNC Producer***, periodically sends an SYNC message, with which the message transmission can be synchronized network-wide. In the case of a PDO that is configured for synchronous transmission, sending and processing of the received data are linked to the SYNC message.

- ***Asynchronous transmission***:
  The transmission of data is not linked to the SYNC message. It is event-controlled.

In this way a Transmit PDO can be configured event-controlled. In this case, the PDO is sent if the parameter value transmitted in the PDO has changed. Alternatively, a PDO can be sent if the transmission is requested by a SYNC message. Synchronous processing of a PDO can also be configured on the receive side. This means that the data are only processed when the next SYNC message is received. In this way, several drives can be started simultaneously, for example.

PDOs may only be transmitted when a slave is in **"OPERATIONAL"** status. In **"PRE-OPERATIONAL"** status, the PDO mapping and transmission type can be set for each SDO, before the slave switches over to **OPERATIONAL** status and the PDO transmission begins.

4.2.2.2.1 Request PDOs: Remote Transmission Request

A PDO consumer can request PDO messages from a PDO producer via the **Remote Transmission Request**. To do this, the consumer sends a message with the COB-ID of the requested PDO, in which case the RTR bit of the CAN message is set to 1. This denotes a PDO request. If the PDO producer is configured (transmission type), so that it will react to such requests, it sends the relevant PDO, in which case the RTR bit is set to 0.

4.2.2.2.2 Parameters / objects for PDO configuration

The following table contains the objects that are relevant for the PDO configuration. The details are described in the explanations of the individual parameters from chapter 4.2.4.4 "Objects of the communication profile DS 301" page 185.

Object no.	Description
0x1004	Number of supported PDOs
0x1005	COB-ID of the SYNC message
0x1400	Parameters for RPDO1: COB-ID, Transmission type
0x1401	Parameters for RPDO2: COB-ID, Transmission type
0x1402	Parameters for RPDO3: COB-ID, Transmission type
0x1403	Parameters for RPDO4: COB-ID, Transmission type
0x1404	Parameters for RPDO5: COB-ID, Transmission type
0x1405	Parameters for RPDO6: COB-ID, Transmission type
0x1600	Mapping for RPDO1
0x1601	Mapping for RPDO2
0x1602	Mapping for RPDO3
0x1603	Mapping for RPDO4
0x1604	Mapping for RPDO5
0x1605	Mapping for RPDO6
0x1800	Parameters for TPDO1: COB-ID, Transmission type, Inhibit Time
0x1801	Parameters for TPDO2: COB-ID, Transmission type, Inhibit Time
0x1802	Parameters for TPDO3: COB-ID, Transmission type, Inhibit Time
0x1803	Parameters for TPDO4: COB-ID, Transmission type, Inhibit Time
0x1804	Parameters for TPDO5: COB-ID, Transmission type, Inhibit Time
0x1805	Parameters for TPDO6: COB-ID, Transmission type, Inhibit Time
0x1A00	Mapping for TPDO1
0x1A01	Mapping for TPDO2
0x1A02	Mapping for TPDO3
0x1A03	Mapping for TPDO4
0x1A04	Mapping for TPDO5
0x1A05	Mapping for TPDO6

Table 5: Parameters for PDO configuration

4.2.2.3 EMCY (Emergency Service)

Internal device errors are reported with the EMCY message. As PDOs are an unconfirmed service, an invalid parameter value, which is set via PDO, for example, can result in an EMCY message. Other typical events that are indicated via EMCY are overload situations or overtemperature.

encoTRive can send EMCY messages, but not receive EMCY messages.

In the case of an EMCY message, the 8 bytes of user data in the CAN message can be used to provide information on the cause of the error:

Error code		Error register	Manufacturer-specific error information				
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7

Table 6: EMCY message

encoTRive does not use the field "Manufacturer-specific error information".

Object no.	Description
0x1001	Error register. Stores the value of the "Error register" field of the last EMCY message
0x1003	Pre-defined error field. Stores the contents of the "Error code" field of the last EMCY messages. The last 8 error codes can be stored. With each new EMCY message, the older error codes are moved back one index.
0x1014	COB-ID of the EMCY message

Table 7: Parameters for EMCY

The meaning of the various EMCY error information can be taken from Table 33 and Table 34 from page 248.

Boot-up message

The CANopen profile defines an additional function of the EMCY message:
An EMCY message without a data field represents a ***Boot-up message***, with which a node indicates - after switch-on of all other nodes - that it is ready for operation.

4.2.2.4 Heartbeat

The ***Heartbeat Protocol*** is a protocol for error detection. The ***Heartbeat Producer*** sends a cyclical message with a low priority. This message is received and evaluated by several nodes (***heartbeat consumers***). If the heartbeat message fails, a corresponding EMCY message is sent.

encoTRive can operate as heartbeat producer and as heartbeat consumer.

Object no.	Description
0x1016	Consumer Heartbeat Time Definition of the time interval within which a heartbeat message is expected.
0x1017	Producer Heartbeat Time Time interval for sending a heartbeat message.

Table 8: Parameters for heartbeat

4.2.2.5 Network Management Services

The ***Network Management*** (***NMT***) has the task of initializing nodes of a CANopen network, incorporating the nodes into the network, stopping them and monitoring them.

NMT services are initiated by an ***NMT master***, which addresses individual nodes (***NMT slave***) via their Node ID. An NMT message with Node ID 0 is addressed to **all** NMT slaves.

encoTRive is an NMT slave.

4.2.2.5.1 NMT services for device control

The NMT services for device control use the **COB-ID 0** and thus receive the highest priority.

Only the first two bytes of the CAN message data field are used:

Command	Node ID
Byte 0	Byte 1

Table 9: NMT message for device control

The following commands are defined:

Command	Meaning	Status
-	Automatic initialization after switch-on	(1)
-	End of initialization --> PRE-OPERATIONAL	(2)
0x01	**Start Remote Node** Node must change to OPERATIONAL status and start normal network operation	(3),(6)
0x02	**Stop Remote Node** Node must go to STOPPED status and stop communication. Any active communication monitoring will remain active.	(5),(8)
0x80	**Enter PRE-OPERATIONAL** Node must go into PRE-OPERATIONAL status. All messages except PDOs can be used.	(4),(7)
0x81	**Reset Node** Set values of the object directory profile parameters to default values. Then transition to the RESET COMMUNICATION status.	(9),(10),(11)
0x82	**Reset Communication** Node must go into the RESET COMMUNICATION status and load the values of the object directory communication parameters with default values. Then transition to INITIALIZATION status, first status after switch-on.	(12),(13),(14)

Table 10: NMT services for device control

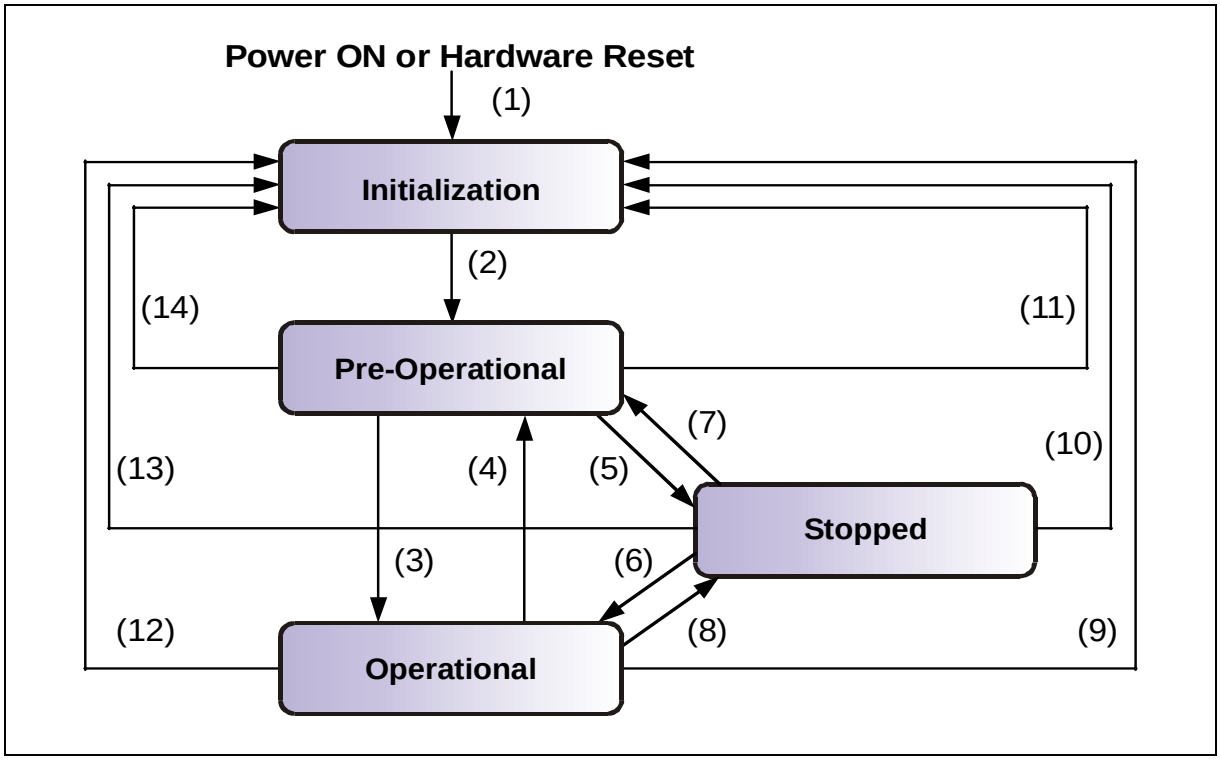


Figure 7: NMT boot-up mechanism

4.2.2.5.2 NMT services for connection monitoring

Connection monitoring enables an NMT master to detect the failure of an NMT slave and/or an NMT slave to detect the failure of the NMT master:

- ***Node Guarding*:**
  An NMT master monitors an NMT slave **with this service**
- ***Life Guarding*:**
  An NMT slave monitors an NMT master **with this service**

In the case of **Node Guarding**, the NMT master requests the status of an NMT slave at regular intervals. If no return message is received within a defined time interval, the NMT master assumes a failure of the NMT slave.

If **Life Guarding** is active, the NMT slave expects such a status request from the NMT master within a defined time interval.

The NMT services for connection monitoring use the function code **`1110 bin`**, so **COB-ID 0x700+Node ID**.

Object no.	Description	
0x100C	**Guard Time [ms]**	After expiry of the time interval **Life Time = Guard Time x Life Time Factor [ms]** at the latest, the NMT slave expects a status query from the master. If the Guard Time = 0, the corresponding NMT slave is not monitored by the master. If the Life Time = 0, Life Guarding is switched off.
0x100D	**Life Time Factor**	

Table 11: Parameters for NMT services

4.2.3 Drive-specific functions

4.2.3.1 DSP 402 state machine

If the drive as CANopen slave is in **OPERATIONAL**, status, drive-specific functions can be addressed. Figure 8 provides an overview of the states defined in the DSP 402 drive profile and the transitions between these states. The current status is shown by the status word (ZSW). In Figure 8 the states are represented by rectangles with rounded corners. The binary representation of the ZSW is provided in each case. Any bit that is marked with 'x' is not relevant for determining the state. The state transitions are marked by arrows, which specify the condition for the corresponding transition. In most cases this is a command in the control word (STW), which is also defined by certain bit combinations. Bits marked with 'x' are not relevant. STW.x denotes bit x of the STW, STW.7: 0->1 means "rising edge STW.7". In an error situation, you go from each of the states in the drawn-in rectangle to the "Error reaction active" state.

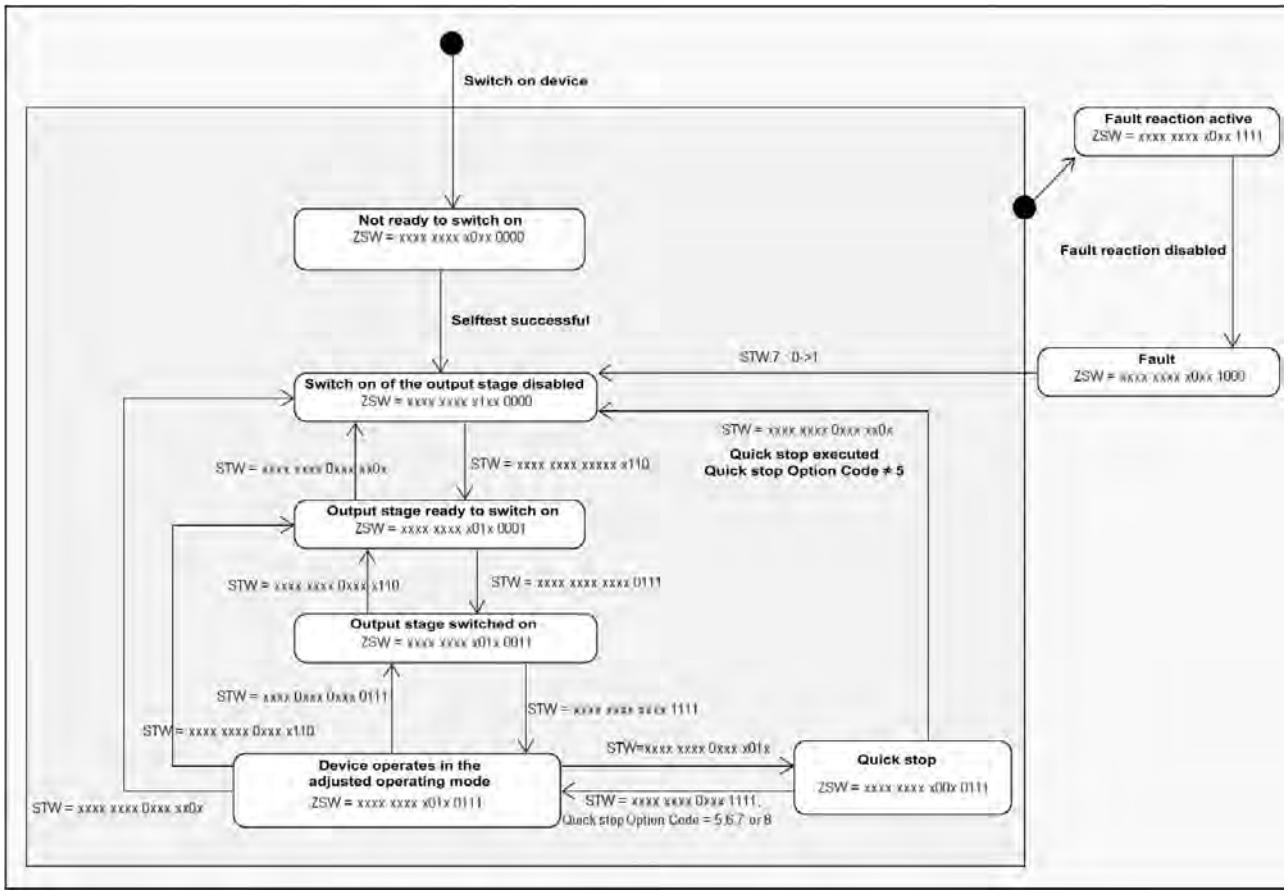


Figure 8: DSP 402 state machine

Table 12 on page 160 explains the different states.

Status	Condition	Description
Not Ready to switch on	ZSW = xxxx xxxx x0xx 0000	The drive is in the initialization phase.
Switch on disabled	ZSW = xxxx xxxx x1xx 0000	Drive initialization ended. The output stage cannot be switched on.
Ready to switch on	ZSW = xxxx xxxx x01x 0001	The output stage can be switched on. Drive parameters can be changed.
Switched on	ZSW = xxxx xxxx x01x 0011	Output stage switched on.
Operation enabled	ZSW = xxxx xxxx x01x 0111	The drive can be used in accordance with the set operating mode: Positioning ramp operating mode: Target position can be accepted. Speed ramp operating mode: Drive accelerates to the preset target speed.
Quick stop active	ZSW = xxxx xxxx x00x 0111	The drive performs a quick stop. If the Quick Stop Option Code (Object 0x605A) is 5, the drive then remains in this state.
Fault reaction active	ZSW = xxxx xxxx x0xx 1111	There is an error. The relevant troubleshooting is carried out.
Fault	ZSW = xxxx xxxx x0xx 1000	Troubleshooting is complete.

Table 12: States

The transition between these states occurs partially via internal events, partially via commands that are transferred in the control word:

Transition	Event / Command	Description
Start → Not ready to switch on	The drive is switched on / reset	The drive performs an initialization.
Not ready to switch on → Switch-on inhibit	The drive initialization is ended.	Communication is activated.
Not ready to switch on → Ready to switch on	The drive receives the "Shutdown" command: STW = xxxx xxxx 0xxx x110	
Ready to switch on → Switched on	The drive receives the "Switch On" command: STW = xxxx xxxx 0xxx 0111	The output stage is released
Switched on → Ready for operation	The drive receives the "Enable Operation" command: STW = xxxx xxxx 0xxx 1111	
Ready for operation → Switched on	The drive receives the "Disable Operation" command: STW = xxxx xxxx 0xxx 0111	
Switched on → Ready to switch on	The drive receives the "Shutdown" command: STW = xxxx xxxx 0xxx x110	The drive unit is deactivated.
Ready to switch on → Switch-on inhibit	The drive receives the "Quick Stop" command: STW = xxxx xxxx 0xxx x01x	
Ready for operation → Ready to switch on	The drive receives the "Shutdown" command: STW = xxxx xxxx 0xxx x110	The drive unit is switched off. If it is in motion, it is stopped according to the Shutdown Option Code (object 0x605B).
Ready for operation → Switch-on inhibit	The drive receives the "Disable Voltage" command: STW = xxxx xxxx 0xxx xx0x	The drive unit is switched off. If it is in motion, it is stopped as quickly as possible and then deactivated.
Switched on → Switch-on inhibit	The drive receives the "Disable Voltage" command STW = xxxx xxxx 0xxx xx0x or "Quick Stop" command (STW = xxxx xxxx 0xxx x01x)	The drive unit is switched off.
Ready for operation → Quick stop active	The drive receives the "Quick Stop" command: STW = xxxx xxxx 0xxx x01x	The drive unit is stopped according to object 0x605A (Quick Stop Option Code).

Continued:

Transition	Event / Command	Description
Quick stop → Switch-on inhibit	The quick stop is executed, or the "Disable Voltage" command is received: STW = xxxx xxxx 0xxx xx0x	The transition is possible if the value ≠ 5 is stored in object 0x605A (Quick Stop Option Code). The drive unit is then stopped as quickly as possible and then switched off.
From any status → Error reaction	An error has occurred in the drive.	The relevant troubleshooting is carried out.
Error reaction → Error	Troubleshooting is complete.	
Error → Switch-on inhibit	The error was acknowledged with a rising edge in bit 7 of the STW.	

Table 13: State transitions

4.2.3.2 Control word and status word

In addition to information on states and state transitions, STW and ZSW contain further meanings, which partially depend on the current operating mode.
The control and status words are therefore specified in more detail in the individual operating modes.

4.2.3.3 "Positioning ramp" operating mode

4.2.3.3.1 Control word

Bit	CONTROL WORD		
0	Value 1: Switch ready for operation	These bits are used to control the state machine	
1	Value 1: Enable voltage		
2	Value 1: Decelerate with intermediate stop ramp		
3	Value 1: Execute operating mode		
4	New target position	0	Do not accept target position
		1	Accept target position
5	Accept parameter immediately	0	End current positioning, start next positioning
		1	Abort current positioning, start next positioning
6	Absolute / relative positioning	0	Target position is an absolute value
		1	Target position is a relative value
7	Transition 0->1: Reset error		
8	Stop	0	Execute positioning
		1	Stop axis
9-15	not used		

Table 14: Control word (object 0x6040), "Positioning ramp" operating mode

4.2.3.3.2 Status word

Bit	STATUS WORD		
0	Value 1: Ready to switch on		
1	Value 1: Ready for operation		
2	Value 1: Operating mode activated		
3	Value 1: Error present		
4	Value 1: Voltage switched on		
5	Value 1: Quick stop active		
6	Value 1: Not ready for operation		
7	Value 1: Warning present		
8	not used		
9	Manual operation	0	No parameter access via CAN bus
		1	Parameter access via CAN bus
10	Target reached	0	Stop = 0: Target position not reached Stop = 1 Axis decelerates
		1	Stop = 0: Target position reached Stop = 1 Axis speed = 0
11	Value 1: Exit working range Position value outside the preset range. Definition via object 0x607D.		
12	Acknowledgement of target position	0	Positioning values not accepted
		1	Positioning values accepted
13	Value 1: Tracking error A tracking error is indicated if the difference between position value and calculated ramp value is greater than the tracking error window preset in object 0x6065 for the minimum duration of the time interval specified in object 0x6066 (Tracking Error Timeout).		
14-15	not used		

Table 15: Status word (object 0x6041), "Positioning ramp" operating mode

4.2.3.3.3 Perform positioning

If the drive is in the "Ready for operation" state, positioning movements can be performed.

For this purpose the "Positioning ramp" operating mode must be set and the operating mode must then be executed with the control word:

- Object 0x6060 operating mode = 1

A positioning ramp is started by a falling edge (1->0) in STW.8. The positioning movement is carried out on a ***ramp***, which is derived from the current settings for the speed, acceleration and deceleration:

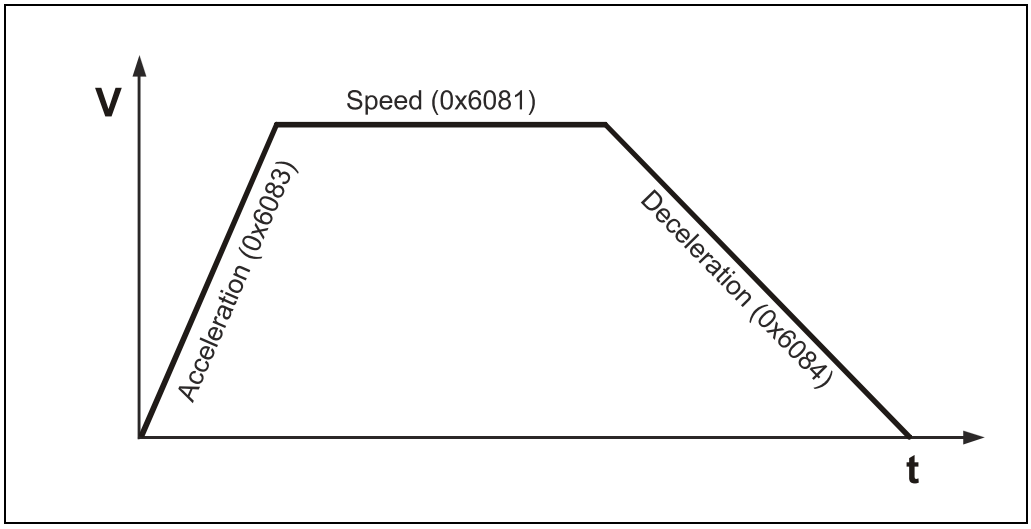


Figure 9: Positioning ramp

Depending on the distance between actual position at the start of positioning and target position, the required end speed (0x6081) is reached or not. If the travel is short, the phase with constant acceleration is directly followed by a phase with constant deceleration. There is no phase with constant speed in this case.

The conclusion of a positioning movement is indicated by ZSW.10. If this bit has the value 1, the positioning is complete. The applicable criterion for the end of a positioning movement is that the current position value lies within the position window (0x6067) around the target position for the duration of the position window time interval (0x6068).

4.2.3.3.4 Absolute / relative positioning

A positioning movement can be performed ***absolutely*** or ***relatively***. This is differentiated by means of STW.6. If this bit is set, a relative positioning movement is performed, otherwise an absolute positioning movement is performed.

In the case of **absolute positioning**, the value for the target position (0x607A) is interpreted absolutely, i.e. the distance to be covered is the difference between actual and target position. In the case of relative positioning, the content of object 0x607A is interpreted as the distance to be covered:

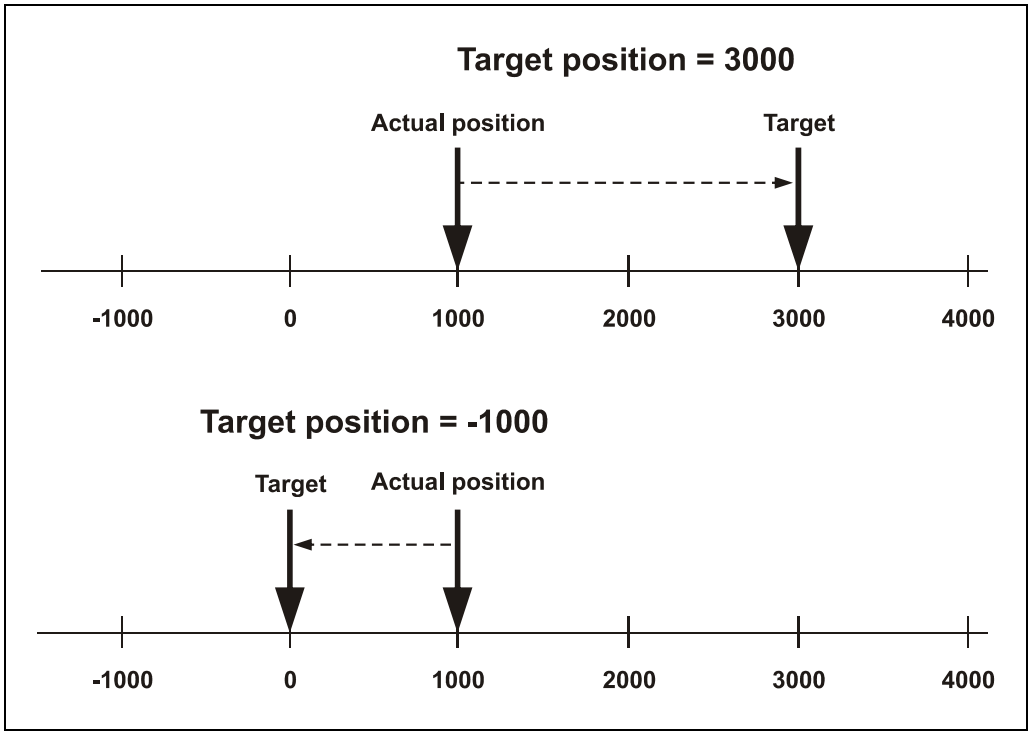


Figure 10: Absolute positioning (top) and relative positioning (bottom)

Figure 10 shows two examples:

In the top part of the figure, an absolute positioning movement is performed to the target position 3000. The drive changes its position until it reaches position 3000. A relative positioning movement is performed in the bottom part: Starting from the actual position 1000, a positioning movement by 1000 to the left (target position = -1000) is to be performed. This means that the positioning movement ends at the absolute position 0.

4.2.3.3.5 Transfer of new movement records

A new movement record (target position, speed, acceleration, deceleration) can be transferred to the drive while the previous positioning movement is still being performed. The movement record is activated with a positive edge in STW.4. Depending on STW.5 ("*Accept parameter immediately*"), the new movement record is acknowledged (ZSW.12 = 1) and executed immediately, or only after the current positioning movement is complete:
If STW.5 = 1, the new movement record is accepted "on-the-fly". The current positioning movement is not completed. If STW.5 = 0, the current positioning movement is performed first. The new positioning movement is then acknowledged (ZSW.12 = 1) and executed; whether or not the drive must stop depends on the current circumstances:
If the new target position requires travel in the other direction, or if the current speed is too high, the drive is stopped and a new ramp is started in the opposite direction. Otherwise, the current speed is compared with the required speed. If the travel is sufficient, the drive is accelerated or braked accordingly.

4.2.3.3.6 Termination of a positioning movement

If STW.8 ("*Stop*") is set during the movement, the current ramp is terminated and the drive stops. The deceleration at which this occurs can be specified in object 0x605D (Halt Option Code). Depending on the value of this parameter, deceleration occurs with the ramp for the quick stop or with the currently set deceleration. In order to then continue, STW.8 must be deleted and a new positioning movement started.

4.2.3.3.7 Relevant parameters

Index	Meaning
0x6040	Control word: Commands to the drive
0x6041	Status word: Return messages from the drive
0x6064	Current position
0x6067	Position window
0x6068	Position window time interval
0x607A	Target position
0x607D	Position range
0x607F	Maximum speed
0x6081	Speed (limited by object 0x607F)
0x6083	Acceleration (limited by object 0x60C5)
0x6084	Deceleration (limited by object 0x60C6)
0x60C5	Maximum acceleration
0x60C6	Maximum deceleration
0x605A	Quick Stop Option Code: Quick stop behavior
0x605D	Halt Option Code: Behavior in the event of termination of a ramp

Table 16: Positioning parameter

4.2.3.4 "Speed ramp" operating mode

WARNING!

There is a risk of physical injury and damage to property if the parameterized software limit switches in object 0x607D subindex 1 and 2 are exceeded!

- The parameterized software limit switches in object 0x607D, which refer to the position actual position value, are inoperative in "Speed ramp" operating mode.

  Range overruns can occur in rotary applications, for example, due to the integral position measuring system. Depending on the direction of rotation, this is expressed by a jump of the position value in object 0x6064:
  Max --> Min / Min --> Max.

  The application must therefore not be dependent on the actual value of position!

4.2.3.4.1 Control word

Bit	CONTROL WORD		
0	Value1: Switch ready for operation	These bits are used to control the state machine	
1	Value1: Enable voltage		
2	Value1: Decelerate with intermediate stop ramp		
3	Value1: Execute operating mode		
4-6	not used		
7	Transition 0->1: Reset error		
8	Stop	0	Execute speed ramp
		1	Stop axis
9-15	not used		

Table 17: Control word (object 0x6040), "Speed ramp operating mode"

4.2.3.4.2 Status word

Bit	STATUS WORD		
0	Value 1: Ready to switch on		
1	Value 1: Ready for operation		
2	Value 1: Operating mode activated		
3	Value 1: Error present		
4	Value 1: Voltage switched on		
5	Value 1: Quick stop active		
6	Value 1: Not ready for operation		
7	Value 1: Warning present		
8	not used		
9	Manual operation	0	No parameter access via CAN bus
		1	Parameter access via CAN bus
10	Target reached	0	Stop = 0: Target speed not reached Stop = 1 Axis decelerates
		1	Stop = 0: Target speed reached Stop = 1 Axis speed = 0
11	Value 1: Exit working range Position value outside the preset range. Definition via object 0x607D.		
12	Speed	0	Speed ≠ 0
		1	Speed = 0
13	Max. tracking error	0	Max. tracking error not reached
		1	Max. tracking error reached
14-15	not used		

Table 18: Status word (object 0x6041), "Speed ramp" operating mode

4.2.3.4.3 Execute speed ramp

When the drive is in "Ready for operation" state, its movement can be speed-controlled.

For this purpose the "Speed ramp" operating mode must be set and the operating mode must then be executed with the control word:

- Object 0x6060 operating mode = 3

The speed ramp is started by a falling edge (1->0) in STW.8.
Alternatively, the speed ramp can also be started as follows:
STW.1, STW.2: 0x06
--> STW.0: 0x07
--> STW.3: 0x0F Execute operating mode, drive rotates

The ***ramp*** results from the current settings for the target speed, acceleration and deceleration:

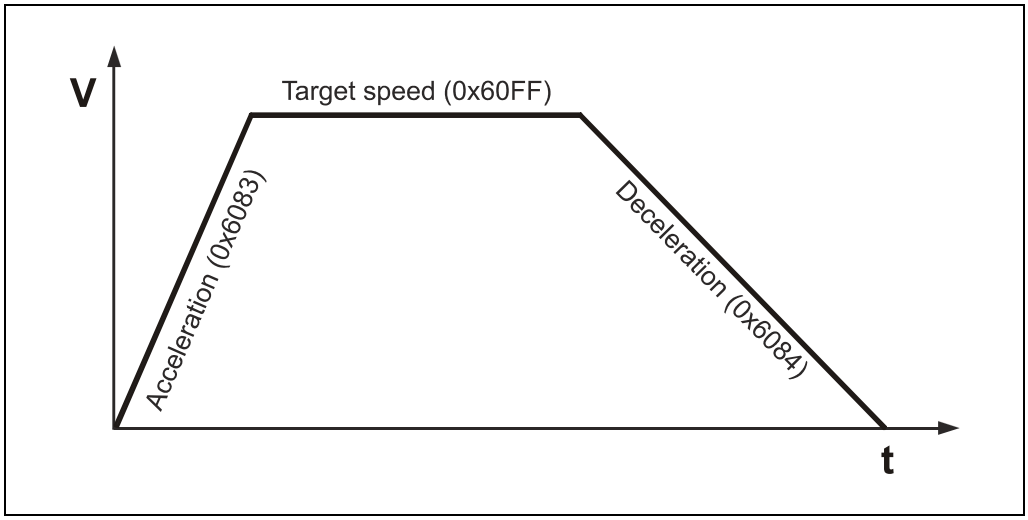


Figure 11: Speed ramp

The drive accelerates up to the preset target speed and retains this speed, until the drive is stopped manually.

The axis can be stopped using different methods:

- Set target speed in object 0x60FF to 0
- Stop axis with control word bit 8 = 0->1
- Execute NMT service "Stop Remote Node" command 0x02
- Reset STW.2: 0x07, end execution of operating mode

A change to the target speed during operation results in immediate execution of the ramp, which accelerates or decelerates the current speed value.

4.2.3.4.4 Relevant parameters

Index	Meaning
0x6040	Control word: Commands to the drive
0x6041	Status word: Return messages from the drive
0x6069	Measured speed
0x6071	Target torque
0x6072	Maximum torque
0x607E	Direction reversal
0x607F	Maximum speed
0x6080	Maximum motor speed
0x6083	Acceleration (limited by object 0x60C5)
0x6084	Deceleration (limited by object 0x60C6)
0x6085	Deceleration for quick stop
0x6094	Speed factor
0x606B	Nominal speed
0x606C	Actual speed
0x60FF	Target speed

Table 19: Speed ramp parameter

4.2.3.5 Units

For the conversion of different units, the DSP 402 profile provides the so-called ***Factor group***. From this group, encoTRive implements the following partial quantity:

4.2.3.5.1 Object 0x608F: Position encoder resolution

This parameter defines the ratio between position increments and motor revolutions:

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

Formula 1: Position encoder resolution

The fraction counter is defined in subindex 1, the denominator in subindex 2.
The default value is

$$\text{Position encoder resolution}_{\text{Default}} = \frac{1024}{1} = 1024\left[\frac{\text{Position increments}}{\text{Motor revolutions}}\right]$$

Formula 2: Default value, position encoder resolution

4.2.3.5.2 Object 0x6090: Speed encoder resolution

This parameter defines the ratio between speed increments per second and motor revolutions per second:

$$\text{Speed encoder resolution} = \frac{\frac{\text{Speed increments}}{\text{sec}}}{\frac{\text{Motor revolutions}}{\text{sec}}}$$

Formula 3: Speed encoder resolution

The fraction counter is defined by subindex 1, the denominator by subindex 2.

The default value is

$$\text{Speed encoder resolution}_{\text{Default}} = \frac{2^{31}}{5000}\left[\frac{\text{Speed increments}}{\text{Motor revolutions}}\right]$$

Formula 4: Default value, speed encoder resolution

4.2.3.5.3 Object 0x6093: Position factor

The position factor converts the desired position into increments, which the drive requires for the internal calculations. By multiplying with the position factor, position data can be converted into **user-defined *position units***.
Position units are length units, angles or increments.

$$\text{Position factor} = \frac{\text{Position encoder resolution} \bullet \text{Gear ratio}}{\text{Travel per gear revolution}}$$

Formula 5: Position factor

The position encoder resolution is defined by object 0x608F. The ***gear ratio*** denotes the gear reduction, given by

$$\text{Gear ratio} = \frac{\text{Motor revolutions}}{\text{Drive shaft revolutions}}$$

Formula 6: Gear ratio

The ***travel per gear revolution*** is given in position units.

The **counter** of the position factor is specified as subindex 1 of object 0x6093, the **denominator** as subindex 2.

The following applies:
Position [position units] * Position factor = Position [position increments]

The unit of the position factor is position increments/position units

The position unit is therefore implicitly defined by the position factor.

The default value of object [**0x6093**] for the position factor is

$$\text{Position factor}_{Default} = \frac{1024}{1024}$$

Formula 7: Default value, position factor

This value, together with the encoder resolution from Formula 2, corresponds to a gear reduction of 1:1 and the position unit increments.

Example 1:

- Position unit = [mm]
- Pitch (travel per gear revolution) = 3 mm
- Gear ratio = 1
- Position encoder resolution = 1024 inc

The following values must be entered in the position factor object [**0x6093**] :

Subindex 1: 1024 / position encoder resolution x gear factor
Subindex 2: 3 / pitch (travel per gear revolution)

Example 2:

- Position unit = [inc]
- Travel per gear revolution = 1024 inc
- Gear ratio = 1
- Position encoder resolution = 1024 inc

The following values must be entered in the position factor object [**0x6093**] :

Subindex 1: 1024 / position encoder resolution x gear factor
Subindex 2: 1024 / travel per gear revolution

Example 3:

- Position unit = [degrees /10]
- Travel per gear revolution = 360 degrees = 3600/10 degrees
- Gear ratio = 1
- Position encoder resolution = 1024 inc

The following values must be entered in the position factor object [**0x6093**] :

Subindex 1: 1024 / position encoder resolution x gear factor
Subindex 2: 3600 / travel per gear revolution (360 degrees)

4.2.3.5.4 Object 0x6094: Speed factor

The speed factor converts the desired speed into increments, which the drive requires for the internal calculations. By multiplying with the speed factor, speed data can be converted into user-defined ***speed units***.

$$\text{Speed factor} = \frac{\text{Speed encoder resolution} \bullet \text{Gear reduction}}{\text{Speed units\_per} \dfrac{\text{Output revolution}}{\text{sec}}}$$

Formula 8: Speed factor

The **counter** of the speed factor is specified in subindex 1 of object 0x6094, the **denominator** as subindex 2.

The default value is:

$$\text{Speed factor}_{Default} = \frac{2^{31}}{300000}$$

Formula 9: Default value, speed factor

This value, together with the speed encoder resolution from Formula 4 and a gear reduction of 1:1, corresponds to the speed unit "rpm".

Object 0x606C (Actual Velocity) contains the current output speed in the desired unit. Default unit "rpm".
This means, for example, that if a value of 4350 is preset in object 0x6081 (Profile Velocity), an output speed of 4350 rpm will be reached.
Object 0x607F (Max. Profile Velocity) defines the maximum permitted speed.
The desired unit is now used in all speed-related objects, with the exception of object 0x6069 (Sensor Velocity), which specifies the current speed in speed increments.

Example 1: Default

- Speed unit = r/min (at the output)
- Output revolution = 1 r/ sec = 60 r/min
- Gear ratio = 1
- Default value for speed encoder resolution, 2^{31}/ 5000, see Formula 4

Calculation: (2^{31}/ 5000) x 1 / 60 = 2^{31}/ [(5000*60) / 1]

The following values must be entered in the speed factor object [**0x6094**] :

Subindex 1: 2^{31}

Subindex 2: $\frac{5000 \bullet 60}{1} = 300000$

Example 2:

- Speed unit = (degrees/10) /sec (at the output)
- Output revolution =1 r/sec = 3600 (degrees /10 [1]) / sec
- Gear ratio = 1
- Default value for speed encoder resolution, 2^{31}/ 5000, see Formula 4

Calculation: (2^{31}/ 5000) x 1 / 3600 = 2^{31}/ [(5000*3600) / 1]

The following values must be entered in the speed factor object [**0x6094**] :

Subindex 1: 2^{31}

Subindex 2: $\frac{5000 \bullet 3600}{1} = 18000000$

Example 3:

- Speed unit = (mm/100) /sec (at the output)
- Travel per gear revolution = 2 mm / object (0x6093)
- Output revolution = 1 r/sec = 2 (mm x 100 [1] /sec)
- Gear ratio = 1
- Default value for speed encoder resolution, 2^{31}/ 5000, see Formula 4

Calculation: (2^{31}/ 5000) x 1 / 200 = 2^{31}/ [(5000*200) / 1]

The following values must be entered in the speed factor object [**0x6094**] :

Subindex 1: 2^{31}

Subindex 2: $\frac{5000 \bullet 200}{1} = 1000000$

[1] Resolution factor for higher resolution. Result: the possibility to exact value input.

4.2.3.5.5 Object 0x6097: Acceleration factor

The acceleration factor converts the acceleration into increment/s, which the drive requires for the internal calculations. By multiplying with the acceleration factor, acceleration data is converted into user-defined ***acceleration units***.

The **counter** of the acceleration factor is specified as subindex 1 of object 0x6097, the **denominator** as subindex 2.

$$\text{Acceleration factor} = \frac{\text{Speed encoder resolution} \bullet \text{Gear reduction}}{\text{Acceleration units\_per} \frac{\text{Output revolution}}{\text{sec}^2}}$$

Formula 10: Acceleration factor

The default value is:

$$\text{Acceleration factor}_{Default} = \frac{2^{31}}{300000}$$

Formula 11: Default for acceleration factor

This value, together with the speed encoder resolution from Formula 4 and a gear reduction of 1:1, corresponds to the acceleration unit "(rpm) / sec".

Example 1: Default

- Acceleration unit = (r/min) /sec (at the output)
- Output revolution = 1 r/ sec = 60 (r/min) /sec
- Gear ratio = 1
- Default value for speed encoder resolution, 2^{31}/ 5000, see Formula 4

Calculation: (2^{31}/ 5000) x 1 / 60 = 2^{31}/ [(5000*60) / 1]

The following values must be entered in the speed factor object [**0x6097**] :

Subindex 1: 2^{31}

Subindex 2: $\frac{5000 \bullet 60}{1} = 300000$

Example 2:

- Acceleration unit = (degrees/10) /sec^2 (at the output)
- Output revolution =1 r/sec = 3600 (degrees /10 [1]) / sec^2
- Gear ratio = 1
- Default value for speed encoder resolution, 2^{31}/ 5000, see Formula 4

Calculation: (2^{31}/ 5000) x 1 / 3600 = 2^{31}/ [(5000*3600) / 1]

The following values must be entered in the speed factor object [**0x6097**] :

Subindex 1: 2^{31}

Subindex 2: $\frac{5000 \bullet 3600}{1} = 18000000$

Example 3:

- Acceleration unit = (mm/100) /sec^2 (at the output)
- Travel per gear revolution = 2 mm / object (0x6093)
- Output revolution = [1] r/sec = 2 (mm x 100 1 /sec)
- Gear ratio = 1
- Default value for speed encoder resolution, 2^{31}/ 5000, see Formula 4

Calculation: (2^{31}/ 5000) x 1 / 200 = 2^{31}/ [(5000*200) / 1]

The following values must be entered in the speed factor object [**0x6097**] :

Subindex 1: 2^{31}

Subindex 2: $\frac{5000 \bullet 200}{1} = 1000000$

[1] Resolution factor for higher resolution. Result: the possibility to exact value input.

Conversion into position increments

With the above factors it is possible to convert any user-defined units

- for travel,
- speed and acceleration into position increments,
- speed increments/sec and speed increments/sec²

The following factors can be used to express the latter units in position increments again:

$$VelocityToPositionUnitFactor = \frac{\text{Speed factor} \bullet \text{Position encoder resolution}}{\text{Position factor} \bullet \text{Speed encoder resolution}}$$

$$AccelerationToPositionUnitFactor = \frac{\text{Acceleration factor} \bullet \text{Position encoder resolution}}{\text{Position factor} \bullet \text{Speed encoder resolution}}$$

By multiplying with the ***VelocityToPositionUnitFactor***, a user-defined speed in speed units is converted into a measuring system-related speed in position increments/sec. Similarly, user-defined acceleration data in acceleration units is converted into measuring system-related acceleration data in position increments/sec² by multiplying with the ***AccelerationToPositionUnitFactor***.

Examples:

In the following examples, the default values for speed encoder resolution and position encoder resolution are used as the basis.

1. Ramp calculation:

- Gear reduction = 1:1
- Covered distance s = 700000 degrees
- Acceleration from standstill = 150 rpm/sec
- Further acceleration from 150 rpm/sec to 2640 rpm, then constant
- Deceleration = 150 rpm/sec, until stationary

How long does the ramp processing take, and which sections are covered ?

The speed factor results from Formula 9.

The acceleration phase and the braking phase each last

$t_{acc} = t_{dec} = 2640/150 = 17.60$ sec.

VelocityToPositionUnitFactor = $[(2^{31}/300000)* 2^{16}]/[(2^{16}/360)* (2^{31}/5000)]$

$= 5000*360/300000 = 6$,

AccelerationToPositionUnitFactor = 6.

The speed of 2640 rpm is as follows in position units:

$v_{end} = 2640*6 = 15840$ degrees/sec.

The acceleration and deceleration of 150 rpm/sec results in

$a = b = 150*6 = 900$ degrees/sec²,

During the acceleration phase, the distance covered is

$s_a = 0.5* a* t_{acc}^2 = 0.5*900*17.6^2 = 139392$ degrees.

The deceleration distance is likewise

$s_b = t_{dec}* v_{end} – 0.5* b* t_{dec}^2 = 17.6*15840 – 0.5*900*17.6^2 = 139392$ degrees.

Consequently, for travel at constant speed v_{end}

$s – (s_a + s_b) = 700000 – 2*139392 = 421216$ degrees still remain.

This phase therefore lasts 421216/15840 = 26.6 sec.

In total, the ramp is (theoretically) traveled in 2*17.6+26.6 = 61.8 sec.

2. Ramp calculation:

- Drive with slide, powered by a screw link actuator (translatory movement)
- Feed = 1 mm/spindle revolution
- Gear reduction = 40:1 = 40 motor revolutions per spindle revolution
- Slide movement s = 1000 mm
- Acceleration from standstill = 20 rpm/sec
- Further acceleration from 20 rpm/sec to 240 rpm, then constant
- Deceleration = 60 rpm/sec, until stationary

Position factor = 2^{16}*40

1 mm feed = 2^{16}*40 position increments.

The following results for the speed factor:

speed factor = 40*2^{31}/300000.

This results in

VelocityToPositionUnitFactor = [(40*2^{31}/300000)*2^{16}]/[(40*2^{16})*(2^{31}/5000)]
= 5000/300000 = 1/60

AccelerationToPositionUnitFactor = 1/60

The acceleration phase lasts t_{acc} = 12 sec, the braking phase t_{dec} = 4 sec.

The following results for the acceleration:

a = 20*(1/60) = 1/3 mm/sec²,

and for the deceleration

b = 60*(1/60) = 1 mm/sec²,

The end speed of 240 rpm corresponds to

v_{end} = 240*(1/60) = 4 mm/sec.

During the acceleration phase, the distance covered is

$s_a = 0.5 * a * t_{acc}^2$ = 0.5*(1/3)*12² = 24 mm

The deceleration distance is likewise

$s_b = t_{dec} * v_{end} - 0.5 * b * t_{dec}^2 = 4*4 - 0.5*1*4^2$ = 8 mm.

Therefore, for travel at constant speed v_{end}

$s - (s_a + s_b)$ = 1000 – 28 = 968 mm still remain.

This phase therefore lasts 968/4 = 242 sec.

In total, the ramp is (theoretically) traveled in 258 sec.

4.2.4 The object directory

4.2.4.1 Object types, data types

A parameter in the CANopen object directory can be a simple value, an array or a data structure. encoTRive uses the following types, which are distinguished by the ***object code***:

Object code	Name	Meaning
7	VAR	Simple value, e.g. INTEGER8
8	ARRAY	Array from several elements of the same data type
9	RECORD	Data field, which is a combination of various simple data types

Table 20: Object codes in encoTRive

In the case of an ARRAY or RECORD parameter, the individual elements are accessed via the subindex. For simple values (VAR), the subindex is 0.

A parameter or an element of a parameter also has attributes which define the access to this parameter:

Attribute	Meaning
rw	read/write: Parameter can be read and written
ro	read only: Parameter can only be read
wo	write only: Parameter can only be written
const	Value is constant and read only.

Table 21: Attributes

encoTRive uses the following data types:

Coding	Data type	Length	Description
1	BOOLEAN	8 bit	Two possible values: 0 (false) or 1 (true)
2	INTEGER8	8 bit	Integer 8-bit value with sign. Range of values: -128 ... 127
3	INTEGER16	16 bit	Integer 16-bit value with sign. Range of values: -32768 ... 32767
4	INTEGER32	32 bit	Integer 32-bit value with sign. Range of values: $-2^{31}...2^{31}-1$
5	UNSIGNED8	8 bit	Integer 8-bit value without sign. Range of values: 0...255
6	UNSIGNED16	16 bit	Integer 16-bit value without sign. Range of values: $0..2^{16}-1$ (0-65535)
7	UNSIGNED32	32 bit	Integer 32-bit value without sign. Range of values: $0..2^{32}-1$
9	Visible String	Variable	ASCII character string

Table 22: CANopen data types used by encoTRive

4.2.4.2 EDS file

In a text file, the ***Electronic Data Sheet***, the following is specified:

- which objects a CANopen node implements,
- which type of objects are involved,
- how the objects can be accessed

The format of the EDS file is defined in an individual standard, DSP 306 [CiA(2001)].

A configuration tool can load the EDS file and thus obtain information on which objects are present and how they can be accessed.

The EDS file (file ending "*.eds*") is included in the scope of supply of encoTRive. An excerpt from the EDS file for encoTRive follows:

```
[6081]
ParameterName=Profile Velocity
ObjectType=0x07
DataType=0x0007
LowLimit=1
AccessType=rw
DefaultValue=12000
PDOMapping=1
; Unit: Velocity unit
; Def.: 12000 rpm

[6083]
ParameterName=Profile Acceleration
ObjectType=0x07
DataType=0x0007
LowLimit=1
AccessType=rw
DefaultValue=3000
PDOMapping=1
; Unit: Acceleration unit
; Def.: 3000 rpm/sec

[6084]
ParameterName=Profile Deceleration
ObjectType=0x07
DataType=0x0007
LowLimit=1
AccessType=rw
DefaultValue=3000
PDOMapping=1
; Unit: Acceleration unit
; Def.: 3000 rpm/sec
```

Figure 12: Excerpt from an encoTRive EDS

4.2.4.3 Explanations of the parameter list

The following descriptions are used in the following list of all encoTRive objects:

Subindex — Selects the individual elements of an object.

Name — Name of the object/parameter

Attribute — Specification in the form

Access/Flash memory/Factory setting

Access: Access to the parameter (see Table 21)

Flash memory:	f	Parameter is stored in Flash at the request "Store in Flash"
	-	Parameter is not stored in Flash
Factory setting:	**w**	Parameter is preconfigured with default value when factory settings are loaded
	-	Value is not preset to the default value

Object code — Parameter type according to Table 20.

Default — Factory setting

Min — Minimum value

Max — Maximum value

PDO mapping — Yes: The object can be transferred as part of a PDO
- : no PDO mapping

4.2.4.4 Objects of the communication profile DS 301

4.2.4.4.1 Object 0x1000: Device type

Object code	VAR
Data type	UNSIGNED32
Attribute	ro
Default	0x20192
Min	-
Max	-
PDO mapping	-
Description	The low-value 16 bits of the device type specify the device profile DSP 402 = 0x192 Value 2 in the higher value word denotes a servo drive

4.2.4.4.2 Object 0x1001: Error register

Object code	VAR
Data type	UNSIGNED8
Attribute	ro
Default	0
Min	-
Max	-
PDO mapping	yes
Description	The error register displays the error status of the device in bit code. A set bit indicates that a corresponding error is present. The exact cause of the error is indicated by object 0x1003. At the time of occurrence, an error is indicated by an EMCY message.

Bit	Meaning
0	General error
1	Current
2	Voltage
3	Temperature
4	Communication error (overflow, status error)
5	Device profile-specific error
6	reserved
7	Manufacturer-specific

4.2.4.4.3 Object 0x1003: Predefined error field

Object code	ARRAY	
Description	This parameter stores the last 8 occurring errors as maximum. The "newest" error is stored in subindex 1. As the subindex increases, the error entries become older. The error entries are of the type UNSIGNED32. The higher-value part (bits 16-31) is intended for device-specific information and is not used by the encoTRive. The meaning of bits 0-15 can be taken from Table 34 on page 250.	
	Subindex 0: Number of error entries	
	Data type	UNSIGNED8
	Attribute	rw
	Default	0
	Min	0
	Max	8
	PDO mapping	-
	Description	Contains the number of stored errors. Writing the value 0 in this element means that the complete error list is deleted. Other values are not permitted and result in an abort message with the error code 0x0609 0030.
	Subindex 1: Error no. 1	
	Data type	UNSIGNED32
	Attribute	ro
	Default	0
	Min	0
	Max	-
	PDO mapping	-
	Description	Last occurring error
	Subindex 2: Error no. 2	
	Data type	UNSIGNED32
	Attribute	ro
	Default	0
	Min	0
	Max	-
	PDO mapping	-
	Description	Older error message (position 2)
	...	
	Subindex 8: Error no. 8	
	Data type	UNSIGNED32
	Attribute	ro
	Default	0
	Min	0
	Max	-
	PDO mapping	-
	Description	Oldest error message (position 8)

4.2.4.4.4 Object 0x1004: Number of supported PDOs

This object is not implemented at present, but is required by certain controls in order to be able to read the predefined PDOs.

Object code	RECORD	
Description	This parameter specifies how many PDOs are supported in each transmission direction.	
	Subindex 0: Number of PDOs	
	Data type	UNSIGNED32
	Attribute	ro
	Default	0x00060006
	Min	-
	Max	-
	PDO mapping	-
	Description	Contains the maximum number of PDOs in both transmission directions. Bit 0-15: PDOs sent by encoTRive (6) Bit 16-31: PDOs received by encoTRive (6)
	Subindex 1: Number of synchronous PDOs	
	Data type	UNSIGNED32
	Attribute	ro
	Default	0x00060006
	Min	-
	Max	-
	PDO mapping	-
	Description	Contains the maximum number of synchronous PDOs in both transmission directions. Bit 0-15: PDOs sent by encoTRive (6) Bit 16-31: PDOs received by encoTRive (6)
	Subindex 2: Number of asynchronous PDOs	
	Data type	UNSIGNED32
	Attribute	ro
	Default	0x00060006
	Min	-
	Max	-
	PDO mapping	-
	Description	Contains the maximum number of asynchronous PDOs in both transmission directions. Bit 0-15: PDOs sent by encoTRive (6) Bit 16-31: PDOs received by encoTRive (6)

4.2.4.4.5 Object 0x1005: COB-ID of the SYNC message

Object code	VAR
Data type	UNSIGNED32
Attribute	rw
Default	0x00000080
Min	-
Max	-
PDO mapping	-
Description	The object defines the COB-ID of the SYNC message and specifies whether a SYNC message is sent by the device.

Bit	Meaning
31	without significance
30	0: Device does not generate a SYNC message 1: Device generates a SYNC message
29	0: 11-bit identifier (CAN 2.0A) 1: 29-bit identifier (CAN 2.0B)
28-0	Identifier (29 bit or 11 bit)

Default value = 0x0000 0080:

- encoTRive does not send a SYNC message,
- encoTRive uses the 11-bit identifier 0x80 for a SYNC message

4.2.4.4.6 Object 0x1008: Manufacturer's device name

Object code	VAR
Data type	Visible String
Attribute	const
Default	„EncoTRive“
Min	-
Max	-
PDO mapping	-
Description	The object contains the device name, which consists of the ASCII character string 'E', 'n', 'c', 'o', 'T', 'R', 'i', 'v', 'e'.

4.2.4.4.7 Object 0x1009: Manufacturer's hardware version

Object code	VAR
Data type	Visible String
Attribute	const
Default	„736018a_1kx64k“
Min	-
Max	-
PDO mapping	-
Description	The object contains the hardware version of the device. This consists of an ASCII character string.

4.2.4.4.8 Object 0x100A: Manufacturer's software version

Object code	VAR
Data type	Visible String
Attribute	const
Default	„V4.5 17-Jan-2007 by MAH“
Min	-
Max	-
PDO mapping	-
Description	The object contains the software version of the device. This consists of an ASCII character string. The default value depends on from used firmware.

4.2.4.4.9 Object 0x100C: Guard Time

Object code	VAR
Data type	UNSIGNED16
Attribute	rw
Default	0
Min	0
Max	65535
PDO mapping	-
Description	This parameter defines the time interval for monitoring of the slave by the master. If the parameter is 0, the slave is not monitored. For monitoring of the master by the slave, the time span Life Time = Guard Time x Life Time Factor (Object 0x100D) in ms is relevant. If Life Time = 0, there is no monitoring of the master by the slave (no Life Guarding).

4.2.4.4.10 Object 0x100D: Life Time Factor

Object code	VAR
Data type	UNSIGNED8
Attribute	rw
Default	0
Min	0
Max	255
PDO mapping	-
Description	For monitoring of the master by the slave, the time span Life Time = Guard Time (Object 0x100C) x Life Time Factor in ms is relevant. If Life Time = 0, there is no monitoring of the master by the slave (no Life Guarding).

4.2.4.4.11 Object 0x1010: Store parameters

Object code	ARRAY	
Description	This parameter causes the remanent storage of defined parameters. The parameters to be stored are selected by means of the parameter elements.	
	Subindex 0: Number of entries	
	Data type	UNSIGNED8
	Attribute	ro
	Default	4
	Min	-
	Max	-
	PDO mapping	-
	Description	Contains the largest array subindex
	Subindex 1: Store all parameters	
	Data type	UNSIGNED32
	Attribute	rw
	Default	1
	Min	-
	Max	-
	PDO mapping	-
	Description	Stores all storable parameters. The parameters are only stored if the value **0x65766173** is written. This is the numeric value resulting from the "save" character string ('s' : 0x73, 'a': 0x61, 'v' : 0x76, 'e': 0x65).

Subindex 2: Store communication parameters	
Data type	UNSIGNED32
Attribute	rw
Default	1
Min	-
Max	-
PDO mapping	-
Description	Stores all storable parameters from the range 0x1000-0x1FFF. The parameters are only stored if the value **0x65766173** is written. This is the numeric value resulting from the "save" character string ('s' : 0x73, 'a': 0x61, 'v' : 0x76, 'e': 0x65).

Subindex 3: Store application parameters	
Data type	UNSIGNED32
Attribute	rw
Default	1
Min	-
Max	-
PDO mapping	-
Description	Stores all storable drive profile parameters (range 0x6000-0x9FFF). The parameters are only stored if the value **0x65766173** is written. This is the numeric value resulting from the "save" character string ('s' : 0x73, 'a': 0x61, 'v' : 0x76, 'e': 0x65).

Subindex 4: Store manufacturer-specific parameters	
Data type	UNSIGNED32
Attribute	rw
Default	1
Min	-
Max	-
PDO mapping	-
Description	Stores all storable manufacturer-specific parameters (range 0x2000-0x5FFF). The parameters are only stored if the value **0x65766173** is written. This is the numeric value resulting from the "save" character string ('s' : 0x73, 'a': 0x61, 'v' : 0x76, 'e': 0x65).

4.2.4.4.12 Object 0x1011: Load factory settings

Object code	ARRAY	
Description	This parameter causes defined parameters to be configured with their factory settings.	
	Subindex 0: Number of entries	
	Data type	UNSIGNED8
	Attribute	ro
	Default	4
	Min	-
	Max	-
	PDO mapping	-
	Description	Contains the largest array subindex
	Subindex 1: Factory settings for all parameters	
	Data type	UNSIGNED32
	Attribute	rw
	Default	1
	Min	-
	Max	-
	PDO mapping	-
	Description	Configures all parameters, for which factory settings are available, with their factory settings. This only occurs if the value **0x64616F6C** is written. This is the numeric value resulting from the "load" character string ('l' : 0x6C, 'o': 0x6F, 'a' : 0x61, 'd': 0x64).
	Subindex 2: Factory settings for communication parameters	
	Data type	UNSIGNED32
	Attribute	rw
	Default	1
	Min	-
	Max	-
	PDO mapping	-
	Description	Configures all parameters from the range 0x1000-0x1FFF, for which factory settings are available, with their factory settings. This only occurs if the value **0x64616F6C** is written. This is the numeric value resulting from the "load" character string ('l' : 0x6C, 'o': 0x6F, 'a' : 0x61, 'd': 0x64).

Subindex 3: Factory settings for application parameters	
Data type	UNSIGNED32
Attribute	rw
Default	1
Min	-
Max	-
PDO mapping	-
Description	Configures all parameters from the range 0x6000-0x9FFF, for which factory settings are available, with their factory settings. This only occurs if the value **0x64616F6C** is written. This is the numeric value resulting from the "load" character string ('l' : 0x6C, 'o': 0x6F, 'a' : 0x61, 'd': 0x64).

Subindex 4: Factory settings for manufacturer-specific parameters	
Data type	UNSIGNED32
Attribute	rw
Default	1
Min	-
Max	-
PDO mapping	-
Description	Configure all parameters from the range 0x2000-0x5FFF, for which factory settings are available, with their factory settings. This only occurs if the value **0x64616F6C** is written. This is the numeric value resulting from the "load" character string ('l' : 0x6C, 'o': 0x6F, 'a' : 0x61, 'd': 0x64).

4.2.4.4.13 Object 0x1014: COB-ID of the EMCY message

Object code	VAR
Data type	UNSIGNED32
Attribute	ro
Default	Node ID + 0x80
Min	-
Max	-
PDO mapping	-
Description	This parameter defines the COB-ID of the EMCY message. A node with Node ID 0x23 uses 0x23+0x80 = 0xA3 as COB-ID for the EMCY message

4.2.4.4.14 Object 0x1015: Inhibit Time for EMCY

Object code	VAR
Data type	UNSIGNED16
Attribute	rw
Default	96
Min	0
Max	65535
PDO mapping	-
Description	This parameter defines the inhibit time for the EMCY message in 0.1 ms steps. Parameter value 100 = 10 ms. If encoTRive has issued an EMCY message, the next EMCY message may not be sent before this time has expired. Parameter value 0 = no inhibit time for EMCY.

4.2.4.4.15 Object 0x1016: Consumer Heartbeat Time

Object code	ARRAY	
Description	This parameter defines the time interval within which a heartbeat message is expected. A maximum of two nodes are supported as heartbeat producer.	
	Subindex 0: Number of entries	
	Data type	UNSIGNED8
	Attribute	ro
	Default	2
	Min	-
	Max	-
	PDO mapping	-
	Description	Contains the largest array subindex
	Subindex 1: Consumer Heartbeat Time 1	
	Data type	UNSIGNED32
	Attribute	rw
	Default	0
	Min	-
	Max	-
	PDO mapping	-
	Description	The parameter value comprises two pieces of data:

Bits 31-24	Bits 23-16	Bits 15-0
0	Node ID	Heartbeat time

The heartbeat time defines the time interval within which a heartbeat message is expected from the node whose node ID is stored in bits 23-16. A heartbeat time of 0x00 means that no heartbeat message is expected.

Subindex 2: Consumer Heartbeat Time 2	
Data type	UNSIGNED32
Attribute	rw
Default	0
Min	-
Max	-
PDO mapping	-
Description	The parameter value comprises two pieces of data:

Bits 31-24	Bits 23-16	Bits 15-0
0	Node ID	Heartbeat time

The heartbeat time defines the time interval within which a heartbeat message is expected from the node whose node ID is stored in bits 23-16. A heartbeat time of 0x00 means that no heartbeat message is expected.

4.2.4.4.16 Object 0x1017: Heartbeat Producer Time

Object code	VAR
Data type	UNSIGNED16
Attribute	rw
Default	0
Min	0
Max	65535
PDO mapping	-
Description	This parameter defines the time interval in ms for sending the heartbeat message. If the parameter value is unequal to 0, the encoTRive operates as heartbeat producer. Parameter value 0 means that no heartbeat messages are sent.

4.2.4.4.17 Object 0x1018: Identity object device information

Object code	ARRAY	
Description	This parameter contains general information on the encoTRive.	
	Subindex 0: Number of entries	
	Data type	UNSIGNED8
	Attribute	ro
	Default	4
	Min	-
	Max	-
	PDO mapping	-
	Description	Contains the largest array subindex
	Subindex 1: Manufacturer's identification	
	Data type	UNSIGNED32
	Attribute	ro
	Default	0
	Min	-
	Max	-
	PDO mapping	-
	Description	Manufacturer's unique identification
	Subindex 2: Product code	
	Data type	UNSIGNED32
	Attribute	ro
	Default	736018
	Min	-
	Max	-
	PDO mapping	-
	Description	Identifies the device version
	Subindex 3: Revision number	
	Data type	UNSIGNED32
	Attribute	ro
	Default	0x00040005
	Min	-
	Max	-
	PDO mapping	-
	Description	The revision number consists of ***Major Revision Number*** (bits 31-16) and ***Minor Revision Number*** (bits 15-0). The Major Revision Number is incremented when an extension is made to the CANopen behavior of encoTRive, e.g. new objects.

Subindex 4: Serial number	
Data type	UNSIGNED32
Attribute	ro
Default	1
Min	-
Max	-
PDO mapping	-
Description	Serial number of the device

4.2.4.4.18 Objects 0x1400-0x1405: Communication, Receive PDOs

The communication parameters for Receive PDOs (***RPDO***s) define which PDOs are used, which COB-IDs are used for this purpose, and how received process data is processed. The parameter structure is identical for all six parameters.

- Object 0x1400: Communication parameter for RPDO1
- Object 0x1401: Communication parameter for RPDO2
- Object 0x1402: Communication parameter for RPDO3
- Object 0x1403: Communication parameter for RPDO4
- Object 0x1404: Communication parameter for RPDO5
- Object 0x1405: Communication parameter for RPDO6

Object code	RECORD	
Description	Each of these parameters configures a Receive PDO	
	Subindex 0: Number of entries	
	Data type	UNSIGNED8
	Attribute	ro
	Default	2
	Min	-
	Max	-
	PDO mapping	-
	Description	Contains the largest array subindex
	Subindex 1: COB-ID	
	Data type	UNSIGNED32
	Attribute	rw
	Default	Index 0x1400: Node ID + 0x200 Index 0x1401: Node ID + 0x300 Index 0x1402: Node ID + 0x400 Index 0x1403: Node ID + 0x500 Index 0x1404: Node ID + 0x8000 0480 Index 0x1405: Node ID + 0x8000 0380
	Min	-
	Max	-
	PDO mapping	-
	Description	Defines whether the relevant RPDO is used and defines its COB-ID.

Bit	Meaning
31	0 : PDO is valid 1 : PDO is not used
30	0 : Reacts to RTR 1: No reaction to RTR
29	0: 11-bit identifier (CAN 2.0A) 1: 29-bit identifier (CAN 2.0B)
28-0	COB-ID

Subindex 2: Transmission type

Data type	UNSIGNED8
Attribute	rw
Default	255
Min	0
Max	255
PDO mapping	-
Description	Defines how received PDO data is to be processed by the encoTRive.

Table 23: Transmission type (RPDO)

Value	Meaning
0-240	Synchronous: RPDO is evaluated immediately after receipt of the next SYNC message.
255	RPDO is evaluated immediately after receipt.

4.2.4.4.19 Objects 0x1600-0x1605: Mapping, Receive PDOs

These parameters define which contents are transported in RPDOs.

- Object 0x1600: Mapping parameter for RPDO1
- Object 0x1601: Mapping parameter for RPDO2
- Object 0x1602: Mapping parameter for RPDO3
- Object 0x1603: Mapping parameter for RPDO4
- Object 0x1604: Mapping parameter for RPDO5
- Object 0x1605: Mapping parameter for RPDO6

Object code	ARRAY
Description	Each of these parameters configures the mapping for a Receive PDO.

Subindex 0: Number of entries

Data type	UNSIGNED8
Attribute	rw
Default	Object 0x1600 - 0x1605 : 0
Min	0
Max	8
PDO mapping	-
Description	Actual number of mapped objects. Value 0 means: RPDO deactivated.

If mapping entries have to be added, the new entries must be defined first of all. Only then the number of elements may be changed.

Subindex 1: First mapped object

Data type	UNSIGNED32
Attribute	rw
Default	Object 0x1600: 0x0000 0000 Object 0x1601: 0x0000 0000 Object 0x1602: 0x0000 0000 Object 0x1603: 0x0000 0000 Object 0x1604: 0x0000 0000 Object 0x1605: 0x0000 0000
Min	-
Max	-
PDO mapping	-
Example	Object 0x1601: 0x6040 0010
Description	Specifies the index, the subindex and the width of the relevant RPDO sub-area.

Bits 31-16	Bits 15-8	Bits 7-0
Index	Subindex	Length in bits

In the mapping of the RPDO2 in the above example, the first two bytes (length = 0x10 = 16 bits) represent byte 3 and 4 (subindex = 0x00) and byte 5 to 8 (index = 0x6040 = control word). Mapping of the control word in RPDO1 is not strictly necessary.

Subindex 2: Second mapped object

Data type	UNSIGNED32
Attribute	rw
Default	Object 0x1600: 0x0000 0000 Object 0x1601: 0x0000 0000 Object 0x1602: 0x0000 0000 Object 0x1603: 0x0000 0000 Object 0x1604: 0x0000 0000 Object 0x1605: 0x0000 0000
Min	-
Max	-
PDO mapping	-
Example	Object 0x1601: 0x6081 0020
Description	Specifies the index, the subindex and the width of the relevant RPDO sub-area.

Bits 31-16	**Bits 15-8**	**Bits 7-0**
Index	Subindex	Length in bits

In the mapping of the RPDO2 in the above example, the first two bytes (length = 0x20 = 32 bits) represent byte 3 and 4 (subindex = 0x00) and byte 5 to 8 (index = 0x6081 = speed)

Example: RPDO mapping

A PDO message enables the transmission of up to 8 bytes of data from areas of the object directory which support a PDO mapping.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
RPDO1:	Maximum current (0x6073; 16 bit)		Nominal operating mode (0x6060; 16 bit)		Target position (0x607A; 32bit)			
RPDO2:	Control word (0x6040; 16 bit)		Speed (0x6081; 32 bit)				-	
RPDO3:	Acceleration (0x6083; 32bit)				Deceleration (0x6084; 32 bit)			
RPDO4:	-							
RPDO5:	Object 0x1404 subindex 1 – default setting "PDO not used". In the event of activation, note the classification of the function code.							
RPDO6:	Object 0x1405 subindex 1 – default setting "PDO not used ". In the event of activation, note the classification of the function code.							

4.2.4.4.20 Objects 0x1800-0x1805: Communication, Transmit PDOs

The communication parameters for the Transmit PDOs (***TPDO***s) define which PDOs are used, which COB-IDs are used for this purpose, and when process data is sent. The parameter structure is identical for all six parameters.

- Object 0x1800: Communication parameter for TPDO1
- Object 0x1801: Communication parameter for TPDO2
- Object 0x1802: Communication parameter for TPDO3
- Object 0x1803: Communication parameter for TPDO4
- Object 0x1804: Communication parameter for TPDO5
- Object 0x1805: Communication parameter for TPDO6

Object code	RECORD	
Description	Each of these parameters configures a TPDO	
	Subindex 0: Number of entries	
	Data type	UNSIGNED8
	Attribute	ro
	Default	5
	Min	-
	Max	-
	PDO mapping	-
	Description	Contains the largest array subindex
	Subindex 1: COB-ID	
	Data type	UNSIGNED32
	Attribute	rw
	Default	Object 0x1800: Node ID + 0x180 Object 0x1801: Node ID + 0x280 Object 0x1802: Node ID + 0x380 Object 0x1803: Node ID + 0x480 Object 0x1804: Node ID + 0x8000 0500 Object 0x1805: Node ID + 0x8000 0400
	Min	-
	Max	-
	PDO mapping	-
	Description	Defines whether the relevant TPDO is used and defines its COB-ID.

Bit	Meaning
31	0 : PDO is valid 1 : PDO is not used
30	0 : Reacts to RTR 1: No reaction to RTR
29	0: 11-bit identifier (CAN 2.0A) 1: 29-bit identifier (CAN 2.0B)
28-0	COB-ID

Subindex 2: Transmission type

Data type	UNSIGNED8
Attribute	rw
Default	255
Min	0
Max	255
PDO mapping	-
Description	Defines when PDO data is to be sent by the encoTRive.

Table 24: Transmission type (TPDO)

Value	Meaning
0	***Synchronous***, but ***acyclical***: The TPDO is sent immediately after receipt of the next SYNC message, but only if an event has occurred before the SYNC message.
1-240	***Synchronous*** and ***cyclical***: The TPDO is always sent after the number of SYNC messages specified in the value.
252	***RTR-linked:*** The TPDO is only sent if requested by a corresponding Remote Transmission Request. The transmission data is only updated upon receipt of the SYNC message.
253	***RTR-linked:*** The TPDO is only sent if requested by a corresponding Remote Transmission Request. The transmission data is updated upon receipt of the Remote Transmission Request.
254	***Asynchronous:*** The TPDO is sent event-controlled, and the triggering event is defined manufacturer-specifically.
255	***Asynchronous:*** The TPDO is sent event-controlled, and the triggering event is defined profile-specifically.

Subindex 3: Inhibit time

Data type	UNSIGNED16
Attribute	rw
Default	500
Min	0
Max	65535
PDO mapping	-
Description	Inhibit time in 0.1 ms steps. The next PDO with the same COB-ID may only be sent after expiry of this time. Parameter value 1000 = 100 ms inhibit time.

	Subindex 5: Event Timer
Data type	UNSIGNED16
Attribute	rw
Default	0
Min	0
Max	65535
PDO mapping	-
Description	Defines a time interval in 0.1 ms steps whose expiry is seen as the triggering event for sending the PDO. This event operates in addition to other events which trigger transmission. Parameter value 0 deactivates this mechanism.

4.2.4.4.21 Objects 0x1A00-0x1A05: Mapping, Transmit PDOs

These parameters define which contents are transported in TPDOs.

- Object 0x1A00: Mapping parameter for TPDO1
- Object 0x1A01: Mapping parameter for TPDO2
- Object 0x1A02: Mapping parameter for TPDO3
- Object 0x1A03: Mapping parameter for TPDO4
- Object 0x1A04: Mapping parameter for TPDO5
- Object 0x1A05: Mapping parameter for TPDO6

Object code	ARRAY
Description	Each of these parameters configures the mapping for a Transmit PDO

Subindex 0: Number of entries

Data type	UNSIGNED8
Attribute	rw
Default	Object 0x1A00 - 0x1A05 : 0
Min	0
Max	8
PDO mapping	-
Description	Actual number of mapped objects. Value 0 means: TPDO deactivated.

If mapping entries have to be added, these must be defined first of all. Only then the number of elements may be changed.

Subindex 1: First mapped object

Data type	UNSIGNED32
Attribute	rw
Default	Object 0x1A00: 0x0000 0000 Object 0x1A01: 0x0000 0000 Object 0x1A02: 0x0000 0000 Object 0x1A03: 0x0000 0000 Object 0x1A04: 0x0000 0000 Object 0x1A05: 0x0000 0000
Min	-
Max	-
PDO mapping	-
Example	Object 0x1A00: 0x6041 0010
Description	Specifies the index, the subindex and the width of the relevant TPDO sub-area.

Bits 31-16	Bits 15-8	Bits 7-0
Index	Subindex	Length in bits

In the mapping of the TPDO1 in the above example, the first two bytes (length = 0x10 = 16 bits) represent byte 3 and 4 (subindex = 0x00) and byte 5 to 8 (index = 0x6041 = status word). Mapping of the status word in TPDO1 is not strictly necessary.

Subindex 2: Second mapped object

Data type	UNSIGNED32
Attribute	rw
Default	Object 0x1A00: 0x0000 0000 Object 0x1A01: 0x0000 0000 Object 0x1A02: 0x0000 0000 Object 0x1A03: 0x0000 0000 Object 0x1A04: 0x0000 0000 Object 0x1A05: 0x0000 0000
Min	-
Max	-
PDO mapping	-
Example	Object 0x1A00: 0x6064 0020
Description	Specifies the index, the subindex and the width of the relevant RPDO sub-area.

Bits 31-16	**Bits 15-8**	**Bits 7-0**
Index	Subindex	Length in bits

In the mapping of the TPDO1
in the above example, the first two bytes (length = 0x20 = 32 bits) represent byte 3 and 4 (subindex = 0x00) and byte 5 to 8 (index = 0x6064 = actual position value)

Example: TPDO mapping

A PDO message enables the transmission of up to 8 bytes of data from areas of the object directory which support a PDO mapping.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
TPDO1:	Status word (0x6041; 16 bit)		Actual position value (0x6064; 32 bit)				-	
TPDO2:	Actual speed (0x606C; 32 bit)				Actual value of current (0x6078; 16 bit)		-	
TPDO3:	CPU temp. (0x2E02 [2];16 bit)		-					
TPDO4:	-							
TPDO5:	Object 0x1804 subindex 1 – default setting "PDO not used". In the event of activation, note the classification of the function code.							
TPDO6:	Object 0x1805 subindex 1 – default setting "PDO not used". In the event of activation, note the classification of the function code.							

4.2.4.5 Manufacturer-specific objects

4.2.4.5.1 Object 0x2E02: Temperature / Bus address / Baud rate

Object code	RECORD	
Description	Temperature sensors and the DIP switches for Node ID and Baud Rate can be read out with these parameters.	
	Subindex 0: Number of entries	
	Data type	UNSIGNED8
	Attribute	ro
	Default	8
	Min	-
	Max	-
	PDO mapping	yes
	Description	Number of following entries
	Subindex 1: Self-test result	
	Data type	UNSIGNED16
	Attribute	ro
	Default	0
	Min	-
	Max	-
	PDO mapping	yes
	Description	Error code during execution of self-test after power supply ON. If no error has occurred, the result is 0. Otherwise an EMCY message is issued (see chapter "6.2 EMCY error information" from page 248), and the drive is switched off. Status word = "Not ready for operation".
	Subindex 2: Temperature sensor 1	
	Data type	UNSIGNED16
	Attribute	ro
	Default	0
	Min	-
	Max	-
	PDO mapping	yes
	Description	The sensor value specifies the temperature of the CPU board in °C. If the sensor fails, the EMCY message "4280" is issued (see chapter "6.2 EMCY error information" from page 248), and the drive is switched off.

Subindex 3: Temperature sensor 2	
Data type	UNSIGNED16
Attribute	ro
Default	0
Min	-
Max	-
PDO mapping	yes
Description	The sensor value specifies the temperature of the power board in °C. If the sensor fails, the EMCY message "4380" is issued (see chapter "6.2 EMCY error information" from page 248), and the drive is switched off.

Subindex 4: Address switch and baud rate	
Data type	UNSIGNED16
Attribute	ro
Default	0
Min	-
Max	-
PDO mapping	yes
Description	Contains the actual value of the HEX switches for baud rate and node address. Byte n = Node address Byte n+1 = Baud rate

Subindex 6: Actual torque generating current limitation	
Data type	UNSIGNED16
Attribute	ro
Default	0
Min	-
Max	-
PDO mapping	yes
Description	Unit: Rated motor current /1000 The roots of the square sum of object 6073 "Maximum current" and object 2F02 "Idle current limitation" SubA must not exceed the drive's physical current limitation. The idle current is limited first of all, then the torque generating current.

Subindex 7: Actual idle current limitation	
Data type	UNSIGNED16
Attribute	ro
Default	0
Min	-
Max	-
PDO mapping	yes
Description	Unit: Rated motor current/1000. For description see Subindex 6.

Subindex 8: Time span of the last position ramp calculation in PWM cycles

Data type	UNSIGNED16
Attribute	ro
Default	0
Min	-
Max	-
PDO mapping	yes
Description	-

4.2.4.5.2 Object 0x2F02: Temperature threshold values

Object code	RECORD	
Description	Temperature threshold values can be set with this parameter; exceeding of these values will result in a warning or cut-out of the drive.	
	Subindex 0: Number of entries	
	Data type	UNSIGNED8
	Attribute	ro
	Default	10
	Min	-
	Max	-
	PDO mapping	-
	Description	Number of following entries
	Subindex 1: Warning temperature	
	Data type	UNSIGNED16
	Attribute	rw
	Default	75
	Min	-
	Max	-
	PDO mapping	yes
	Description	The entry defines the temperature in °C. Exceeding of a temperature sensor will result in a corresponding EMCY message (see chapter "6.2 EMCY error information" from page 248).
	Subindex 2: Cut-out temperature	
	Data type	UNSIGNED16
	Attribute	rw
	Default	90
	Min	-
	Max	-
	PDO mapping	yes
	Description	The entry defines the temperature in °C. Exceeding of a temperature sensor will result in a corresponding EMCY message (see chapter "6.2 EMCY error information" from page 248). In addition, if this temperature is exceeded the drive goes into "Error" status and executes the action defined in object 0x605E.

Subindex A: Idle current limitation	
Data type	UNSIGNED16
Attribute	rw
Default	1250 (125%)
Min	-
Max	-
PDO mapping	yes
Description	Unit: Rated motor current/1000. In order to permit high speeds or low DC bus voltages, the drive can cancel out the rotor field in the stator windings by generating an idle current. This is the limit for the idle current in 1/1000 of the rated current. 0 deactivates the rotor field cancellation. This cancellation does not weaken the field, nor does it reduce the torque.

4.2.4.5.3 Object 0x2F04: Calibration values

Object code	RECORD	
Description	Range calibration for current and voltage measurement	
	Subindex 0: Number of entries	
	Data type	UNSIGNED8
	Attribute	ro
	Default	11
	Min	-
	Max	-
	PDO mapping	-
	Description	Number of following entries
	Subindex 1: Stator current [mA]	
	Data type	UNSIGNED32
	Attribute	rw
	Default	24229
	Min	0
	Max	2^{32}-1
	PDO mapping	Yes
	Description	The entry defines the range limit in mA, which is used for digital/analog conversion of the current value.
	Subindex 2: Minimum bus voltage [mV]	
	Data type	UNSIGNED32
	Attribute	rw
	Default	12000
	Min	0
	Max	2^{32}-1
	PDO mapping	Yes
	Description	The bus voltage is the supply voltage for the power output stages. The total forward or open control loop gain of the current or torque regulation is proportional to this voltage. In order to obtain a constant reaction time for the closed control loop, the gain of the PI controller increases as the bus voltage reduces. When the bus voltage falls below the minimum value, the controller gain reaches its maximum. The total gain therefore reduces, and the reaction time of the closed control loop increases. The minimum bus voltage may fall below 10% of the nominal value, caused by the limit value measurement and the calculation accuracy. 25% is recommended. If this value has been changed, or before a new drive is commissioned, the bus voltage sensor must be checked and is reset with the help of the auto-calibration function.

Subindex 3: Rated bus voltage [mV]	
Data type	UNSIGNED32
Attribute	rw
Default	24000
Min	0
Max	2^{32}-1
PDO mapping	Yes
Description	If the supply voltage of the power output stages lies below this value, the drive can no longer reach the rated speed and rated power. If this value has been changed, or before a new drive is commissioned, the bus voltage sensor must be checked and is reset with the help of the auto-calibration function.

Subindex 4: Maximum bus voltage [mV]	
Data type	UNSIGNED32
Attribute	rw
Default	57000
Min	0
Max	2^{32}-1
PDO mapping	Yes
Description	If the supply voltage of the power output stages exceeds this value, the load resistors are activated. Their purpose is to destroy the superfluous electrical energy generated during braking. Otherwise the bus capacitors would become so highly charged, that the overvoltage would destroy the power supply components.

Subindex 5: Magnetic brake rated voltage [mV]	
Data type	UNSIGNED32
Attribute	rw
Default	24000
Min	0
Max	2^{32}-1
PDO mapping	Yes
Description	This is the power supply that is required to release the magnetic brake. When the drive is switched off, the motor axis is blocked by the magnetic brake. If a magnetic brake is connected to a PWM-regulated brake output, the corresponding rated voltage must be entered here.

Subindex 6: Ballast voltage [mV]	
Data type	UNSIGNED32
Attribute	rw
Default	3000
Min	0
Max	2^{32}-1
PDO mapping	Yes
Description	This is the surplus voltage above the max. bus voltage which is required to switch the ballast resistors on completely. A value of 4000 mV prevents switching peaks in the supply lines.

Subindex 7: Stator inductivity [µH]	
Data type	UNSIGNED32
Attribute	rw
Default	406
Min	0
Max	2^{32}-1
PDO mapping	Yes
Description	The proportional gains of the PI current controller are set by the DC voltage, divided according to stator inductivity, in order to maintain the necessary switch-off frequency of the closed control loop. Depending on the motor type, this inductivity can fall to 1/3 if the stator current is increased from 0 to the nominal value. In order to prevent gains that are too high and cause vibrations, the stator inductivity should be measured at maximum current.

4.2.4.6 Objects of the DSP 402 device profile

4.2.4.6.1 Object 0x6007: Abort Connection Code

Object code	VAR
Data type	INTEGER16
Attribute	rw
Default	0
Min	-32768
Max	32767
PDO mapping	yes
Description	This parameter specifies the action to be taken by encoTRive when the network connection is interrupted. The following values are defined: 0: No action 1: Go into "Fault" status 2: Execute "Disable Voltage" command, see Table 13 page 162 3: Execute "Quick Stop" command, see Table 13 page 162

4.2.4.6.2 Object 0x603F: Error Code

Object code	VAR
Data type	UNSIGNED16
Attribute	ro
Default	0
Min	0
Max	65535
PDO mapping	yes
Description	This parameter stores the last occurring error and corresponds to the low-value 16 bits of object 0x1003.

4.2.4.6.3 Object 0x6040: Control word

Object code	VAR
Data type	UNSIGNED16
Attribute	rww
Default	0
Min	0
Max	65535
PDO mapping	yes
Description	Commands are transmitted to the drive in this object, see sections 4.2.3.2, 4.2.3.3.1 and 4.2.3.4.1.

4.2.4.6.4 Object 0x6041: Status word

Object code	VAR
Data type	UNSIGNED16
Attribute	ro
Default	0
Min	0
Max	65535
PDO mapping	yes
Description	The drive transmits status information in this object, see sections 4.2.3.2, 4.2.3.3.2 and 4.2.3.4.2.

4.2.4.6.5 Object 0x604D: Pole pair number

Object code	VAR
Data type	UNSIGNED8
Attribute	ro
Default	4
Min	2
Max	-
PDO mapping	yes
Description	Contains the number of pole pairs of the motor used.

4.2.4.6.6 Object 0x605A: Quick stop behavior

(Quick Stop Option Code)

Object code	VAR
Data type	INTEGER16
Attribute	rw
Default	2
Min	-32768
Max	32767
PDO mapping	-
Description	The reaction after execution of a quick stop can be defined with this parameter.

The following values are supported by the encoTRive:

Table 25: Values for Quick Stop Option Code

Value	Meaning
0	Switches drive off. After coming to a stop, the drive goes into "Switch-on inhibit" status.
1	Decelerates drive to a stop with current deceleration. The drive then goes into "Switch-on inhibit" status.
2	Decelerates drive with deceleration for quick stop (0x6085). After coming to a stop, the drive goes into "Switch-on inhibit" status.
5	Decelerates with current deceleration. After coming to a stop the drive remains in "quick stop" status.
6	Decelerates with deceleration for quick stop (0x6085). After coming to a stop the drive remains in "Quick stop" status.

4.2.4.6.7 Object 0x605B: Shutdown behavior

(Shutdown Option Code)

Object code	VAR
Data type	INTEGER16
Attribute	rw
Default	0
Min	-32768
Max	32767
PDO mapping	-
Description	This parameter is used to define the reaction of the drive during the state transition *Ready for operation → Ready to switch on*.

The following values are supported by the encoTRive:

Table 26: Values for Shutdown Option Code

Value	Meaning
0	Switches drive off.
1	Decelerates drive to a stop with current deceleration. Then switches drive off.

4.2.4.6.8 Object 0x605C: Disable Operation behavior

(Disable Operation Option Code)

Object code	VAR
Data type	INTEGER16
Attribute	rw
Default	1
Min	-32768
Max	32767
PDO mapping	-
Description	This parameter is used to define the reaction of the drive during the state transition *Ready for operation → Switched on*.

The following values are supported by the encoTRive:

Table 27: Values for Disable Operation Option Code

Value	Meaning
0	Switches drive off.
1	Decelerates drive to a stop with current deceleration. Then switches drive off.

4.2.4.6.9 Object 0x605D: Stop behavior

(Halt Option Code)

Object code	VAR
Data type	INTEGER16
Attribute	rw
Default	1
Min	-32768
Max	32767
PDO mapping	-
Description	This parameter is used to define the reaction of the drive when bit 8 = "Stop motor" is active in the control word. The following values are supported by the encoTRive:

Table 28: Values for Halt Option Code

Value	Meaning
0	Switches drive off.
1	Decelerates drive to a stop with current deceleration.
2	Decelerates drive with deceleration for Quick stop (0x6085).
3	Decelerates drive at current limit.
4	Decelerates drive at voltage limit.

4.2.4.6.10 Object 0x605E: Fault behavior

(Fault Reaction Option Code)

Object code	VAR
Data type	INTEGER16
Attribute	rw
Default	2
Min	-32768
Max	32767
PDO mapping	-
Description	This parameter is used to define the reaction of the drive in the event of a fault.

The following values are supported by the encoTRive:

Table 29: Values for Fault Reaction Option Code

Value	Meaning
0	Switches drive off.
1	Decelerates drive to a stop with current deceleration.
2	Decelerates drive with deceleration for Quick stop (0x6085).
3	Decelerates drive at current limit.
4	Decelerates drive at voltage limit.

4.2.4.6.11 Object 0x6060: Operating mode

Object code	VAR
Data type	INTEGER8
Attribute	wo
Default	0
Min	-128
Max	127
PDO mapping	yes
Description	This parameter is used to define the current operating mode.

The following values are supported by the encoTRive:

Table 30: Values for operating mode

Value	Meaning
1	Positioning ramp
3	Speed ramp

4.2.4.6.12 Object 0x6061: Operating mode display

Object code	VAR
Data type	INTEGER8
Attribute	ro
Default	-
Min	-128
Max	127
PDO mapping	yes
Description	This parameter is used to display the current operating mode (see Table 30, page 220).

4.2.4.6.13 Object 0x6062: Nominal position

Object code	VAR
Data type	INTEGER32
Attribute	ro
Default	0
Min	-
Max	-
PDO mapping	yes
Description	This object, in position units, contains the position to be moved to, resulting from the control algorithm.

4.2.4.6.14 Object 0x6064: Actual position

Object code	VAR
Data type	INTEGER32
Attribute	rw
Default	0
Min	-
Max	-
PDO mapping	yes
Description	This object displays the actual position value in user-defined position units, see chapter "4.2.3.5 Units" page 172.

4.2.4.6.15 Object 0x6065: Tracking error window

Object code	VAR
Data type	UNSIGNED32
Attribute	rw
Default	2^{32}-1 (tracking error monitoring switched off)
Min	0
Max	2^{32}-1
PDO mapping	yes
Description	This object defines the interval for permissible actual position values around the nominal position value (0x6062). If the actual position value is outside this interval, a tracking error exists. A parameter value of 2^{32}-1 deactivates the tracking error monitoring. A tracking error can occur • if the drive is blocked • if the target speed cannot be reached • if the control parameters are set unfavorably

4.2.4.6.16 Object 0x6066: Tracking error timeout

Object code	VAR
Data type	UNSIGNED16
Attribute	rw
Default	100
Min	0
Max	2^{16}-1
PDO mapping	yes
Description	The parameter value specifies the minimum time in 1 ms steps for which a tracking error must be present before it is displayed as such in the status word.

4.2.4.6.17 Object 0x6067: Position window

Object code	VAR
Data type	UNSIGNED32
Attribute	rw
Default	2^{32}-1 (position window monitoring switched off)
Min	0
Max	2^{32}-1
PDO mapping	yes
Description	This object defines a symmetrical interval around the target position (0x607A). If the actual position value is in this interval, the target position is considered to have been reached. A parameter value of 2^{32}-1 deactivates the position window monitoring.

4.2.4.6.18 Object 0x6068: Position window timeout

Object code	VAR
Data type	UNSIGNED16
Attribute	rw
Default	100
Min	0
Max	2^{16}-1
PDO mapping	yes
Description	The parameter value specifies the minimum time in 1 ms steps for which the actual position value must be present in the position window, before "Target reached" is indicated in the status word.

4.2.4.6.19 Object 0x6069: Measured speed

Object code	VAR
Data type	INTEGER32
Attribute	ro
Default	-
Min	-
Max	-
PDO mapping	yes
Description	The parameter contains the speed value measured on the speed encoder in speed increments.

4.2.4.6.20 Object 0x606B: Nominal speed

Object code	VAR
Data type	INTEGER32
Attribute	ro
Default	-
Min	-
Max	-
PDO mapping	yes
Description	This object contains the target speed in speed units, which results from the control algorithm.

4.2.4.6.21 Object 0x606C: Actual speed

Object code	VAR
Data type	INTEGER32
Attribute	ro
Default	-
Min	-
Max	-
PDO mapping	yes
Description	This object contains the current speed in speed units.

4.2.4.6.22 Object 0x6071: Target torque

Object code	VAR
Data type	INTEGER16
Attribute	rw
Default	0
Min	0
Max	10000
PDO mapping	yes
Description	This parameter contains the target torque. Unit: Thousandths of rated torque (0x6076).

4.2.4.6.23 Object 0x6072: Maximum torque

Object code	VAR
Data type	UNSIGNED16
Attribute	rw
Default	1250
Min	0
Max	10000
PDO mapping	yes
Description	This object contains the maximum permissible torque of the motor. Unit: Thousandths of rated torque. The default value 1250 corresponds to 125 % of the rated torque (0x6076).

4.2.4.6.24 Object 0x6073: Maximum current

Object code	VAR
Data type	UNSIGNED16
Attribute	rw
Default	1250
Min	0
Max	10000
PDO mapping	yes
Description	This object contains the maximum permissible motor current. Unit: Thousandths of rated current (0x6075).

4.2.4.6.25 Object 0x6074: Nominal torque

Object code	VAR
Data type	INTEGER16
Attribute	ro
Default	0
Min	-
Max	-
PDO mapping	yes
Description	This object contains the torque which results as output parameter of the torque limitation. Unit: Thousandths of rated torque (0x6076).

4.2.4.6.26 Object 0x6075: Rated motor current

Object code	VAR
Data type	UNSIGNED32
Attribute	rw
Default	7600
Min	-
Max	-
PDO mapping	yes
Description	This object specifies the rated motor current in mA.

4.2.4.6.27 Object 0x6076: Rated motor torque

Object code	VAR
Data type	UNSIGNED32
Attribute	rw
Default	630
Min	0
Max	2^{32}-1
PDO mapping	yes
Description	This object contains the rated torque of the motor in mNm.

4.2.4.6.28 Object 0x6077: Actual torque value

Object code	VAR
Data type	INTEGER16
Attribute	ro
Default	0
Min	-
Max	-
PDO mapping	yes
Description	This object contains the current torque of the motor. Unit: Thousandths of rated torque (0x6076)

4.2.4.6.29 Object 0x6078: Actual value of current

Object code	VAR
Data type	INTEGER16
Attribute	ro
Default	0
Min	-
Max	-
PDO mapping	yes
Description	This object contains the actual current value of the motor. Unit: Thousandths of rated current (0x6075)

4.2.4.6.30 Object 0x6079: Voltage at DC voltage intermediate circuit

Object code	VAR
Data type	UNSIGNED32
Attribute	ro
Default	0
Min	-
Max	-
PDO mapping	yes
Description	This object contains the current voltage of the DC voltage intermediate circuit in mV.

4.2.4.6.31 Object 0x607A: Target position

Object code	VAR
Data type	INTEGER32
Attribute	rww
Default	0
Min	-2^{31}
Max	$2^{31}-1$
PDO mapping	yes
Description	This object contains the target position in position units, see chapter "4.2.3.5 Units" page 172. Depending on the "absolute/relative" bit of the STW, the target position is interpreted as absolute or relative.

4.2.4.6.32 Object 0x607B: Position range

Object code	ARRAY	
Description	This parameter specifies the value range for the position value. When a position limit is reached or exceeded, the position value automatically jumps to the opposite end of the range. This behavior can be prevented with parameter 0x607D.	
	Subindex 0: Number of entries	
	Data type	UNSIGNED8
	Attribute	ro
	Default	2
	Min	-
	Max	-
	PDO mapping	-
	Description	Specifies the number of following entries
	Subindex 1: Lower position limit	
	Data type	INTEGER32
	Attribute	rw
	Default	-2^{31}
	Min	-2^{31}
	Max	$+2^{31}-1$
	PDO mapping	yes
	Description	Minimum position value
	Subindex 2: Upper position limit	
	Data type	INTEGER32
	Attribute	rw
	Default	2^{31}
	Min	-2^{31}
	Max	$+2^{31}-1$
	PDO mapping	yes
	Description	Maximum position value

4.2.4.6.33 Object 0x607D: Software position range

Object code	ARRAY	
Description	This parameter contains the absolute limits for position values. The values are specified in position units relative to the machine zero point. In the case of a new target position, a check is made to see whether this is located in the software position range.	
	Subindex 0: Number of entries	
	Data type	UNSIGNED8
	Attribute	ro
	Default	2
	Min	-
	Max	-
	PDO mapping	-
	Description	Specifies the number of following entries
	Subindex 1: Lower position limit	
	Data type	INTEGER32
	Attribute	rw
	Default	-2^{31}
	Min	-2^{31}
	Max	$+2^{31}-1$
	PDO mapping	yes
	Description	Minimum position value
	Subindex 2: Upper position limit	
	Data type	INTEGER32
	Attribute	rw
	Default	$2^{31}-1$
	Min	-2^{31}
	Max	$+2^{31}-1$
	PDO mapping	yes
	Description	Maximum position value

4.2.4.6.34 Object 0x607E: Direction reversal

Object code	VAR
Data type	UNSIGNED8
Attribute	rw
Default	0
Min	0
Max	255
PDO mapping	yes
Description	This parameter defines whether the direction of position and/or speed is to be reversed. If a direction reversal is defined, the values of actual position value and nominal position or actual speed value and nominal speed are multiplied by -1. The parameter value is interpreted as follows:

Table 31: Direction reversal

Bit 7	Bit 6	Bit 5-0
1 – Direction reversal at position 0 – No direction reversal at position	1 – Direction reversal at speed 0 – No direction reversal at speed	reserved

4.2.4.6.35 Object 0x607F: Maximum speed

Object code	VAR
Data type	UNSIGNED32
Attribute	rw
Default	8100
Min	0
Max	2^{32}-1
PDO mapping	yes
Description	This parameter defines the maximum speed in speed units during a positioning movement.

4.2.4.6.36 Object 0x6080: Maximum motor speed

Object code	VAR
Data type	UNSIGNED32
Attribute	rw
Default	9000
Min	1
Max	2^{32}-1
PDO mapping	yes
Description	This parameter specifies the maximum permitted speed of the motor shaft in rpm.

4.2.4.6.37 Object 0x6081: Speed

Object code	VAR
Data type	UNSIGNED32
Attribute	rw
Default	5000
Min	1
Max	2^{32}-1
PDO mapping	yes
Description	This parameter specifies the speed in speed units, which is to be reached at the end of an acceleration ramp, see chapter " 4.2.3.3 "Positioning ramp" operating mode " page 163.

4.2.4.6.38 Object 0x6083: Acceleration

Object code	VAR
Data type	UNSIGNED32
Attribute	rw
Default	500
Min	1
Max	2^{32}-1
PDO mapping	yes
Description	This parameter specifies in acceleration units the acceleration for the acceleration ramp, see chapter " 4.2.3.3 "Positioning ramp" operating mode " page 163.

4.2.4.6.39 Object 0x6084: Deceleration

Object code	VAR
Data type	UNSIGNED32
Attribute	rw
Default	1000
Min	1
Max	2^{32}-1
PDO mapping	yes
Description	This parameter specifies in acceleration units the deceleration used to slow down during a positioning movement, see chapter " 4.2.3.3 "Positioning ramp" operating mode " page 163.

4.2.4.6.40 Object 0x6085: Deceleration for quick stop

Object code	VAR
Data type	UNSIGNED32
Attribute	rw
Default	2000
Min	1
Max	2^{32}-1
PDO mapping	yes
Description	This parameter specifies in acceleration units the deceleration used to slow down to a quick stop, if the Quick Stop Option Code has a value of 2.

4.2.4.6.41 Object 0x6087: Torque increase

Object code	VAR
Data type	UNSIGNED32
Attribute	rw
Default	500000
Min	1
Max	2^{32}-1
PDO mapping	yes
Description	This parameter specifies the change in torque. Unit: Thousandths of rated torque per second.

4.2.4.6.42 Object 0x608F: Position encoder resolution

Object code	ARRAY	
Description	This parameter defines the ratio of position increments to motor revolutions, see chapter "4.2.3.5 Units" page 172.	
	Subindex 0: Number of entries	
	Data type	UNSIGNED8
	Attribute	ro
	Default	2
	Min	-
	Max	-
	PDO mapping	-
	Description	Specifies the number of following entries
	Subindex 1: Position increments	
	Data type	UNSIGNED32
	Attribute	ro
	Default	1024
	Min	-
	Max	-
	PDO mapping	-
	Description	Position increments
	Subindex 2: Motor revolutions	
	Data type	INTEGER32
	Attribute	ro
	Default	1
	Min	-
	Max	-
	PDO mapping	-
	Description	Motor revolutions

4.2.4.6.43 Object 0x6090: Speed encoder resolution

Object code	ARRAY	
Description	This parameter defines the ratio of speed increments per second to motor revolutions per second, see chapter "4.2.3.5 Units" page 172.	
	Subindex 0: Number of entries	
	Data type	UNSIGNED8
	Attribute	ro
	Default	2
	Min	-
	Max	-
	PDO mapping	-
	Description	Specifies the number of following entries
	Subindex 1: Speed increments	
	Data type	UNSIGNED32
	Attribute	ro
	Default	2^{31}
	Min	1
	Max	2^{32}-1
	PDO mapping	-
	Description	Speed increments per second
	Subindex 2: Motor revolutions	
	Data type	UNSIGNED32
	Attribute	ro
	Default	5000
	Min	1
	Max	2^{32}-1
	PDO mapping	-
	Description	Motor revolutions per second

4.2.4.6.44 Object 0x6093: Position factor

Object code	ARRAY	
Description	Position increments are obtained from user-defined position units (e.g. degrees) by multiplying with the position factor. The position unit used by the user is implicitly defined by the position factor. If necessary gear and feed are also taken into account, see chapter "4.2.3.5 Units" page 172. The position factor is the ratio between position increments and position units, defined by Formula 5.	
	Subindex 0: Number of entries	
	Data type	UNSIGNED8
	Attribute	ro
	Default	2
	Min	-
	Max	-
	PDO mapping	-
	Description	Specifies the number of following entries
	Subindex 1: Position increments (counter)	
	Data type	UNSIGNED32
	Attribute	rw
	Default	1024
	Min	1
	Max	2^{32}-1
	PDO mapping	yes
	Description	Position increments
	Subindex 2: Position increments (denominator)	
	Data type	UNSIGNED32
	Attribute	rw
	Default	1024
	Min	1
	Max	2^{32}-1
	PDO mapping	yes
	Description	Position units

4.2.4.6.45 Object 0x6094: Speed factor

Object code	ARRAY	
Description	Speed increments are obtained from user-defined speed units (e.g. rpm) by multiplying with the speed factor. The speed unit used by the user is implicitly defined by the speed factor. If necessary gear and feed are also taken into account, see chapter "4.2.3.5 Units" page 172. The speed factor is the ratio between speed increments per second and speed units, defined by Formula 8.	
	Subindex 0: Number of entries	
	Data type	UNSIGNED8
	Attribute	ro
	Default	2
	Min	-
	Max	-
	PDO mapping	-
	Description	Specifies the number of following entries
	Subindex 1: Speed increments (counter)	
	Data type	UNSIGNED32
	Attribute	rw
	Default	2^{31}
	Min	1
	Max	2^{32}-1
	PDO mapping	yes
	Description	Speed increments per second
	Subindex 2: Speed units (denominator)	
	Data type	UNSIGNED32
	Attribute	rw
	Default	300000
	Min	1
	Max	2^{32}-1
	PDO mapping	yes
	Description	Speed units

4.2.4.6.46 Object 0x6097: Acceleration factor

Object code	ARRAY	
Description	Speed increments per second are obtained from user-defined acceleration units (e.g. rpm/sec) by multiplying with the acceleration factor. The acceleration unit used by the user is implicitly defined by the acceleration factor. If necessary gear and feed are also taken into account, see chapter "4.2.3.5 Units" page 172. The acceleration factor is the ratio between speed increments per second and acceleration units, defined by Formula 10.	
	Subindex 0: Number of entries	
	Data type	UNSIGNED8
	Attribute	ro
	Default	2
	Min	-
	Max	-
	PDO mapping	-
	Description	Specifies the number of following entries
	Subindex 1: Acceleration increments (counter)	
	Data type	UNSIGNED32
	Attribute	rw
	Default	2^{31}
	Min	1
	Max	2^{32}-1
	PDO mapping	yes
	Description	Speed increments per second
	Subindex 2: Acceleration units (denominator)	
	Data type	UNSIGNED32
	Attribute	rw
	Default	300000
	Min	1
	Max	2^{32}-1
	PDO mapping	yes
	Description	Acceleration units

4.2.4.6.47 Object 0x60C5: Maximum acceleration

Object code	VAR
Data type	UNSIGNED32
Attribute	rw
Default	100000
Min	1
Max	2^{32}-1
PDO mapping	yes
Description	This parameter specifies the maximum permissible acceleration in user-defined acceleration units.

4.2.4.6.48 Object 0x60C6: Maximum deceleration

Object code	VAR
Data type	UNSIGNED32
Attribute	rw
Default	100000
Min	1
Max	2^{32}-1
PDO mapping	yes
Description	This parameter specifies the maximum permissible deceleration in user-defined acceleration units.

4.2.4.6.49 Object 0x60FD: Digital inputs

Object code	VAR
Data type	UNSIGNED32
Attribute	ro
Default	0
Min	-
Max	-
PDO mapping	yes
Description	This parameter specifies the current status of the digital inputs. encoTRive supports the following digital inputs: Bit 0: lower position value reached (1 = yes, 0 = no) Bit 1: upper position value reached (1 = yes, 0 = no)

4.2.4.6.50 Object 0x60FE: Digital outputs

Object code	ARRAY	
Description	This parameter enables control of the digital outputs provided by the encoTRive.	
	Subindex 0: Number of entries	
	Data type	UNSIGNED8
	Attribute	ro
	Default	2
	Min	-
	Max	-
	PDO mapping	-
	Description	Specifies the number of following entries
	Subindex 1: Output bits	
	Data type	UNSIGNED32
	Attribute	rw
	Default	0x0000 0000
	Min	-
	Max	-
	PDO mapping	yes
	Description	This element contains the defined output bits: Bit 0 = 0: stop brake closed
	Subindex 2: Bit mask	
	Data type	UNSIGNED32
	Attribute	rw
	Default	0x0000 0000
	Min	-
	Max	-
	PDO mapping	yes
	Description	Release / block output: Bit 0 = 1: output 0 is being used

4.2.4.6.51 Object 0x60FF: Target speed

Object code	VAR
Data type	INTEGER32
Attribute	rww
Default	0
Min	-2^{31}
Max	$2^{31} - 1$
PDO mapping	yes
Description	This parameter specifies the target speed in user-defined speed units in the "Speed ramp" operating mode.

4.2.4.6.52 Object 0x6402: Motor type

Object code	VAR
Data type	UNSIGNED16
Attribute	ro
Default	3
Min	-
Max	-
PDO mapping	-
Description	This parameter specifies the type of motor used. In the case of encoTRive, this is a synchronous motor with permanent magnet = value 3

4.2.4.6.53 Object 0x6502: Supported operating modes

Object code	VAR
Data type	UNSIGNED32
Attribute	ro
Default	0x0005
Min	-
Max	-
PDO mapping	-
Description	This parameter specifies the operating modes supported by the drive in bit code. The default value 5 (0000 0101) means that encoTRive supports the "Positioning ramp" (bit 0) and "Speed ramp" (bit 2) operating modes.

5 Example of a positioning movement with frame sequence

5.1 Prerequisites

The drive must

- be connected to the voltage supply
- be integrated into the CANopen network,
- and be able to communicate with the master via the CANopen network

5.2 Definitions

- Node address set on the drive = 0x70

This results in the COB-IDs

- 0x580 + 0x70 = 0x5F0, drive --> SDO client
- 0x600 + 0x70 = 0x670, SDO client --> drive

- Operating mode preset = positioning ramp, object 0x6060
- Position factor preset = 1024 for the denominator, object 0x6093 SUB2
- Position range preset = 0 for the lower position limit, object 0x607B SUB1
- Position range preset = 1.073.741.823 for the upper position limit, object 0x607B SUB2
- Software position range preset = 1 for the lower position limit, object 0x607D SUB1
- Software position range preset = 1.073.741.822 for the upper position limit, object 0x607D SUB2
- Speed preset = 4.350, object 0x6081
- Acceleration preset = 12.000, object 0x6083
- Deceleration preset = 12.000, object 0x6084
- Target position preset = 450.000, object 0x607A

For the frame structure and the meaning of the CCD function code, the information from the chapters "SDO (Service Data Object)" page 149 and "SDO message format" page 150 is relevant.

For the individual state transitions, the information from the chapters "DSP 402 state machine" page 159 and "Control word and status word" page 162 is relevant.

5.3 Frame sequence

Boot-up message after switching on

After switching on, the drive first of all registers with the boot-up message COB-ID 0x700 + Node ID 0x70 = 0x770, indicating to all other nodes that it is ready for communication. The drive is in NMT state **PRE-OPERATIONAL** NMT state and can be addressed by SDO messages.

	COB-ID
	11 bit
Rx_SDO	**0x770**

CCD	Index		SUB	Data			
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x00							

With status word 0x6041 bit 6, the drive indicates "Not ready for operation", `xxxx xxxx x1xx 0000 bin`.

	COB-ID
	11 bit
Tx_SDO	0x670
Rx_SDO	0x5F0

CCD	Index		SUB	Data			
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x40	0x41	0x60	0x00	0x00	0x00	0x00	0x00
0x4B	0x41	0x60	0x00	**0x40**	**0x00**	0x00	0x00

Start node

With the command **START-REMOTE-NODE** CCD = 0x01, the drive with node address 0x70 is put into **OPERATIONAL** NMT state. As response, the drive returns information on the Transmit PDOs for the communication.

	COB-ID
	11 bit
Tx_SDO	0x000
Rx_SDO	0x1F0
Rx_SDO	0x2F0
Rx_SDO	0x3F0
Rx_SDO	0x4F0

CCD	Index		SUB	Data			
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	**0x70**						

Define operating mode

The positioning ramp operating mode is set with object 0x6060 = 1.

	COB-ID
	11 bit
Tx_SDO	0x670
Rx_SDO	0x5F0

CCD	Index		SUB	Data			
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x2F	0x60	0x60	0x00	**0x01**	0x00	0x00	0x00
0x60	0x60	0x60	0x00	0x00	0x00	0x00	0x00

Define position factor

The position factor 1024 = 0x400 for the denominator is set with subindex 2 in object 0x6093. The default value is retained for the counter.

	COB-ID
	11 bit
Tx_SDO	0x670
Rx_SDO	0x5F0

CCD	Index		SUB	Data			
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x23	0x93	0x60	0x02	**0x00**	**0x04**	**0x00**	**0x00**
0x60	0x93	0x60	0x02	0x00	0x00	0x00	0x00

Define position ranges

The position value for the lower position limit = 0 and the upper position limit = 0x3FFF FFFF is set with subindices 1 and 2 in object 0x607B. Two frames are required for this purpose.

	COB-ID
	11 bit
Tx_SDO	0x670
Rx_SDO	0x5F0
Tx_SDO	0x670
Rx_SDO	0x5F0

CCD	Index		SUB	Data			
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x23	0x7B	0x60	0x01	**0x00**	**0x00**	**0x00**	**0x00**
0x60	0x7B	0x60	0x01	0x00	0x00	0x00	0x00
0x23	0x7B	0x60	0x02	**0xFF**	**0xFF**	**0xFF**	**0x3F**
0x60	0x7B	0x60	0x02	0x00	0x00	0x00	0x00

Define software position ranges

The software position value for the lower position limit = 1 and the upper position limit = 0x3FFF FFFE is set with subindices 1 and 2 in object 0x607D. Two frames are required for this purpose.

	COB-ID
	11 bit
Tx_SDO	0x670
Rx_SDO	0x5F0
Tx_SDO	0x670
Rx_SDO	0x5F0

CCD	Index		SUB	Data			
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x23	0x7D	0x60	0x01	**0x01**	**0x00**	**0x00**	**0x00**
0x60	0x7D	0x60	0x01	0x00	0x00	0x00	0x00
0x23	0x7D	0x60	0x02	**0xFE**	**0xFF**	**0xFF**	**0x3F**
0x60	0x7D	0x60	0x02	0x00	0x00	0x00	0x00

Define speed

The speed 4.350 = 0x10FE is set with object 0x6081.

	COB-ID
	11 bit
Tx_SDO	0x670
Rx_SDO	0x5F0

CCD	Index		SUB	Data			
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x23	0x81	0x60	0x00	**0xFE**	**0x10**	**0x00**	**0x00**
0x60	0x81	0x60	0x00	0x00	0x00	0x00	0x00

Define acceleration

The acceleration 12.000 = 0x2EE0 is set with object 0x6083.

	COB-ID
	11 bit
Tx_SDO	0x670
Rx_SDO	0x5F0

CCD	Index		SUB	Data			
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x23	0x83	0x60	0x00	**0xE0**	**0x2E**	**0x00**	**0x00**
0x60	0x83	0x60	0x00	0x00	0x00	0x00	0x00

Define deceleration

The deceleration 12.000 = 0x2EE0 is set with object 0x6084.

	COB-ID
	11 bit
Tx_SDO	0x670
Rx_SDO	0x5F0

CCD	Index		SUB	Data			
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x23	0x84	0x60	0x00	**0xE0**	**0x2E**	**0x00**	**0x00**
0x60	0x84	0x60	0x00	0x00	0x00	0x00	0x00

Switch motor output stage to 'ready to switch on'

The motor output stage of the drive can now be put into the 'Output stage ready to switch on' state with bits 1 and 2 in control word 0x6040, `xxxx xxxx xxxx x110 bin`.

	COB-ID
	11 bit
Tx_SDO	0x670
Rx_SDO	0x5F0

CCD	Index		SUB	Data			
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x2B	0x40	0x60	0x00	**0x06**	**0x00**	0x00	0x00
0x60	0x40	0x60	0x00	0x00	0x00	0x00	0x00

With status word 0x6041 bits 0 and 5, the drive indicates "Ready to switch on" and is now in "Quick stop active" status, `xxxx xxxx x01x 0001 bin`.

	COB-ID
	11 bit
Tx_SDO	0x670
Rx_SDO	0x5F0

CCD	Index		SUB	Data			
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x40	0x41	0x60	0x00	0x00	0x00	0x00	0x00
0x4B	0x41	0x60	0x00	**0x21**	**0x00**	0x00	0x00

Switch drive to 'ready for operation'

The sequence can now be continued in the state machine and the drive can be transferred to the "Ready for operation" status. The motor output stage is switched on with execution of the command.
In comparison with the previous bit pattern, the bit 0 must be set in control word 0x6040 in addition to bits 1 and 2, `xxxx xxxx xxxx 0111 bin`.

	COB-ID
	11 bit
Tx_SDO	0x670
Rx_SDO	0x5F0

CCD	Index		SUB	Data			
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x2B	0x40	0x60	0x00	**0x07**	**0x00**	0x00	0x00
0x60	0x40	0x60	0x00	0x00	0x00	0x00	0x00

With status word 0x6041 bit 1, the drive indicates "Readiness for operation",
`xxxx xxxx x01x 0011 bin`.

	COB-ID
	11 bit
Tx_SDO	0x670
Rx_SDO	0x5F0

CCD	Index		SUB	Data			
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x40	0x41	0x60	0x00	0x00	0x00	0x00	0x00
0x4B	0x41	0x60	0x00	**0x23**	**0x00**	0x00	0x00

Execute operating mode

The set operating mode must be executed in order to reach the next status in the state machine.
In comparison with the previous bit pattern, bit 3 must be set in control word 0x6040 in addition to bits 0, 1 and 2, `xxxx xxxx xxxx 1111 bin`.

	COB-ID
	11 bit
Tx_SDO	0x670
Rx_SDO	0x5F0

CCD	Index		SUB	Data			
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x2B	0x40	0x60	0x00	**0x0F**	**0x00**	0x00	0x00
0x60	0x40	0x60	0x00	0x00	0x00	0x00	0x00

With status word 0x6041 bits 2 and 4, the drive indicates "Operating mode activated" and "Voltage switched on", `xxxx xxxx x011 0111 bin`. The drive is in controlled state.

	COB-ID
	11 bit
Tx_SDO	0x670
Rx_SDO	0x5F0

CCD	Index		SUB	Data			
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x40	0x41	0x60	0x00	0x00	0x00	0x00	0x00
0x4B	0x41	0x60	0x00	**0x37**	**0x00**	0x00	0x00

Define target position

In order to start the positioning movement, the target position 450.000 = 0x6DDD0 must be communicated to the drive in object 0x607A.

	COB-ID
	11 bit
Tx_SDO	0x670
Rx_SDO	0x5F0

CCD	Index		SUB	Data			
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x23	0x7A	0x60	0x00	**0xD0**	**0xDD**	**0x06**	**0x00**
0x60	0x7A	0x60	0x00	0x00	0x00	0x00	0x00

Start positioning movement

The drive can now be started with control word 0x6040 bit 4. The previously set bits 0 to 4 remain set, `xxxx xxxx xxx1 1111 bin`.

	COB-ID
	11 bit
Tx_SDO	0x670
Rx_SDO	0x5F0

CCD	Index		SUB	Data			
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x2B	0x40	0x60	0x00	**0x1F**	**0x00**	0x00	0x00
0x60	0x40	0x60	0x00	0x00	0x00	0x00	0x00

With status word 0x6041 bit 12, the drive indicates "Target position acknowledged", `xxx1 xx1x x011 0111 bin`. This means that the drive has not yet reached the target.

	COB-ID
	11 bit
Tx_SDO	0x670
Rx_SDO	0x5F0

CCD	Index		SUB	Data			
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x40	0x41	0x60	0x00	0x00	0x00	0x00	0x00
0x4B	0x41	0x60	0x00	**0x37**	**0x12**	0x00	0x00

Positioning movement concluded

When the drive has reached the target, this is indicated in status word 0x6041 with bit 10, `xxx1 x1xx x011 0111`.

	COB-ID
	11 bit
Tx_SDO	0x670
Rx_SDO	0x5F0

CCD	Index		SUB	Data			
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x40	0x41	0x60	0x00	0x00	0x00	0x00	0x00
0x4B	0x41	0x60	0x00	**0x37**	**0x16**	0x00	0x00

6 Troubleshooting and diagnosis options

6.1 SDO error codes

Error code	Meaning
0x0503 0000	Toggle bit not changed.
0x0504 0000	SDO timeout.
0x0504 0001	Invalid/unknown command.
0x0504 0002	Invalid block length.
0x0504 0003	Invalid sequence number.
0x0504 0004	CRC error.
0x0504 0005	No memory available.
0x0601 0000	Unsupported access to an object.
0x0601 0001	Attempt to read an object that only allows write access.
0x0601 0002	Attempt to write to a write-protected object.
0x0602 0000	Object not in object directory.
0x0604 0041	Object cannot be mapped onto a PDO.
0x0604 0042	Total lengths of mapped objects exceed the PDO length.
0x0604 0043	General parameter incompatibility.
0x0604 0047	General internal incompatibility in the device.
0x0606 0000	Access failed due to hardware error.
0x0607 0010	Data type does not match / Parameter length does not match.
0x0607 0012	Data type does not match. Parameter too long.
0x0607 0013	Data type does not match. Parameter too short.
0x0609 0011	Subindex not present.
0x0609 0030	Parameter value range exceeded.
0x0609 0031	Parameter value too large.
0x0609 0032	Parameter value too small.
0x0609 0036	Maximum value is smaller than minimum value.
0x0800 0000	General error.
0x0800 0020	Data cannot be transferred / stored.
0x0800 0021	Data cannot be transferred / stored due to local control.
0x0800 0022	Data cannot be transferred / stored due to current device state.
0x0800 0023	Dynamic generation of object directory failed / No object directory present. For example, if the object directory is generated from a file and an error occurs when accessing the file.

Table 32: SDO error codes

6.2 EMCY error information

6.2.1 Error register, object 0x1001

The error register specifies the cause of the error in bit code. Several errors can also be displayed simultaneously by a set bit.
The more detailed error cause can be found in bits 0 - 15 in object 0x1003, see following pages. At the time of occurrence, an error is indicated by an EMCY message.

Bit	Meaning
0	General error
1	Current
2	Voltage
3	Temperature
4	Communication error (overflow, status error)
5	Device profile-specific
6	Reserved, always 0
7	Manufacturer-specific

Table 33: EMCY error register, object 0x1001

6.2.2 Error code, object 0x1003 (bits 0-15)

6.2.2.1 General information

The objeckt is from data typ ARRAY. This parameter stores the last 8 occurring errors as maximum. The "newest" error is stored in subindex 1. As the subindex increases, the error entries become older.

The error list in object 0x1003 can be deleted by three different methods:

1. Writing the value "0" to subindex 0 in object 1003
2. Executing the NMT service "Reset Node", command 0x81
3. Executing the NMT service "Reset Communication", command 0x82

Some error codes are also deleted automatically, for example EMCY messages which indicate bus errors. The reason for this is that the message can only be transmitted when the bus error has been eliminated.
The error code "0x0000" is transmitted for each EMCY message that is deleted. The result can be found in object 0x1003.

6.2.2.2 Profile-specific error code, CiA DSP 402

Error code	Meaning
0x0000	Error reset / no error
0x1000	General error
0x4210	Drive switched off due to overtemperature of the CPU board. When this temperature has fallen below the cut-out temperature again (object 0x2F02), the error can be acknowledged with STW.7. The drive then goes into the "Switch-on of output stage blocked" state in accordance with the state machine, see Figure 8 page 159.
0x4280	Drive switched off due to failure of the temperature sensor for the temperature on the CPU board. When the temperature sensor is working again, the error can be acknowledged with STW.7. The drive then goes into the "Switch-on of output stage blocked" state in accordance with the state machine, see Figure 8 page 159.
0x4290	Warning: Overtemperature on CPU board. When the temperature has fallen below the warning limit (object 0x2F02), the warning is deleted automatically.
0x4310	Drive switched off due to overtemperature on the power board. When this temperature has fallen below the cut-out temperature again (object 0x2F02), the error can be acknowledged with STW.7. The drive then goes into the "Switch-on of output stage blocked" state in accordance with the state machine, see Figure 8 page 159.
0x4380	Drive switched off due to failure of the temperature sensor for the supply board. When the temperature sensor is working again, the error can be acknowledged with STW.7. The drive then goes into the "Switch-on of output stage blocked" state in accordance with the state machine, see Figure 8 page 159.
0x4390	Warning: Overtemperature on power board. When the temperature has fallen below the warning limit (object 0x2F02), the warning is deleted automatically.

Profile-specific error code continued

Error code	Meaning
0x8100	Communication error
0x8110	Communication error: buffer overflow, incorrect state
0x8120	CAN error, passive mode
0x8130	Lifeguard or heartbeat error
0x8140	CAN active again after Bus off
0x8150	CAN COB-ID collision (multiple allocation of COB-ID)
0x8200	Protocol error
0x8210	PDO ignored due to length error
0x8220	PDO length exceeded

Table 34: Profile-specific EMCY error codes, object 1003

6.2.2.3 Manufacturer-specific error codes

Error code	Meaning
0x2300	A fatal error caused by overcurrent. Causes: • Short-circuit or interruption of a motor phase • Damaged power output stage • Overspeed • Sudden voltage drop • Current controller incorrectly set • No voltage or pole pair calibration If the EMCY message still occurs after calibration, the position encoder may be defective. The drive must be replaced.
4210	Error due to excess temperature of the CPU board. When the temperature has fallen below the cut-out threshold, the error code can be deleted with STW.7 "Reset error".
4280	Error due to drop-out of the temperature sensor on the CPU board. When the temperature sensor reacts again, the error code can be deleted with STW.7 "Reset error".
4290	Warning due to excess temperature of the CPU board. When the temperature has fallen below the warning threshold, the error code is deleted automatically.
4310	Error due to excess temperature of the power output stages. When the temperature has fallen below the cut-out threshold, the error code can be deleted with STW.7 "Reset error".
4380	Error due to drop-out of the temperature sensor for the power output stages. When the temperature sensor reacts again, the error code can be deleted with STW.7 "Reset error".
4390	Warning due to excess temperature of the power output stages, see object 0x2E02 subindex 3 on page 207. When the temperature has fallen below the warning threshold, the error code is deleted automatically.

Manufacturer-specific error codes continued

Error code	Meaning
6262 6263 6264 6265 6266	6262: Different object length 6263: Different object attributes 6264: Different data pointer 6265: Object not found 6266: Maximum mapping length of 8 bytes exceeded After restoration of the COM parameters from the non-volatile memory, the PDO mapping parameters are compared with the object directory. If the parameters of the mapping entries differ from those in the object directory, corresponding EMCY messages are issued. The reason may be a firmware update with changes in the object directory. In order to delete the error code, the default COM parameters must be loaded and the NMT service "Reset Node" executed.
6341	The position factor is too large, less than 2 position units fit in the maximum measuring section. Without the electronic gear, the maximum measuring section is 65536 revolutions.
6342	The electronic gear is deactivated, and the current measuring section is not a genuine divisor of the maximum measuring section. Without the electronic gear, the maximum measuring section is 65536 revolutions. The current measuring section in revolutions must be a power of 2 from $2^0 = 1$ to $2^{16} = 65536$. If this is not the case, the measuring section or the limits of the position range must be automatically set to the nearest possible values. The measuring section in position units corresponds to the distance between the measuring section limitations +1.
6343	An error has been triggered, because the requested position ramp would last longer than the maximum ramp duration. The limits for speed, acceleration and deceleration are too small. The error code is deleted when the error status has been deleted.
6345	Warning: The operation mode couldn't be changed, because the drive operation was enabled or enabling the drive operation failed, because the requested mode is not supported. This emergency is cleared when the drive operation is enabled successfully.
6348	Warning: An acceleration or deceleration limit was too low. It was increased to 27,9 rpm internally in order to avoid overflows when calculating the duration of the velocity ramps. This emergency is cleared if proper limits are set.
7300	An error has been triggered, because too many consecutive read errors have occurred on the rotary encoder of the position sensor. Safe operation of the drive is therefore no longer guaranteed. This may be caused by a damaged sensor or a loose magnet. The drive must be stopped immediately and replaced.
731F	An error has been triggered, because too many consecutive read errors have occurred on a reduction rotary encoder of the position sensor. Safe operation of the drive is therefore no longer guaranteed. This may be caused by a damaged gear, a damaged gear sensor, a loose rotor or magnet. The drive must be stopped immediately and replaced.

Manufacturer-specific error codes continued

Error code	Meaning
7320 7321 7322 7323 7324	The first error code means that the decoded reduction positions are different from the incrementally counted revolutions of the rotary encoder. The following error codes are warnings for the reduction rotary encoders 1 to 4 and mean that the adjustment is almost outside the permissible range. This may be caused by failure to perform a reduction rotary encoder calibration. If the error code still occurs after repeating and storing the calibration, the gear, a gear sensor or a magnet is damaged. The drive must be stopped immediately and replaced. The error codes are deleted when it has been possible to successfully complete the reduction rotary encoder calibration.
8110	A message was lost because the previously received messages could not be processed on time. If the message rate is too high, the baud rate can be decreased in order to prevent this problem. However, if TPDOs are blocked due to an overloaded CAN bus, the baud rate must be increased, or the TPDO message rate must be reduced. For transmission types 254 and 255 the inhibit time must be increased. Transmission types 0 to 239 must be increased, so that more SYNC messages are omitted between two TPDO messages. The error code is deleted immediately after the transmission.
8120	The node has been switched to "Error Passive Mode" state, as too many errors have occurred on the bus. This means that the node may only now send passive error frames. The red bus error LED flashes at 1.6 second intervals. Causes for this may be: • The bus lines are damaged, not connected or do not comply with the CANopen cabling requirements in accordance with "CiA DR-302-1". • The bus terminating resistors are incorrectly set or the baud rate is too high. • Two messages with the same identifier or COB-ID and the same RTR bit have collided on the bus. In this case, the arbitration (negotiation of media access) no longer works and the message with the lowest priority is rejected. If the data is not identical, both messages are rejected. New attempts would fail for the same reason, therefore the node is switched to "Error Passive Mode". Duplications of COB-IDs must be avoided at any rate, as they considerably reduce the performance and safety of the bus. To prevent duplications, the node addresses and also the configurable COB-IDs must therefore be checked. The error code is deleted immediately after the transmission.

Manufacturer-specific error codes continued

Error code	Meaning
8130	The remote requests for the node guarding or heartbeat messages for the heartbeat consuming have exceeded the preset time frame. The node has switched over from OPERATIONAL state to PRE-OPERATIONAL state and is executing the "Abort Connection Option Code" object 0x6070. The green bus status LED changes from static state to flashing mode. The red bus error LED flashes twice at 1.6 second intervals. Causes for this may be: • An error in the monitored node. • A bus overload, e.g. caused by event-triggered TPDOs and a too short inhibit time. • Incorrect configuration of the error control services. • The bus lines are damaged, not connected or do not comply with the CANopen cabling requirements in accordance with "CiA DR-302-1". • The bus terminating resistors are incorrectly set or the baud rate is too high. The error code is deleted immediately after the transmission.
8140	The node has just been incorporated back into the bus from "Bus-Off" state. A "Bus-Off" can be forced by a short-circuit between the CAN_L and CAN_H lines. In this case, the red bus error LED illuminates statically. The error code is not automatically deleted.
8150	An attempt was made to allocate a COB-ID which was already assigned to the same node. The receipt of these messages (COBs) was therefore blocked. This is possible for the configurable COB-IDs of the SYNC receipt and the PDOs. The following must be noted for assignment of the COB-IDs: • A node may not have two different types of COBs, which are configured with the same COB-ID. If both COBs are used for receipt and transmission, this will cause faults at any rate. In this case, it will no longer be possible to tell from the COB-ID of a message which type the COB is. The bidirectional use of a COB-ID must also be blocked. The reason for this is that a receive COB requires transmitted COBs from another node, so that both nodes can try to send COBs with the same ID at the same time. See also function code 8120. • EMCY message 8150 can only check duplicated COB-IDs at the same node. Duplicated IDs for transmit COBs at different nodes are detected with EMCY message 8120. For synchronization purposes it is even necessary to receive receive COBs with the same ID from different nodes. This refers to the receipt of time stamps or to RPDOs, if several drives are moved synchronously. Continued on next page.

Manufacturer-specific error codes continued

Error code	Meaning
8150	Error code description 8150 continued • EMCY message 8150 can also be indicated during a change of COB-ID by an SDO access, or while the stored COM parameters are being loaded in switch-on moment. An active, non-standard COB-ID for a PDO should not be stored. The reason for this is that during loading in switch-on moment, the 7 low-value bits are overwritten by the current node address. If a non-standard COB-ID is required, e.g. for an RPDO which must synchronize the drives, it must be stored with a standard value or with reset flag, bit 2^{31}. Before switching over to OPERATIONAL state, the necessary value must be assigned via SDO. • The EMCY message can also be indicated after a firmware update. In this case the default COM parameters must be loaded and the NMT service "Reset Node" command 0x81 executed. The error code is deleted immediately after the transmission.
8611	The drive has switched over to error state, because the tracking error of the position control function has exceeded the tracking error window for longer than the time specified in the tracking error timeout. Causes for this may be: • The supply voltage is not connected or is switched off. • The motor phase is not connected to the drive output. • The motor is overloaded or blocked. • The required speed is too high for the available power or bus supply voltage. The bus voltage has temporarily collapsed due to excessively long supply lines or a weak power supply. This can be checked with an oscilloscope at the connection terminals of the supply voltage. • The necessary acceleration or deceleration is too high for the torque limitation and mass inertia of the coupled load. The error reaction is defined by object 0x605E "Behavior in the event of error". Bit 13 "Tracking error" is set in the status word in addition to bit 3 "Error present". Both bits and also the EMCY message are deleted as soon as the error status has been cleared.
8612	Warning: The requested position ramp could not be executed, because the destination is outside of the position range limits or the software position limits. This emergency is cleared when the handshake is finished and the acknowledge is cleared again. In the incremental mode also the distance specified by target position must be less than the stroke. If the software position limits are not tighter than the position range limits, protruding distances are wrapped around the next range limit, instead of triggering this emergency.

Table 35: Manufacturer-specific EMCY error code, object 1003