KOLLMORGEN

CANopen Reference Guide

Rev B: July 2007

Part #: M-S2-021-11

Keep all product manuals as a product component during the life span of the servo amplifier.
Pass all product manuals to future users/owners of the servo amplifier.

NOTICE:

1.) This S200 Option requires the use of special user interface software called S200 OC Tools. This software can be installed using the included CD ROM. This device will ***not*** communicate with the standard S200 Tools software.

2.) Common Problems

a.) If all dip switches are set to ON (Toggled to the right), the unit enters a perpetual rest state and does not communicate. **Change dip switch settings**.

b.) Most error codes generated at the drive-level display an 'F20' fault code. Connect the GUI and look in the Status Screen for a 'b-level'- fault code with an explanation of the fault condition.

c.) When selecting a non-SFD motor, be certain to enter the 'motor poles' data.

d.) Always remember to Save the configuration to Non-volatile memory.

DANAHER MOTION

Helping you build a better machine, faster.

Record of Manual Revisions

Revision	Date	Description of Revision
~	07/2006	Initial Release
A	05/30/07	Removed inactive references
B	7/10/07	Major rework of document. Added Examples.

NOTICE:

Danaher Motion® is a registered trademark of the Danaher Corporation. Danaher Motion makes every attempt to ensure accuracy and reliability of the specifications in this publication. Specifications are subject to change without notice. Danaher Motion provides this information "AS IS" and disclaims all warranties, express or implied, including, but not limited to, implied warranties of merchantability and fitness for a particular purpose. It is the responsibility of the product user to determine the suitability of this product for a specific application.

Safety Symbols

WARNING ***Warnings alert users to potential physical danger or harm. Failure to follow warning notices could result in personal injury or death.***

CAUTION ***Cautions direct attention to general precautions which, if not followed, could result in personal injury and/or equipment damage.***

NOTE ***Notes highlight information critical to your understanding or use of the product.***

Safety

WARNING ***READ these instructions before connecting power. Damage can result from MISWIRING at the power terminals.***

DANGEROUS voltages are present on power input and motor output terminals.

Only qualified personnel are permitted to transport, assemble, commission, and maintain this equipment. Properly qualified personnel are persons who are familiar with the transport, assembly, installation, commissioning and operation of motors, and who have the appropriate qualifications for their jobs.

Read all available documentation before assembling and using. Incorrect handling of products described in this manual can result in injury and damage to people and/or machinery. Strictly adhere to the technical information regarding installation requirements.

- Keep all covers and cabinet doors shut during operation.
- Be aware that during operation, the product has electrically charged components and hot surfaces. Control and power cables can carry a high voltage, even when the motor is not rotating.
- Never disconnect or connect the product while the power source is energized.
- After removing the power source from the equipment, wait at least 5 minutes before touching or disconnecting sections of the equipment that normally carry electrical charges (e.g., capacitors, contacts, screw connections). To be safe, measure the electrical contact points to each other and to electrical safety earth with a meter before touching the equipment.

Table of Contents

1. INTRODUCTION

This manual describes the basic services and communication objects of the **CAN**open communication profile DS 301, which are used in the S200 Position Node product. It is provided as a subset to the *S200 Position Node with **CAN**open/DeviceNet* manual, which is located either on the Product Support Package CD-ROM shipped with your product or on the Danaher Motion website (http://www.danahermotion.com). The *S200 Position Node with **CAN**open/DeviceNet* manual contains information for hardware and software. Refer to the *S200 Position Node with **CAN**open/DeviceNet* manual for details.

Additional documentation:

CAN Application (CAL) for Industrial Applications	CiA e.V.
Draft Standards 301 (from Version 4.0), 402	CiA e.V.
CAN Specification Version 2.0	CiA e.V.
ISO 11898 ... Controller Area Network (CAN) for high-speed communication	
S200 Position Node with CANopen/DeviceNet Installation Manual	Danaher Motion

1. 1 SYSTEM REQUIREMENTS

— S200 servo amplifier with model number suffix –CNS (ex: S20360-CNS)
— Master station with a **CAN**open interface (eg., PC with **CAN**open card and Software)

1. 2 BASIC FEATURES

When working with the position controller that is integrated in S200 digital servo amplifiers, the following functions are available:

Setup And General Functions:
— Homing, set reference point
— Provision of a digital set point for speed and torque control
— Support of the following modes of the **CAN**open Profile DS402:
  » Profile position mode
  » Homing mode
  » Profile torque mode
  » Interpolated position mode
  » Profile velocity mode

Positioning Functions:
— Execution of a motion task from the motion block memory of the servo amplifier
— Execution of a direct motion task
— Absolute trajectory, ip-Mode

Data Transfer Functions:
— Transmit a motion task to the motion block memory of the servo amplifier
— A motion task consists of the following elements:
  » Position set point (absolute task) or path setpoint (relative task)
  » Speed set point
  » Acceleration time, braking time
  » Type of motion task (absolute/relative)
  » Number of a following task (with or without pause)
— Read a motion task from the motion block memory of the servo amplifier
— Read actual values
— Read the error register

— Read the status register
— Read/write control parameters

Transmission Rate And Procedure
— Bus connection and bus medium: **CAN**-standard ISO 11898 (**CAN** high-speed)
— Transmission rate: max. 1Mbit/s
» Possible settings for the servo amplifier:
125, 250, 500 (default), or 1000 kbit/s (kpps)

1. 3 *BUSOFF COMMUNICATION FAULTS*

The communication fault BUSOFF is directly monitored and signaled by Level 2 (**CAN** controller). This message may have various causes.

A few examples:
— Telegrams are transmitted, although there is no other **CAN** node connected
— **CAN** nodes have different transmission rates
— The bus cable is faulty
— Faulty cable termination causes reflections on the cable.

A BUSOFF is only signaled by the S200 if another **CAN** node is connected and at least one object was successfully transmitted to start off with. The BUSOFF condition is signaled by the error message F23. If the output stage is enabled for the execution of a motion task at the moment when this fault occurs, then the drive is decelerated to a stop, using the emergency stop ramp, and the output stage is disabled.

1. 4 *DRIVE ENABLE*

The S200 Position Node product is designed for general applications. There exists both hardware and software enable control. Hardware enable is via the first digital input – D!NP1 – which must be active to have power to the motor. The software enable control defaults to ‘Enabled’ and is controlled by the variable AENA (Auto Enable). While the default setting of software enable is appropriate for stand-alone systems, many field bus systems would rather default the software enable status to ‘DISABLED’. By setting AENA variable to ‘0’ and performing a ‘save to NVMEM’ the field bus system will power up in ‘software disabled’ mode.

1. 5 *GUI SUPPORT*

The S200OCTOOLS software interface can be used with S200 Position Nodes in CANopen applications. It can be especially useful when commissioning a system (setup, tuning, archiving, etc.) However, at the present time S200OCTOOLS does not have many support utilities for CANopen support. Under the Utilities menu there is a variable browser that allows the user to query the value of variables. After settinge them through CANopen they can be viewed to help verify system operation.

2. OVERVIEW

CANopen (Controller Area Network) is a development of the CIA (**CAN**-in-Automation) international user and manufacturers group and is standardized in the European standard EN 50325-4. **CAN**open is a very popular industrial communication system. **CAN**open was originally designed for motion-oriented machine control networks, such as handling systems. It is used in many more industries, such as medical equipment, off-road vehicles, maritime electronics, public transportation and building automation. **CAN**open products are certified by the CIA user organization, guaranteeing worldwide compatibility.

CANopen provides standardized communication objects for real-time data (Process Data Objects, PDO), configuration data (Service Data Objects, SDO), and special functions (Time Stamp, Sync message, and Emergency message) as well as network management data (Boot-up message, NMT message, and Error Control). All communication objects are listed in the Object Dictionary.

The Object Dictionary describes the complete functionality of a device by way of communication objects and is the interface between the communication interface and the application program. **CAN**open supports both cyclic and acyclic event driven communication. This makes it possible to reduce the bus loading to a minimum and maintain short reaction times. **CAN**open achieves high communication performance at low baud rates, thus reducing EMC problems and minimizing cable cost.

Generic device profiles describe just the interface of a single device. Application profiles describe all the device interfaces that are part of an application. Popular examples are the device profiles for I/O devices, electric drives, encoders or transducer and closed loop controllers.

2. 1 *COMMUNICATION PROFILE*

It is assumed that the basic operating functions of the communication profile are known and available as reference documentation.

The transmission method used is defined in ISO 11898 (Controller Area Network **CAN** for high-speed communication). The Layer-1/2 protocol (Physical Layer/Data Link Layer) that is implemented in all **CAN** modules provides, amongst other things, the requirements for data. Data transport or data request is made by means of a data telegram (Data Frame) with up to 8 bytes of user data, or by a data request telegram (Remote Frame). Communication objects (COBs) are labeled by an 11-bit Identifier (ID) that also determines the priority of objects.

A Layer-7 protocol (Application Layer) was developed to decouple the application from the communication. The service elements, provided by the Application Layer, make it possible to implement an application across the network. These service elements are described in the **CAN** Application Layer (CAL) for Industrial Applications. The communication profile **CAN**open and the drive profile are mounted on the CAL. The basic structure of a communication object is shown in the following diagram:

SOM	COB-ID	RTR	CTRL	Data Segment	CRC	ACK	EOM

SOM		Start of message
COB-ID		Communication Object Identifier (11-bit)
RTR		Remote Transmission Request
CTRL		Control Field (e.g. Data Length Code)
Data Segment	0 ... 8	byte (Data-COB)
	0	byte (Remote-COB)
CRC		Cyclic Redundancy Check
ACK		Acknowledge slot
EOM		End of message

2. 2 COMMUNICATION OBJECT IDENTIFIER

The following diagram shows the layout of the COB Identifier (COB-ID). The Function Code defines the interpretation and priority of the particular object.

10	9	8	7	6	5	4	3	2	1	0
Function-Code				**Module-ID**						

Bit 0 .. 6 Module ID (servo amplifier's **CAN**-bus address, range 1 ... 127; is set up in the setup by switches S12 (MSB) and S11 (LSB) on the servo amplifier)

Bit 7... 10 Function Code (number of the communication object that is defined in the server)

WARNING ***If an invalid Module-ID number (=0 or >127) is set, the drive will default internally to 1.***

The following tables show the default values for the COB Identifier after switching on the servo amplifier. The objects, which are provided with an index (Communication Parameters at Index), can have a new ID assigned after the initialization phase. The indices in brackets are optional.

Predefined broadcast objects (send to all nodes):

Object	Function code (binary)	Resulting COB-IDs		Communication parameters at index
		Dec.	Hex.	
NMT	0000	0	0h	—
SYNC	0001	128	80h	(1005h)
TIME	0010	256	100h	not supported

Predefined Peer-to-Peer objects (node sends to node):

Object	Function code (binary)	Resulting COB-IDs		Communication parameters at index	Priority
		Dec.	Hex.		
EMERGENCY	0001	129..255	81h..FFh	—	high
TPDO 1	0011	385..511	181h..1FFh	1800h	
RPDO 1	0100	513..639	201h..27Fh	1400h	
TPDO 2	0101	641..767	281h..2FFh	1801h	
RPDO 2	0110	769..895	301h..37Fh	1401h	
TPDO 3	0111	897..1023	381h..3FFh	1802h	
RPDO 3	1000	1025..1151	401h..47Fh	1402h	
TPDO 4	1001	1153..1279	481h..4FFh	1803h	
RPDO 4	1010	1281..1407	501h..57Fh	1403h	
SDO (tx*)	1011	1409..1535	581h..5FFh		
SDO (rx*)	1100	1537..1663	601h..67Fh		low
Nodeguard	1110	1793..1919	701h..77Fh	(100Eh)	

** tx = direction of transmission: ➔ Master*
rx = direction of transmission: Master ➔ S200

2. 3 DATA TYPE DEFINITION

This section defines the data types that are used. Each data type can be described by bit- sequences. These bit-sequences are grouped into "Octets" (bytes). The so-called "Little – Endian" format (e.g., Intel format) is used for numerical data types (see also: *DS301 Application Layer "General Description of Data Types and Encoding Rules*").

2.3.1. BASIC DATA TYPES

Unsigned Integer

Data in the basic data type UNSIGNED*n* define exclusively positive integers.

The value range is from 0 ... 2^{n-1}. The bit sequence b = b0 ... b_{n-1} defines the value
UNSIGNED*n*(b) = $b_{n-1}\ 2^{n-1} + ... + b_1\ 2^1 + b_0\ 2^0$

Example The value 266 = $10A_h$ is transmitted in the data type UNSIGNED16, in the form of two octets (1st octet = $0A_h$, 2nd octet = 01_h).

Transmission syntax for the data type UNSIGNED*n*:

Octet Number	1	2	3	4	5	6	7	8
UNSIGNED8	b7..b0							
UNSIGNED16	b7..b0	b15..b8						
UNSIGNED24	b7..b0	b15..b8	b23..b16					
UNSIGNED32	b7..b0	b15..b8	b23..b16	b31..b24				
UNSIGNED40	b7..b0	b15..b8	b23..b16	b31..b24	b39..b32			
UNSIGNED48	b7..b0	b15..b8	b23..b16	b31..b24	b39..b32	b47..b40		
UNSIGNED56	b7..b0	b15..b8	b23..b16	b31..b24	b39..b32	b47..b40	b55..b48	
UNSIGNED64	b7..b0	b15..b8	b23..b16	b31..b24	b39..b32	b47..b40	b55..b48	b63..b56

Signed Integer

Data in the basic data type INTEGERn define both positive and negative integers.
The value range is from $-2^{n-1}-1$ ... $2^{n-1}-1$. The bit sequence b = $b_0..b_{n-1}$ defines the value
INTEGER*n*(b) = $b_{n-2}\ 2^{n-2} + ... + b_1\ 2^1 + b_0\ 2^0$ with $b_{n-1} = 0$

Negative numbers are represented as 2's complement, which means:
INTEGERn(b) = - INTEGERn(b) - 1 with $b_{n-1} = 1$

Example The value -266 = $FEF6_h$ is transmitted in the data type INTEGER16, in the form of two octets (1st octet = F6h, 2nd octet = FE_h).

Transmission syntax for the data type INTEGER*n*:

Octet Number	1	2	3	4	5	6	7	8
UNSIGNED8	b7..b0							
UNSIGNED16	b7..b0	b15..b8						
UNSIGNED24	b7..b0	b15..b8	b23..b16					
UNSIGNED32	b7..b0	b15..b8	b23..b16	b31..b24				
UNSIGNED40	b7..b0	b15..b8	b23..b16	b31..b24	b39..b32			
UNSIGNED48	b7..b0	b15..b8	b23..b16	b31..b24	b39..b32	b47..b40		
UNSIGNED56	b7..b0	b15..b8	b23..b16	b31..b24	b39..b32	b47..b40	b55..b48	
UNSIGNED64	b7..b0	b15..b8	b23..b16	b31..b24	b39..b32	b47..b40	b55..b48	b63..b56

2.3.2. MIXED DATA TYPES

Mixed data types combine basic data types (INTEGER*n*, UNSIGNED*n*, REAL). Two types of mixed data are distinguished:

STRUCT This data type is composed of elements with different data types.

ARRAY This data type is composed of elements of the same data type.

2.3.3. EXTENDED DATA TYPES

Extended data types are derived from basic data types and mixed data types. The types of extended data that are supported are defined below.

Octet String

The data type *OCTET_STRING* is defined with the data type *ARRAY*. *Length* is the length of the octet string.

ARRAY[length] OF UNSIGNED8	OCTET_STRINGlength

Visible String

The data type *VISIBLE_STRING* can be defined with the data type *UNSIGNED8* or the data type *ARRAY*. Permissible values are 00_h and the range from 20_h to $7E_h$. The data are interpreted as 7 bit ASCII code (as per ISO 646-1973(E)). Length is the length of the visible string.

UNSIGNED8	VISIBLE_CHAR
ARRAY[length] OF VISIBLE_CHAR	VISIBLE_STRINGlength

2. 4 COMMUNICATION OBJECTS

Communication objects are described with the help of service elements and protocols. Two basic types of service elements are distinguished:

Unconfirmed services PDO
Confirmed services SDO

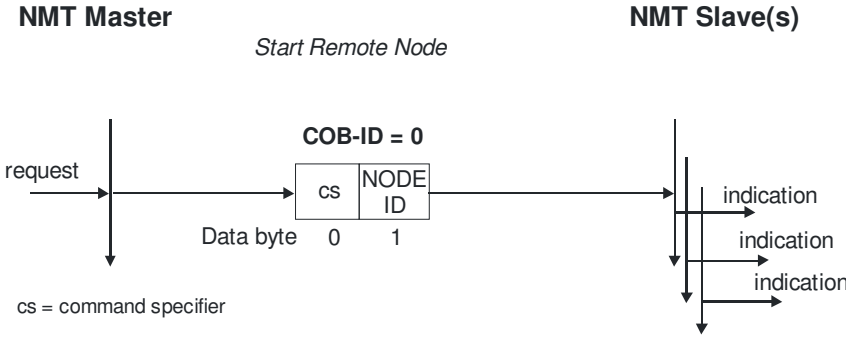
All services require faultless operation of the Data Link and Physical Layer.

The S200 supports communication objects that are described in detail in the following sections:

Network Management Objects (NMT)
Synchronization Object (SYNC)
Emergency Object (EMCY)
Process Data Object (PDO)
Service Data Object (SDO)
Nodeguard

2.4.1. NETWORK MANAGEMENT OBJECTS (NMT)

The NMT telegram looks like this:



The drive supports the following network management functions:

cs = 129, reset node	Causes a cold-start of the drive. This deletes all parameters saved in the RAM and loads the values stored in the EEProM.
cs = 130, reset communication node	Causes a stop of PDO-communication, gives a new bootup-message
cs = 1, start remote node:	Starts the **CAN** node. For example, the PDOs of the drive are enabled for operation. From this moment, transmit-PDOs are transmitted under event control, and cyclical process data operation can commence.
cs = 2, stop remote node	Stops the **CAN** node. The drive no longer responds to any received PDOs or transmits any PDOs.

2.4.2. SYNCHRONIZATION OBJECT (SYNC)

The *SYNC* object usually is used as a periodic Broadcast Object and provides the basic clock for the bus. *SYNC* has a high priority, to ensure constant time intervals. The usage of this protocol is explained in the application note ***CAN****open for S200: Setup and Application Examples*. You can use the *SYNC* object to start motion task of several axes simultaneously for example.

2.4.3. TIME-STAMP OBJECT (TIME)

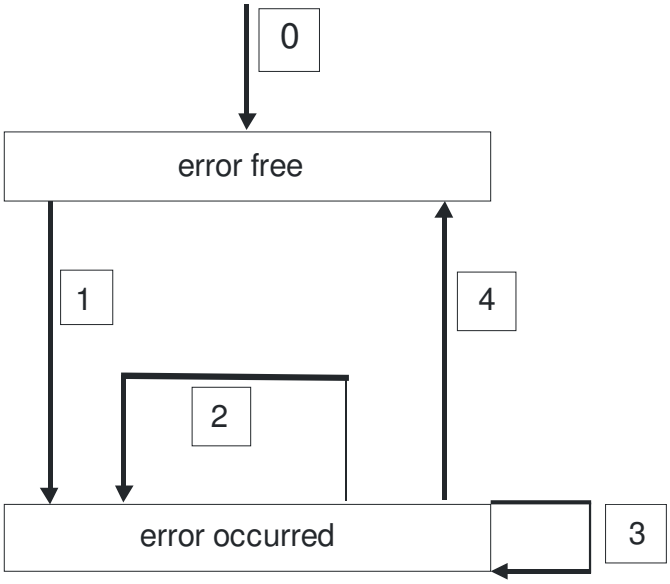
This communication object is not supported by the S200 Position Node Product Line.

2.4.4. EMERGENCY OBJECT (EMCY)

EMCY is event-triggered and generated by an internal fault/error situation. This object is transmitted afresh for every error. Since the error codes are device-dependent, they are described in Emergency Messages. The last 8 Emergency error codes can be read via SDO 1003.

Application of the Emergency Object

The reaction in the event of an error or fault depends on the error class and is therefore variable. For this reason, the reaction is described with the aid of an error status machine. The error conditions error- free and error occurred are distinguished.



The following transitions are defined:

0. After initialization, the error-free status is taken up if no errors are detected. No error signal is generated in this condition.
1. The S200 detects an internal error and indicates this in the first three bytes of the emergency telegram (error code in Bytes 0,1 and error register in Byte 2). Since the S200 can distinguish between different types of error, Byte 3 of the manufacturer-specific error field is used to indicate the error category.
2. One error has been reset, but not all. The *EMCY* telegram contains error code 0000_h and the error register indicates the remaining errors that are present. The manufacture-specific area is set to zero.
3. A new error has occurred. The S200 remains in the error status and transmits an *EMCY* Object with the corresponding error code. The new error code is entered in the S200, Bytes 0 and 1.
4. All errors have been reset. The *EMCY* telegram contains the error code 0000_h, the error register does not indicate any other errors. The manufacture-specific area is set to zero.

Composition of the Emergency Object

The Emergency Object is composed of 8 bytes, divided as follows:

Byte	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Content	Emergency error code		Error register (object 1001_h)	Category	Reserved			

If an Emergency Object is generated, the error condition is then signaled to the status machine (error free / error occurred) by the generation of a second Emergency Object. Only the first four bytes are relevant in this case (*Emergency Error code*, *Error register*, *Category*). Byte 0/1 contains the Error Reset code (0000_h) and Byte 2 indicates if a possible further error is present. If the error register contains 00h, the error status is error-free. Byte 3 contains the category. The interpretations of the error numbers (*error code*) and the error categories are described below. The error register is defined through object 1001_h *Error register*.

Emergency Messages

Emergency messages are triggered by internal equipment errors. They have a high ID-priority, to ensure quick access to the bus. An *Emergency message* contains an error field with pre-defined error/fault numbers (2 bytes), an error register (1byte), the error category (1 byte) and additional information. The higher-value byte of the error number describes the error category, and the lower-value byte provides the error number in this category.

Error numbers from 0000_h to $7FFF_h$ are defined in the communication or drive profile. Error numbers from $FF00_h$ to $FFFF_h$ have manufacturer-specific definitions. The error category can be used to classify the significance of any errors that occur. The following error categories are defined:

1 Errors that can only be cleared by a reset (*COLDSTART* command, or Bit 7 in the control word). These errors are also indicated by blinking of the LED display in the front panel. (Fxx, xx = error number)
2 Errors that can be cleared by Bit 11 in the control word.
3 Error messages that may appear when a PDO is processed.
4 Faults, that **cannot** be cleared by the user.
5 Operating errors/warnings.

The following table describes the various Error Codes:

Error Code	Category	Description
0000_h	—	Error reset or no error (mandatory)
1000_h	—	Generic error (mandatory)
1080_h	5	No BTB/RTO (status not ready for operation)
2330_h	1	Error in ground connection (F22)
2380_h	1	Error in motor connection (phase fault) (F12)
3100_h	1	No mains/line-BTB (F16)
3110_h	1	Overvoltage in DC-bus/DC-link (F02)
3120_h	1	Under-voltage in DC-bus/DC-link (F05)
4110_h	1	Ambient temperature too high (F13)
4210_h	1	Heat sink temperature too high (F01)
4310_h	1	Motor temperature too high (F06)
5380_h	1	Fault in A/D converter (F17)
5400_h	1	Fault in output stage (F14)
5420_h	1	Ballast (chopper) (F18)
5530_h	1	Serial EEProM (F09)
6320_h	3	Parameter error
7111_h	1	Braking error/fault (F11)
7122_h	1	Commutation error (F25)
7181_h	5	Could not enable
7303_h	1	Feedback device error (F04)
7305_h	1	Signal failure digital encoder input (F10)
8053_h	1	Handling error (F21)
8182_h	1	**CAN** bus off (F23)
8331_h	1	I2t (torque fault, F15)
8480_h	1	Over-speed (F08)
8611_h	2	Lag/following error (n03/F03)
8681_h	5	Invalid motion task number
$FF01_h$	4	Serious exception error (F32)
$FF02_h$	3	Error in PDO elements
FF04	1	Slot error (F20)
FF06	2	Warning display as error (F24)
FF07	2	Homing error (drove onto HW limit switch) (F26)
FF11	2	Emergency timeout failure(F30)

2.4.5. SERVICE DATA OBJECTS (SDO)

SDOs are used to implement access to the Object Dictionary. The SDOs are required for parameterization and for status polling. Access to an individual object is made with a multiplexer via the Index and Subindex of the Object Dictionary. The following communication protocols are supported by the S200:

- Initiate SDO Download Protocol
- Download SDO Segment Protocol
- Initiate SDO Upload Protocol
- Upload SDO Segment Protocol
- Abort SDO Transfer Protocol

The definitions of the individual communication services and protocols can be found in DS301. Examples of the usage of SDOs can be found in the apendicies.


NOTE

Since an SDO is a confirmed service, the system must always wait for the SDO response telegram before it is allowed to transmit a new telegram

Composition of the Service Data Object

An SDO consists of the following components:

Byte	1	2	3	4	5	6	7	8
Content	rw	Index		Subindex	Data			

1. The control byte (Byte 1):
   The control byte determines whether the SDO should write or read the content of the entry in the Object Dictionary (see Appendix).

   Data exchange with the S200 is governed by the CMS multiplexed domain protocols standard, as described in the **CAN** standard DS 202.

   To read data, the control byte must be written in the manner shown below:

Bit	7	6	5	4	3	2	1	0
Content	ccs*=2			X	X	X	X	X

   *ccs ➔ client command specifier (ccs = 2 ➔ initiate upload request)
   X ➔ free data

   So a value of 0100 0000 (binary) or 40h has to be transmitted in the control byte.

   The servo amplifier sends back a corresponding response byte:

Bit	7	6	5	4	3	2	1	0
Content	scs*=2			X	n		e	s

   *scs ➔ server command specifier (scs = 2 ➔ initiate upload response)
   n ➔ only valid for e = s = 1
   if this is so, n contains the number of bytes that do not contain data
   X ➔ free data

   If reading is successful, the response byte always has set the bits 0 and 1 (e = s = 1).

   Encoded byte length in the SDO response:
   0x43 - 4 bytes
   0x47 - 3 bytes
   0x4B - 2 bytes
   0x4F - 1 byte.

If an error occurs, *scs* is set to 4, the response byte is 0x80 and the error information is in the four byte data field. The decoding of the error can be found section 2.4.4, Emergency Messages.

To write data, the control byte must be written in the manner shown below:

Client — *Initiate Domain Download* — Server

	Byte 1								2	3	4	5	6	7	8	
request	7	6	5	4	3	2	1	0								indication
→	ccs=1			X	n		e	s	m			d				→

n,e and *s* are defined like in the reading case
m: index + sub-index
d: data field with 4 bytes

The data length of an object can be taken from the object dictionary in the appendix. The control byte should be:

0x23 for a 4-byte access
0x27 for a 3-byte access
0x2B for a 2-byte access
0x2F for a 1-byte access

	Byte 1								2	3	4	5	6	7	8	
confirm	7	6	5	4	3	2	1	0								response
←	scs=3			X					m			reserved				←

2. Index (Bytes 2 and 3):
   The Index is the main entry in the Object Dictionary, and divides the parameters into groups. (Example: Index 1018h is the Identity Object).
   As for all **CAN** data, the Index is stored with the bytes in reverse order.
   E.g., Index 6040h means Byte 2 = 40h, Byte 3 = 60h)
3. Subindex (Byte 4):
   The Subindex divides the parameters within a group of parameters.
4. Data field (Bytes 5 ... 8):
   These components are used for the exchange of user data. In read-request telegrams to the S200, they are set to 0. They have no content in a write confirmation from the S200 if the transfer was successful, but if the write operation was faulty, they contain an error code (see Abort SDO Protocol).

Initiate SDO Download Protocol

The *Initiate SDO Download* protocol is used for write access to objects with up to 4 bytes of user data (*expedited transfer*) or to initiate a segment transfer (*normal transfer*). *Expedited* transfer is also used for objects that only have the character of a command (e.g. ASCII: *SAVE*) and do not require further user data.

Download SDO Segment Protocol

The *Download SDO Segment* protocol is used for write access to objects with more than 4 bytes of user data (*normal transfer*). This service is not supported by the S200 at present, since there are no objects that make use of more than 4 bytes of user data.

Initiate SDO Upload Protocol

The *SDO Upload* protocol is used for read access to objects with up to 4 bytes of user data (*expedited transfer*) or to initiate a segment transfer (*normal transfer*).

Upload SDO Segment Protocol

The *Upload SDO Segment* protocol is used for read access to objects with more than 4 bytes of user data (*normal transfer*). This service is not supported by the S200 at present, since there are no objects that make use of more than 4 bytes of user data.

Abort SDO Protocol

The *Abort SDO* protocol breaks off SDO transmission, and indicates the error that caused the break in transmission through an abort code (*error code*). The error code is in the format of an UNSIGNED32 value. The following table shows possible reasons for an abort SDO.

Abort Code	Description
$0601\ 0000_h$	Unsupported access to this object
$0601\ 0001_h$	Attempted read access to a write-only object
$0601\ 0002_h$	Attempted write access to a read-only object
$0602\ 0000_h$	Object does not exist in Object Dictionary
0604 0041h	Object cannot be mapped to a PDO
$0604\ 0042_h$	Size and number of mapped objects exceed permissible PDO length
$0604\ 0043_h$	General parameter incompatibility
$0607\ 0010_h$	Data type incompatible, length of service parameter is incompatible
$0609\ 0011_h$	Subindex does not exist
$0609\ 0030_h$	Outside value range for the parameter (only for write access)
$0609\ 0031_h$	Parameter value too high
$0609\ 0032_h$	Parameter value too low
$0800\ 0020_h$	Data cannot be transmitted or saved
$0800\ 0022_h$	Data cannot be transmitted or saved because of device status
$FF03\ 0000_h$	OS cmd buffer full

Abort Codes not listed above are reserved.

Factor Groups (fg) (DS402)

Factor groups define the units of position-, velocity- and acceleration set points. These values are converted into drive-specific parameters.

CAUTION

The unit definitions are not defined in the CANopen profile DS402. Use the SDOs 6089_h to $609E_h$.

Set the drive parameters for the unit definitions as: PUNIT = 0 (counts), VUNIT = 0 (counts / s), ACCUNIT = 3 (counts / s²)

There is a possibility to convert between physical dimensions and sizes, and the internal units used in the device (increments). Several factors can be implemented. This section describes how these factors influence the system, how they are calculated, and which data are necessary to build them. The factors defined in the factor group set up a relationship between device-internal units (increments) and physical units.

The factors are the result of the calculation of two parameters called dimension index and notation index. The dimension index indicates the physical dimension, the notation index indicates the physical unit and a decimal exponent for the values. These factors are directly used to normalize the physical values.

The notation index can be used in two ways:

1. For a unit with decimal scaling and notation index < 64, the notation index defines the exponent/decimal place of the unit.
2. For a unit with non-decimal scaling and notation index > 64, the notation index defines the Subindex of the physical dimension of the unit.

SDOs for Position Calculation

SDO 6089_H: Position Notation Index (DS402)
SDO $608A_H$: position dimension index (DS402)
SDO $608F_H$: Position encoder resolution (DS402)
SDO 6091_H: Gear ratio (DS402)
SDO 6092_H: Feed constant (DS402)
SDO 6093_H: Position factor (DS402)

SDOs for Velocity Calculations

SDO $608B_H$: velocity notation index (DS402)
SDO $608C_H$: velocity dimension index (DS402)

SDOs for Acceleration Calculations

SDO $608D_H$: acceleration notation index (DS402
SDO 608Eh: acceleration dimension index (DS402)
SDO 6097h: Acceleration factor (DS402)

2.4.6. PROCESS DATA OBJECT (PDO)

PDOs are used for real-time data communication. PDOs can, for instance, be used to set up controllers similar to analog drives. Instead of ±10VDC set points and roD feedback, digital speed set points and position feedback are attained via PDOs in this case.

Transmission is carried out unconfirmed without a protocol "overhead". This communication object uses the unconfirmed communication service.

PDOs are defined via the Object Dictionary for the S200. Mapping is made during the configuration phase, with the help of SDOs. Length is defined with the mapped objects.

The definition of the PDO service and protocol can be found in DS301. Examples of the usage of PDOs can be found in the application note, ***CAN****open for the S200: Setup and Application Examples*.

Basically, two types of PDOs can be distinguished, depending on the direction of transmission:

Transmit-PDOs (TPDOs) (S200 ➔ Master)
The TPDOs transmit data from the S200 to control system
(e.g., actual value objects, instrument status).

Receive-PDOs (RPDOs) (Master ➔ S200)
The RPDOs receive data from control system to the S200.
(e.g., set points).

The S200 supports four independent PDO channels for each direction of transmission. The channels are labeled by numbers 1 to 4.

Receive PDOs (RXPDO)

Four Receive PDOs can be configured in the servo amplifier:
— Configuration of the communication (SDOs 1400-1403$_h$)
— Configuration of the PDO-contents (mapping, SDOs 1600-1603$_h$)

RXPDO 1

Subindex	Value	Meaning
0	1	One PDO-mapping entry
1	60 40 00 10	Control word

RXPDO 2

Subindex	Value	Meaning
0	2	Two PDO-mapping entries
1	60 40 00 10	Control word
2	60 60 00 08	Modes of Operation

RXPDO 3

Subindex	Value	Meaning
0	2	Two PDO-mapping entries
1	60 40 00 10	Control word
2	60 7A 00 20	Target Position (Mode PP)

RXPDO 4

Subindex	Value	Meaning
0	2	Two PDO-mapping entries
1	60 40 00 10	Control word
2	60 FF 00 20	Target Velocity (Mode PV)

Transmit PDOs (TXPDO)

Four Transmit PDOs can be configured in the servo amplifier:
— Configuration of the communication (SDOs 1800-1803$_h$)
— Configuration of the PDO-contents (mapping, SDOs 1A00-1A03$_h$)

TXPDO 1

Subindex	Value	Meaning
0	1	One PDO-mapping entry
1	60 41 00 10	Status word

TXPDO 2

Subindex	Value	Meaning
0	2	Two PDO-mapping entries
1	60 41 00 10	Status word
2	60 61 00 08	Modes of Operation display

TXPDO 3

Subindex	Value	Meaning
0	2	Two PDO-mapping entries
1	60 41 00 10	Status word
2	60 64 00 20	Position actual value

TXPDO 4

Subindex	Value	Meaning
0	2	Two PDO-mapping entries
1	60 41 00 10	Status word
2	60 6C 00 20	Velocity actual value

PDO Configuration

There are two types of PDOs: Receive PDOs (RPDOs) and transmit PDOs (TPDOs). The content of the PDOs is pre-defined. If the data content is not appropriate for a special application, the data objects in the PDOs can be remapped freely.

There are two parameter sets each for the configuration of each of the four possible PDOs, and they can be set up through the corresponding SDOs:

1. Mapping parameters, to determine which data are available (mapped) in the selected PDO and to define, which data are contained.
2. Communication parameters that define whether the PDOs operate in synchronized mode or event-driven (SDOs 1400_h to 1403_h, 1800_h to 1803_h).

One data entry in the PDOs looks like:

MSB		LSB
Index (16 bit)	Subindex (8 bit)	Data length in bits (8 bit)

The configuration procedure for a free mapping of a PDO looks like this (example for TPDO1):

1. Delete the actual mapping of the PDO by writing a 0 to the Subindex 0 of the mapping SDO.

COB-ID	Control byte	Index		Sub-index	Data	Comment
		Low byte	High byte			
601	2F	00	1A	00_h	00 00 00 00	Delete actual mapping

2. Build the mapping with object dictionary objects (see page)) which are map-able, e.g.

COB-ID	Control byte	Index		Sub-index	Data	Comment
		Low byte	High byte			
601	23	00	1A	01_h	10 00 41 60	1st entry: **CAN**open status word with 16 bit
601	23	00	1A	02_h	20 00 02 10	2nd entry: Manufacturer status with 32 bits

3. Write the number of mapped objects to Subindex 0 of the mapping SDO.

COB-ID	Control byte	Index		Sub-index	Data	Comment
		Low byte	High byte			
601	2F	00	1A	00_h	02 00 00 00	Check for the right number of entries

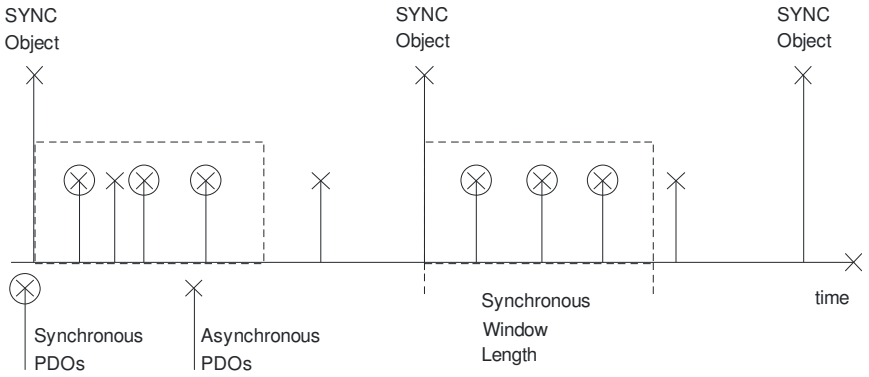
Mapping must be done before the network management is switched to OPERATIONAL.

Transmission Modes

The following PDO transmission modes are available:

Synchronous transmission
Asynchronous transmission

The pre-defined SYNC Object is transmitted periodically (bus clock), to synchronize the drives. Synchronous PDOs are transmitted within a pre-defined time window immediately following the SYNC Object. The transmission modes are set up with the aid of the PDO communication parameters.



Trigger modes

Three different trigger modes are distinguished:

Event driven
The transmission of the telegrams is triggered by an object-specific event.

Time driven
If event driven signals put a high strain on the bus, you can determine the period of time after which a PDO can be transmitted again via the inhibit time (Communication parameter, sub-index 03_h)

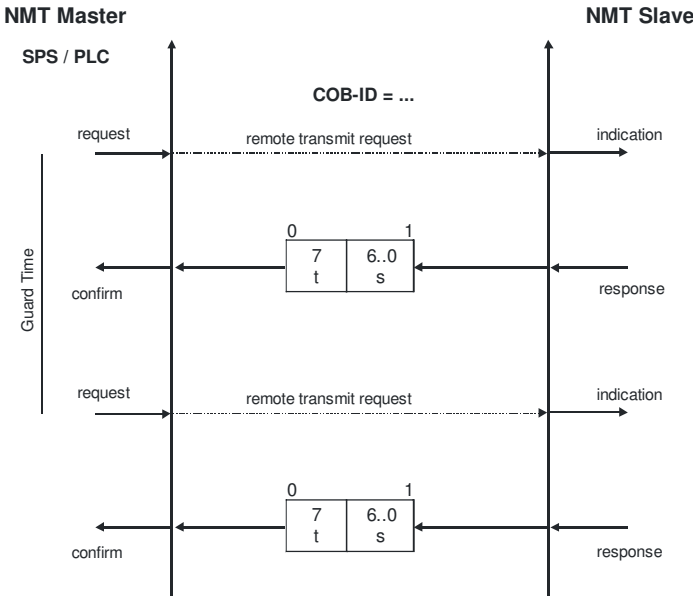
Event Timer driven
If a PDO shall be sent within a defined time interval, even if it doesn't change, this interval can be defined by a special SDO.

2.4.7. NODEGUARD

The Node Guarding protocol is a functional monitoring for the drive. It requires that the drive is accessed at regular intervals by the **CAN**open master.

The maximum time interval that is permitted between two Nodeguard telegrams is given by the product of the *Guard Time* and the *Life Time Factor*. If one of these two values is 0, then the response monitoring is de-activated.

If the drive is not accessed within the time defined by SDOs $100C_h$ und $100D_h$, then Warning N04 (response monitoring) appears on the drive, the drive is braked to a stop with the Quickstop ramp, and any other movement is prevented. (parameter DECSTOP, SDO6085 sub0). The time sequence for node guarding is as shown below:



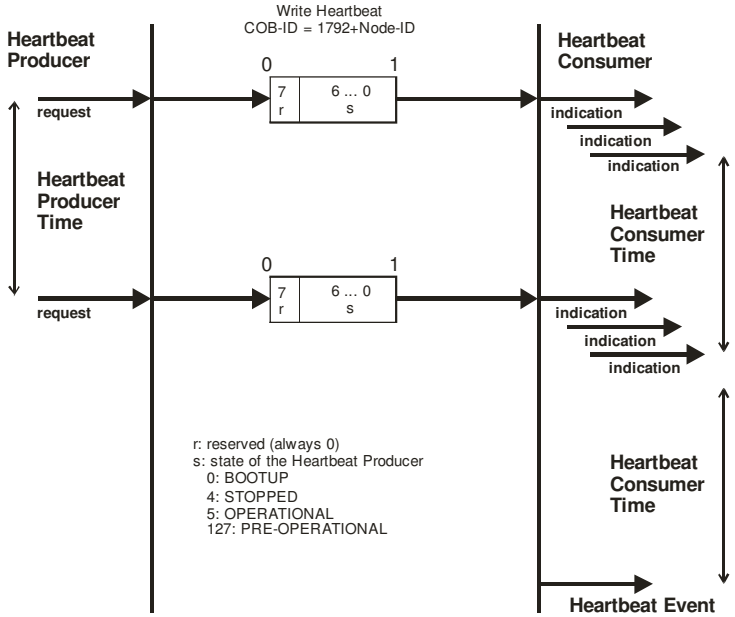
t = toggle Bit, changes its status with every slave telegram
s = status of the NMT slave status machine

Node guarding is carried out by the Master through RTR telegrams with the COB-ID 700_h + slave node address.

2.4.8. HEARTBEAT

The Heartbeat Protocol defines an Error Control Service without need for remote frames. A Heartbeat Producer transmits a Heartbeat message cyclically. One or more Heartbeat Consumer receive the indication. The relationship between producer and consumer is configurable via SDO 1016h / 1017h. The Heartbeat Consumer guards the reception of the Heartbeat within the Heartbeat Consumer Time. If the Heartbeat is not received within the Heartbeat Consumer Time, a Heartbeat Event is generated.

Heartbeat protocol:

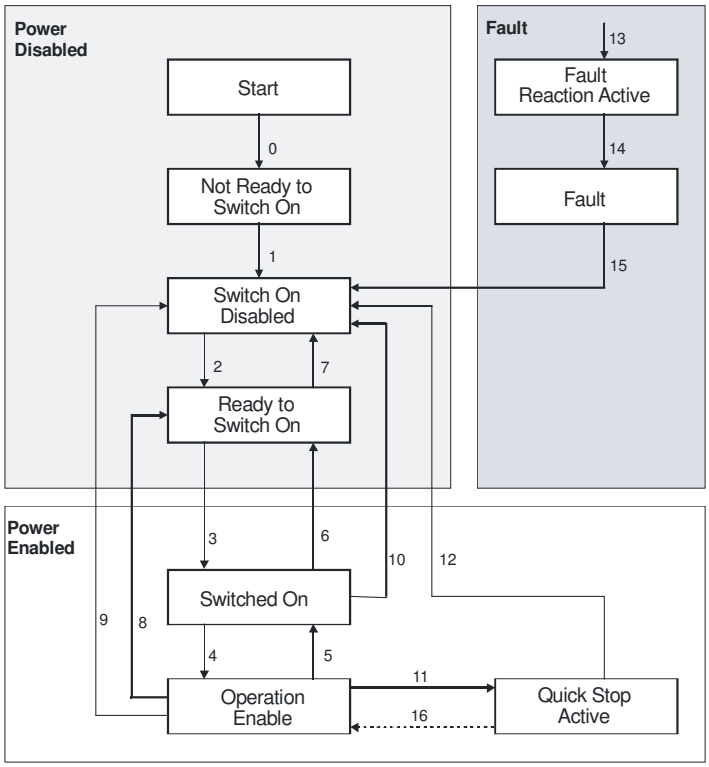


2. 5 DEVICE CONTROL (DC)

The device control of the S200 can be used to carry out all the motion functions in the corresponding modes. The control of the is implemented through a mode-dependent status machine. The status machine is controlled through the control word.

The mode setting is made through the object “Modes of Operation”. The states of the status machine can be revealed by using the status word.

2.5.1. STATUS MACHINE (DS402)



2.5.2. STATES OF THE STATUS MACHINE

State	Description
Not Ready for Switch On	S200 is not ready to switch on, there is no operational readiness (BTB/RTO) signaled from the controller program.
Switch On Disable	S200 is ready to switch on, parameters can be transferred, the DC-link voltage can be switched on, motion functions cannot be carried out yet.
Ready to Switch On	DC-link voltage may be switched on, parameters can be transferred, motion functions cannot be carried out yet.
Switched On	DC-link voltage must be switched on, parameters can be transferred, motion functions cannot be carried out yet, output stage is switched on (enabled).
Operation Enable	No fault present, output stage is enabled, motion functions are enabled.
Quick Stop Active	Drive has been stopped with the emergency ramp, output stage is enabled, motion functions are enabled.
Fault Reaction Active	A fault has occured and the drive is stopped with the quickstop ramp.
Fault	A fault is active, the drive has been stopped and disabled.

2.5.3. TRANSITIONS OF THE STATUS MACHINE

The state transitions are affected by internal events (e.g. switching off the DC-link voltage) and by the flags in the control word (bits 0,1,2,3,7).

Transition	Event	Action
0	Reset	Initialization
1	Initialization completed successfully. S200 is ready to operate.	none
2	Bit 1 Disable Voltage and Bit 2 *Quick Stop* are set in the control word *(Shutdown* command). DC-link voltage may be present.	none
3	Bit 0 is also set (*Switch On* command)	Output stage is switched on (enabled), provided that the hardware enable is present (logical AND). Drive has torque.
4	Bit 3 is also set (*Enable Operation* command)	Motion function is enabled, depending on the mode that is set.
5	Bit 3 is canceled (*Disable Operation* command)	Motion function is inhibited. Drive is stopped, using the relevant ramp (mode-dependent). The present position is maintained.
6	Bit 0 is canceled (*Shutdown* command)	Output stage is disabled. Drive has no torque.
7	Bits 1 and 2 are canceled (*Quick Stop* / *Disable Voltage* command)	none
8	Bit 0 is canceled (*Shutdown* command)	Output stage is disabled. Drive has no torque.
9	Bit 1 is canceled (*Disable Voltage* command)	Output stage is disabled. Drive has no torque.
10	Bits 1 and 2 are canceled (*Quick Stop* / *Disable Voltage* command)	Motion function is enabled, depending on the mode that is set.
11	Bit 2 is canceled (*Quick Stop* command)	Drive is stopped with the emergency braking ramp. The output stage remains enabled. Set points are canceled (motion block number, digital set point, speed for jogging or homing). Bit 2 must be set again before any further motion tasks can be performed.
12	Bit 1 is canceled (*Disable Voltage* command)	Output stage is disabled. Drive has no torque.
13	Fault reaction active	Execute appropriate fault reaction
14	Fault reaction is completed	Drive function is disabled. The power section may be switched off.
15	"Fault Reset" command received from host	A reset of the fault condition is carried out if no fault exists currently on the drive. After leaving the state Fault the Bit7 'Reset Fault' of the control word has to be cleared by the host
16	Bit 2 is set	Motion function is enabled again.

CAUTION

If the servo amplifier is operated through the control word / status word, then no control commands may be sent through another communication channel (RS232, CANopen, ASCII channel, Option board).

3. OPERATION MODES

3. 1 PROFILE VELOCITY MODE (PV) (DS402)

The *profile velocity* mode enables the processing of velocity set points and the associated accelerations.

Index	Object	Name	Type
$60FF_h$	VAR	Target velocity	INTEGER32
6040_h	VAR	Control word	INTEGER16
6041_h	VAR	Status word	UNSIGNED16
$606C_h$	VAR	Velocity actual value	INTEGER32
6063_h	VAR	Position actual value	INTEGER32
6083_h	VAR	Profile acceleration	UNSIGNED32
6084_h	VAR	Profile deceleration	UNSIGNED32
6086_h	VAR	Motion profile type	INTEGER16
6094_h	ARRAY	Velocity encoder factor	UNSIGNED32

3. 2 PROFILE TORQUE MODE (TQ) (DS402)

The *profile torque* mode enables the processing of torque set points and the associated current.

Index	Object	Name	Type
6071_h	VAR	Target torque	INTEGER16
6073_h	VAR	Max current	UNSIGNED16
6077_h	VAR	Torque actual value	INTEGER16

3. 3 POSITION CONTROL FUNCTION (PC) (DS402)

This section describes the actual position values that are associated with the position controller of the drive. They are used for the *profile position* mode.

Index	Object	Name	Type
6040_h	VAR	Control word	INTEGER16
6041_h	VAR	Status word	UNSIGNED16
6063_h	VAR	Position actual value	INTEGER32
6064_h	VAR	Position actual value	INTEGER32
6065_h	VAR	Following error window	UNSIGNED32
6067_h	VAR	Position window	UNSIGNED32
6068_h	VAR	Position window time	UNSIGNED16
$607A_h$	VAR	Target position	INTEGER32
$607C_h$	VAR	Home-offset	INTEGER32
$607D_h$	ARRAY	Software position limit	INTEGER32
$607F_h$	VAR	Maximum profile velocity	UNSIGNED32
6093_h	VAR	Position factor	UNSIGNED32
6094_h	ARRAY	Velocity encoder factor	UNSIGNED32
6096_h	ARRAY	Acceleration factor	UNSIGNED32

4. CONFIGURATION PARAMETERS

CAN Object Number	Parameter Name	Parameter Type	Description
3501 (hex)	ACC	Variable rw	Acceleration Ramp
3502 (hex)	ACCR	Variable rw	Acceleration Ramp for homing/jog modes
3503 (hex)	ACTFAULT	Variable rw	Active Fault Mode
3504 (hex)	ACTIVE	Variable ro	Output stage active/inhibited
3506 (hex)	AENA	Variable rw	Software Auto-Enable
3507 (hex)	ANCNFG	Variable rw	Configuration of Analog Input
3508 (hex)	ANDB	Variable rw	Dead Band of the Analog Velocity Input Signal
3509 (hex)	ANIN1	Variable ro	Voltage at Analog Input 1
350B (hex)	ANOFF1	Variable rw	Analog Offset for analog input 1
3511 (hex)	AVZ1	Variable rw	Filter Time Constant for analog input 1
3518 (hex)	CLRFAULT	Command	Clear Drive Fault
3519 (hex)	CLRHR	Command	Bit 5 of status register STAT is cleared
351B (hex)	CLrwARN	Variable rw	Warning mode
351E (hex)	CTUNE	Command	Calculate current parameters
3522 (hex)	DEC	Variable rw	Deceleration Rate
3523 (hex)	DECDIS	Variable rw	Deceleration used on Disable Output Stage
3524 (hex)	DECR	Variable rw	Deceleration Ramp for homing/jog modes
3525 (hex)	DECSTOP	Variable rw	Quick Stop, braking ramp for emergency situations
3527 (hex)	DICONT	Variable ro	Drive Continuous Current
3529 (hex)	DIPEAK	Variable ro	Drive Peak Rated Current
352C (hex)	DREF	Variable rw	Direction for Homing
352D (hex)	DRVSTAT	Variable ro	internal Status information
352E (hex)	DR_TYPE	Variable ro	Gives the Output Stage Identification
3534 (hex)	ENCMODE	Variable rw	Selection of Encoder Emulation
3535 (hex)	ENCOUT	Variable rw	Resolution Encoder Emulation EEO (ROD)
3537 (hex)	ENCZEro	Variable rw	Zero Pulse Offset EEO (ROD)
3538 (hex)	EXTMUL	Variable rw	ext. Encoder multiplier
3539 (hex)	EXTPOS	Variable rw	Position Value For Position Control
353A (hex)	EXTWD	Variable rw	External Watch Dog (Field bus)
353C (hex)	FILTMODE	Variable rw	Smith Predictor
353E (hex)	GEARI	Variable rw	Input Factor for Electronic Gearing
353F (hex)	GEARMODE	Variable rw	Position Input Electronic Gearing Mode
3540 (hex)	GEAro	Variable rw	Output Factor for Electronic Gearing
3558 (hex)	I	Variable ro	Current Monitor
355A (hex)	I2TLIM	Variable rw	I2T Warning
355D (hex)	ID	Variable ro	D-component of Current Monitor
3561 (hex)	IN1	Variable ro	Status of Digital Input 1
3561 (hex)	IN5_20	Variable ro	Status of digital inputs 5 ...20
3562 (hex)	IN1MODE	Variable rw	Function of Digital Input 1
3562 (hex)	IN5_20MODE	Variable rw	Function of digital inputs 5 ...20
3563 (hex)	IN1TRIG	Variable rw	Variable for IN1MODE
3563 (hex)	IN5_20TRIG	Variable rw	Variable for digital inputs 5 ...20
3564 (hex)	IN2	Variable ro	Status of Digital Input 2
3565 (hex)	IN2MODE	Variable rw	Function of Digital Input 2

CAN Object Number	Parameter Name	Parameter Type	Description
3566 (hex)	IN2TRIG	Variable rw	Variable for IN2MODE
3567 (hex)	IN3	Variable ro	Status of Digital Input 3
3568 (hex)	IN3MODE	Variable rw	Function of Digital Input 3
3569 (hex)	IN3TRIG	Variable rw	Variable for IN3MODE
356A (hex)	IN4	Variable ro	Status of Digital Input 4.
356B (hex)	IN4MODE	Variable rw	Function of Digital Input 4
356C (hex)	IN4TRIG	Variable rw	Variable for IN4MODE
356D (hex)	INPOS	Variable ro	Status of In-Position Signal
356E (hex)	IPEAK	Variable rw	Application Peak Current
3570 (hex)	IQ	Variable ro	Q-Component of Current Monitor
3571 (hex)	ISCALE1	Variable rw	Scaling of Analog Current Set point 1
3572 (hex)	ISCALE2	Variable rw	Scaling of Analog Current Set point 2
3577 (hex)	ML	Variable rw	Stator Inductance of the Motor
3578 (hex)	LATCH2P16	Variable rw	Latched 16-bit Position (positive edge)
3579 (hex)	LATCH2N16	Variable rw	Latched 16-bit Position (negative edge)
357A (hex)	LATCH2P32	Variable rw	Latched 32-bit Position (positive edge)
357B (hex)	LATCH2N32	Variable rw	Latched 32-bit Position (negative edge)
357C (hex)	LATCH1P32	Variable rw	Latched 32-bit Position (positive edge)
357D (hex)	LATCH1N32	Variable rw	Latched 32-bit Position (negative edge)
3584 (hex)	MAXTEMPE	Variable rw	Ambient Temperature Switch off Threshold
3585 (hex)	MAXTEMPH	Variable rw	Heat Sink Temperature Switch off Threshold
3586 (hex)	MAXTEMPM	Variable rw	Motor Temperature Switch off Threshold
358C (hex)	VLIM	Variable rw	Max. Velocity
358D (hex)	MH	Command	Start Homing
358E (hex)	MICONT	Variable rw	Motor Continuous Current Rating
358F (hex)	MIPEAK	Variable rw	Motor Peak Current Rating
3591 (hex)	MJOG	Command	Start Jog Mode
3593 (hex)	MKT	Variable rw	Motor KT
3595 (hex)	MLGC	Variable rw	Current Control loop Adaptive Gain (Q-component at rated current)
3596 (hex)	MLGD	Variable rw	Adaptive Gain for Current Control loop, D-component
3597 (hex)	MLGP	Variable rw	Current Control loop Adaptive Gain (Q-component at peak current)
3598 (hex)	MLGQ	Variable rw	Absolute Gain of Current Control loop
3599 (hex)	MNUMBER	Variable rw	Motor Number
359C (hex)	MPHASE	Variable rw	Motor Phase, Feedback Offset
35A3 (hex)	MSPEED	Variable rw	Maximum Rated Motor Velocity
35A5 (hex)	MTANGLP	Variable rw	Current Lead
35A6 (hex)	MTYPE	Variable rw	Motor Type
35A7 (hex)	MVANGLB	Variable rw	Velocity-dependent Lead (Start Phi)
35A8 (hex)	MVANGLF	Variable rw	Velocity-dependent Lead (Limit Phi)
35AA (hex)	NONBTB	Variable rw	Mains-BTB Check On/Off
35AD (hex)	NREF	Variable rw	Homing Mode
35AE (hex)	O1	Variable rw	State of Digital Output 1
35AE (hex)	O3_18	Variable rw	State of Digital Output 1
35AF (hex)	O1MODE	Variable rw	Function of Digital Output 1
35AF (hex)	O3_18MODE	Variable rw	Function of Digital Output 1

CAN Object Number	Parameter Name	Parameter Type	Description
35B0 (hex)	O1TRIG	Variable rw	Auxiliary Variable for O1MODE
35B0 (hex)	O3_18TRIG	Variable rw	Auxiliary Variable for O1MODE
35B1 (hex)	O2	Variable rw	State of Digital Output 2
35B2 (hex)	O2MODE	Variable rw	Function of Digital Output 2
35B3 (hex)	O2TRIG	Variable rw	Auxiliary Variable for O2MODE
35B4 (hex)	OPMODE	Variable rw	Operating Mode
35B6 (hex)	OVRIDE	Variable rw	Override Function for Motion Tasks
35B7 (hex)	O_ACC	Variable rw	Acceleration Time 1 for Motion Task 0
35B9 (hex)	O_C	Variable rw	Control Variable for Motion Task 0
35BA (hex)	O_DEC	Variable rw	Braking Time 1 for Motion Task 0
35BC (hex)	O_FN	Variable rw	Next Task Number for Motion Task 0
35BD (hex)	O_FT	Variable rw	Delay before Next Motion Task
35BE (hex)	O_P	Variable rw	Target Position/Path for Motion Task 0
35BF (hex)	O_V	Variable rw	Target Speed for Motion Task 0
35C0 (hex)	PBAL	Variable ro	Actual Regen Power
35C1 (hex)	PBALMAX	Variable rw	Maximum Regen Power
35C2 (hex)	PBALRES	Variable rw	Select Regen Resistor
35C5 (hex)	PE	Variable ro	Actual Following Error
35C6 (hex)	PEINPOS	Variable rw	In-Position Window
35C7 (hex)	PEMAX	Variable rw	Max. Following Error
35C8 (hex)	PFB	Variable ro	Actual Position from Feedback Device
35C9 (hex)	PFB0	Variable ro	Position from External Encoder
35CA (hex)	PGEARI	Variable rw	Position Resolution (Numerator)
35CB (hex)	PGEAro	Variable rw	Position Resolution (Denominator)
35CD (hex)	PMODE	Variable rw	Line Phase Mode
35CF (hex)	POSCNFG	Variable rw	Axes Type
35D1 (hex)	PRBASE	Variable rw	Position Resolution
35D2 (hex)	PRD	Variable ro	20-bit Position Feedback
35D6 (hex)	PTMIN	Variable rw	Min. Acceleration Ramp for Motion Tasks
35D7 (hex)	PV	Variable ro	Actual Velocity (Position Control Loop)
35D8 (hex)	PVMAX	Variable rw	Max. Velocity for Position Control
35D9 (hex)	PVMAXN	Variable rw	Max. (Negative) Velocity for Position Control
35DB (hex)	PVMAXP	Variable rw	Max. Velocity for Position Control
35DD (hex)	READY	Variable ro	Status of the Software Enable
35E4 (hex)	REMOTE	Variable ro	Status of the Hardware Enable
35E7 (hex)	roFFS	Variable rw	Reference Offset
35EA (hex)	S	Command	Stop Motor and Disable Drive
35F0 (hex)	SETREF	Command	Set Reference Point
35FB (hex)	STAT	Variable ro	Drive Status Word
35FD (hex)	STATUS	Variable ro	Detailed Amplifier Status
35FE (hex)	STOP	Command	Stop Motion Task
35FF (hex)	STOPMODE	Variable rw	Brake Response for Disable
3600 (hex)	SWCNFG	Variable rw	Configuration of software limit switches
3604 (hex)	SWE1	Variable rw	SW limit switch (smallest position)
3606 (hex)	SWE2	Variable rw	SW limit switch (biggest position)
360E (hex)	T	Command	Digital Current Set point
3613 (hex)	TRJSTAT	Variable ro	Status2 Information

CAN Object Number	Parameter Name	Parameter Type	Description
3617 (hex)	UVLTMODE	Variable rw	Under-voltage Mode
3618 (hex)	V	Variable ro	Actual Velocity
361A (hex)	VBUS	Variable ro	DC-bus voltage
361B (hex)	VBUSBAL	Variable rw	Maximum Line Voltage
361C (hex)	VBUSMAX	Variable ro	Maximum DC-bus Voltage
361D (hex)	VBUSMIN	Variable rw	Minimum DC-bus Voltage
361E (hex)	VCMD	Variable ro	Internal Velocity Set point in RPM
3620 (hex)	VEL0	Variable rw	Standstill Threshold
3621 (hex)	VJOG	Variable rw	Speed for Jog Mode
3622 (hex)	VLIMP	Variable rw	Max. Velocity
3623 (hex)	VLIMN	Variable rw	Max. Negative Velocity
3626 (hex)	VMUL	Variable rw	Velocity Scale Factor
3627 (hex)	VOSPD	Variable rw	Over-speed
3628 (hex)	VREF	Variable rw	Speed for Homing
3629 (hex)	VSCALE1	Variable rw	SW1 Velocity Scaling Factor
362C (hex)	DILIM	Variable rw	DPR current limit
362D (hex)	DENA	Variable rw	DPR software disable reset mode
3630 (hex)	INPT0	Variable rw	In-Position Delay
3636 (hex)	WPOS	Variable ro	Enable Position Registers
3637 (hex)	SRND	Variable rw	Start Position of Modulo Axes
3638 (hex)	ERND	Variable rw	End position of modulo axes
363C (hex)	REFMODE	Variable rw	Source of the Zero Pulse in Homing Mode
363D (hex)	VLO	Variable rw	Software Resolver/Digital Converter Feed-forward
363E (hex)	WMASK	Variable rw	Warning as Fault Mask
363F (hex)	WPOSE	Variable ro	Enable Fast Position Registers 1 ... 16
3640 (hex)	WPOSP	Variable rw	Polarity of Fast Position Registers 1 ... 16
3641 (hex)	WPOSX	Variable rw	Mode of Fast Position Registers 1 ... 16
3642 (hex)	MOVE	Command	Start Motion Task
3643 (hex)	POSRSTAT	Variable rw	Status of Fast Position Registers 1 ... 16
3654 (hex)	PTARGET	Variable rw	Last Target Position
3656 (hex)	roFFSABS	Variable rw	Reference Offset
3658 (hex)	DPRILIMIT	Variable rw	Digital Limiting of the peak Current via DPR
3659 (hex)	ACCUNIT	Variable rw	Type of acceleration setpoint for the system
365A (hex)	VCOMM	Variable rw	Velocity Threshold for Commutation error
365B (hex)	MTMUX	Variable rw	Presetting for motion task that is processed later
365D (hex)	REFLS	Variable rw	
365F (hex)	VUNIT	Variable rw	System wide Definition of Velocity / Speed
3660 (hex)	PUNIT	Variable rw	Set Resolution of the Position
366E (hex)	TBRAKE	Variable rw	Disable Delay time with Holding Brake
366F (hex)	TBRAKE0	Variable rw	Enable Delay time with Holding Brake
3671 (hex)	MSLBRAKE	Variable rw	DEC ramp at sensorless emergency stop
3675 (hex)	ESPEED	Variable ro	Maximum velocity corresponding to the Feedback Type
367F (hex)	LATCH1P16	Variable rw	Latched 16-bit Position (positive edge)
3680 (hex)	LATCH1N16	Variable rw	Latched 16-bit Position (negative edge)
3681 (hex)	EXTLATCH	Variable rw	Selection of the Source of the Latch Inputs
3682 (hex)	STAGECODE	Variable ro	Power Stage Identification
3683 (hex)	SYNCSRC	Variable rw	

CAN Object Number	Parameter Name	Parameter Type	Description
3698 (hex)	VREF0	Variable rw	Homing Mode Reduction factor
36CE (hex)	LASTWMASK	Variable ro	Fault history of WMASK
36D2 (hex)	NREFMT	Variable rw	Homing with following motion task
36D7 (hex)	AUTOHOME	Variable rw	
36D8 (hex)	PASSCNFG	Variable rw	Password Function
36E4 (hex)	DRVCNFG2	Variable rw	Additional drive functions
38e8 (hex)	FwVersion	Variable-ro	Firmware version
38E9 (hex)	Revision	Variable-ro	Firmware revision index
38EC (hex)	Model	Variable-ro	S200 Model
38ED (hex)	Build	Variable-ro	Firmware build number
38EE (hex)	SerialTimeOut	Variable-rw	Serial communication timeout value [ms]
38F0 (hex)	MName	Variable-rw	Motor name
38F1 (hex)	KVP	Variable-rw	Velocity loop proportional gain
38F2 (hex)	KVI	Variable-rw	Velocity loop break out frequency
38F3 (hex)	ILmtPlus	Variable-rw	Upper current limit
38F4 (hex)	ILmtMinus	Variable-rw	Lower current limit
38F5 (hex)	buErrcode	Variable-ro	Error codes for base unit faults
38F6 (hex)	ocErrcode	Variable-ro	Error codes for Option Card faults
38F7 (hex)	WarningCode	Variable-ro	Warning Code from common functions
38F8 (hex)	CommMode	Variable-rw	Commutation feedback type
38F9 (hex)	DPoles	Variable-rw	Motor pole count
38FA (hex)	EncIn1Lines	Variable-rw	Encoder 1 lines
38FC (hex)	HallState	Variable-ro	Current state of Halls
38FD (hex)	CommOffset	Variable-rw	Commutation offset
3900 (hex)	MotorConfigured	Variable-rw	Motor configured flag
3901 (hex)	KPP	Variable-rw	Position Loop Proportional Gain
3902 (hex)	KVFF	Variable-rw	Position loop velocity feed forward
3903 (hex)	KTFF	Variable-rw	Position loop torque feed forward
3A47 (hex)	ReceiptionTimeout	Variable-rw	ModBus timeout while receiving a telegram
3A49 (hex)	**CAN**BaudRate	Variable-rw	**CAN** baud rate
3B40 (hex)	MTemp(BU[PK1])	Variable-ro	Motor Temperature
3B41 (hex)	EMUFunc(BU)	Variable-rw	Emulator A and B I/O mode
3B42 (hex)	CommMode(BU)	Variable-rw	Commutation Mode
3B43 (hex)	CmdSrc(BU)	Variable-rw	Select source of command
3B48 (hex)	DM1Map(BU)	Variable-rw	Selects variable send to DAC monitor 1
3B4A (hex)	DM2Map(BU)	Variable-rw	Selects variable send to DAC monitor 2
3B4B (hex)	Baseunit.Model(BU)	Variable-ro	Base unit Model Number
3B51 (hex)	ILmtPlus(BU)	Variable-rw	Current limit clock-wise
3B52 (hex)	FltDiag(BU)	Variable-rw	Fault diagnostic output
3B54 (hex)	EncOut(BU)	Variable-rw	Emulated encoder output line count
3B5F (hex)	SetupS2(BU)	Variable-ro	State of 4 position DIP switch S2
3B60 (hex)	SetupS1(BU)	Variable-ro	State of rotary switch S1
3B62 (hex)	VelFB(BU)	Variable-ro	Velocity feedback value
3B63 (hex)	LogicVer(BU)	Variable-ro	Version of the drive logic
3B6A (hex)	HSTemp(BU)	Variable-ro	Heat sink temperature
3B71 (hex)	MTempLimit(BU)	Variable-rw	Motor over temperature limits
3B72 (hex)	MTempLimitPol(BU)	Variable-rw	Polarity of MTemp fault comparison

CAN Object Number	Parameter Name	Parameter Type	Description
3B78 (hex)	VBus(BU)	Variable-ro	Bus voltage
3B7F (hex)	ITMode(BU)	Variable-rw	I*t fault mode
3B89 (hex)	IqFB(BU)	Variable-ro	Iq current feedback
3B8A (hex)	IdFB(BU)	Variable-ro	Id current feedback
3B8B (hex)	Velocity(BU)	Variable-ro	Filtered shaft velocity feedback

5. OBJECT DICTIONARY

The following table defines the Object Dictionary.

Index	Sub-Index	Data Type	Access	PDO mapp.	Description
1000h	0	UNSIGNED32	ro	—	Device type
1001h	0	UNSIGNED8	ro	—	Error register
1002h	0	UNSIGNED32	ro	yes	Manufacturer-specific status register
1003h		ARRAY			Pre-defined error field
1003h	0	UNSIGNED8	rw	—	Number of errors
1003h	1...8	UNSIGNED32	ro	—	Standard error field
1005h	0	UNSIGNED32	rw	—	COB—ID SYNC message
1006h	0	UNSIGNED32	rw	No	Communication cycle period
1008h	0	Visible String	ro	—	Manufacturer device name
1009h	0	Visible String	ro	—	Manufacturer hardware version
100Ah	0	Visible String	ro	—	Manufacturer software version
100Ch	0	UNSIGNED16	rw	—	Guard time
100Dh	0	UNSIGNED8	rw	—	Lifetime factor
1010h	0	UNSIGNED8	ro	—	Number of entries
1010h	1	UNSIGNED32	rw	—	Save all parameters
1011h	0	UNSIGNED8	ro	—	Number of entries
1011h	1	UNSIGNED32	rw	—	Restore all default parameters
1014h	0	UNSIGNED32	rw	—	COB—ID for the Emergency Object
1016h		RECORD			Consumer heartbeat time
1016h	0	UNSIGNED8	ro	—	Number of entries
1016h	1	UNSIGNED32	rw	—	Consumer heartbeat time
1017h	0	UNSIGNED16	rw	—	Producer heartbeat time
1018h		RECORD			Identity Object
1018h	0	UNSIGNED8	ro	—	Number of entries
1018h	1	UNSIGNED32	ro	—	Vendor ID
1018h	2	UNSIGNED32	ro	—	Product Code
1018h	3	UNSIGNED32	ro	—	Revision number
1018h	4	UNSIGNED32	ro	—	Serial number
1026h		ARRAY			OS prompt
1026h	0	UNSIGNED8	ro	—	Number of entries
1026h	1	UNSIGNED8	wo	—	StdIn
1026h	2	UNSIGNED8	ro	—	StdOut
1400h		RECORD			RXPDO1 communication parameter
1400h	0	UNSIGNED8	ro	—	Number of entries
1400h	1	UNSIGNED32	rw	—	RXPDO1 COB — ID
1400h	2	UNSIGNED8	rw	—	Transmission type RXPDO1
1401h		RECORD			RXPDO2 communication parameter
1401h	0	UNSIGNED8	ro	—	Number of entries

Index	Sub-Index	Data Type	Access	PDO mapp.	Description
1401h	1	UNSIGNED32	rw	—	RXPDO2 COB — ID
1401h	2	UNSIGNED8	rw	—	Transmission type RXPDO2
1402h		RECORD			RXPDO3 communication parameter
1402h	0	UNSIGNED8	ro	—	Number of entries
1402h	1	UNSIGNED32	rw	—	RXPDO3 COB — ID
1402h	2	UNSIGNED8	rw	—	Transmission type RXPDO3
1403h		RECORD			RXPDO4 communication parameter
1403h	0	UNSIGNED8	ro	—	Number of entries
1403h	1	UNSIGNED32	rw	—	RXPDO4 COB — ID
1403h	2	UNSIGNED8	rw	—	Transmission type RXPDO4
1600h		RECORD			RXPDO1 mapping parameter
1600h	0	UNSIGNED8	ro	—	Number of entries
1600h	1...8	UNSIGNED32	rw	—	Mapping for n—th application object
1601h		RECORD			RXPDO2 mapping parameter
1601h	0	UNSIGNED8	ro	—	Number of entries
1601h	1...8	UNSIGNED32	rw	—	Mapping for n—th application object
1602h		RECORD			RXPDO3 mapping parameter
1602h	0	UNSIGNED8	ro	—	Number of entries
1602h	1...8	UNSIGNED32	rw	—	Mapping for n—th application object
1603h		RECORD			RXPDO4 mapping parameter
1603h	0	UNSIGNED8	ro	—	Number of entries
1603h	1...8	UNSIGNED32	rw	—	Mapping for n—th application object
1800h		RECORD			TXPDO1 communication parameter
1800h	0	UNSIGNED8	ro	—	Number of entries
1800h	1	UNSIGNED32	rw	—	TXPDO1 COB—ID
1800h	2	UNSIGNED8	rw	—	Transmission type TXPDO1
1800h	3	UNSIGNED16	rw	—	Inhibit time
1800h	5	UNSIGNED16	rw	—	event timer
1801h		RECORD			TXPDO2 communication parameter
1801h	0	UNSIGNED8	ro	—	Number of entries
1801h	1	UNSIGNED32	rw	—	TXPDO2 COB—ID
1801h	2	UNSIGNED8	rw	—	Transmission type TXPDO2
1801h	3	UNSIGNED16	rw	—	Inhibit time
1801h	5	UNSIGNED16	rw	—	event timer
1802h		RECORD			TXPDO3 communication parameter
1802h	0	UNSIGNED8	ro	—	Number of entries
1802h	1	UNSIGNED32	rw	—	TXPDO3 COB—ID
1802h	2	UNSIGNED8	rw	—	Transmission type TXPDO3
1802h	3	UNSIGNED16	rw	—	Inhibit time
1802h	5	UNSIGNED16	rw	—	event timer
1803h		RECORD			TXPDO4 communication parameter

Index	Sub-Index	Data Type	Access	PDO mapp.	Description
1803h	0	UNSIGNED8	ro	—	Number of entries
1803h	1	UNSIGNED32	rw	—	TXPDO4 COB—ID
1803h	2	UNSIGNED8	rw	—	Transmission type TXPDO4
1803h	3	UNSIGNED16	rw	—	Inhibit time
1803h	5	UNSIGNED16	rw	—	event timer
1A00h		RECORD			Mapping parameter TXPDO1
1A00h	0	UNSIGNED8	ro	—	Number of entries
1A00h	1...8	UNSIGNED32	rw	—	Mapping for n—th application object
1A01h		RECORD			Mapping parameter TXPDO2
1A01h	0	UNSIGNED8	ro	—	Number of entries
1A01h	1...8	UNSIGNED32	rw	—	Mapping for n—th application object
1A02h		RECORD			Mapping parameter TXPDO3
1A02h	0	UNSIGNED8	ro	—	Number of entries
1A02h	1...8	UNSIGNED32	rw	—	Mapping for n—th application object
1A03h		RECORD			Mapping parameter TXPDO4
1A03h	0	UNSIGNED8	ro	—	Number of entries
1A03h	1...8	UNSIGNED32	rw	—	Mapping for n—th application object
2000h	0	UNSIGNED32	ro	yes	Manufacturer warnings
2040h		RECORD			Gearing factors for electronic gearing
2040h	0	UNSIGNED8		—	Number of entries
2040h	1	INTEGER32	rw	yes	Input factor for electronic gearing
2040h	2	UNSIGNED32	rw	yes	Output factor for electronic gearing
2080h	0	UNSIGNED16	rw	yes	Motion task for profile position mode
2081h	0	UNSIGNED16	rw	yes	Active motion task display
2082h	0	UNSIGNED32	wo	—	Copy motion task
2083h	0	UNSIGNED32	wo	—	Erase flash motion tasks
3500h		RECORD			ASCII Command MAXCMD
3500h	0	UNSIGNED8	ro	—	Number of entries
3500h	1	UNSIGNED32	ro	—	Value
3500h	2	UNSIGNED32	ro	—	Lower limit value
3500h	3	UNSIGNED32	ro	—	Upper limit value
3500h	4	UNSIGNED32	ro	—	Default value
3500h	5	UNSIGNED32	ro	—	Parameter format
3500h	6	UNSIGNED32	ro	—	Parameter control data
3CCF		UNSIGNED32	ro	—	Last entry of configuration parameter channel
4000h	0	UNSIGNED8	ro	—	Number of entries
4000h	1	UNSIGNED8	ro	—	Command status
4000h	2	UNSIGNED16	wo	—	Clear Motion Task
4000h	3	UNSIGNED8	wo	—	Clear All Motion Tasks
4000h	4	UNSIGNED8	wo	—	Default parameters

Index	Sub-Index	Data Type	Access	PDO mapp.	Description
4000h	5	UNSIGNED8	wo	—	Store Parameters
4000h	6	UNSIGNED8	wo	—	Restore parameters
4000h	7	UNSIGNED8	wo	—	Coldstart Drive
3500h+MAXCMD		RECORD			Last entry of the ASCII Object Channel
6040h	0	UNSIGNED16	wo	yes	Control word
6041h	0	UNSIGNED16	ro	yes	Status word
6060h	0	INTEGER8	rw	yes	Modes of Operation
6061h	0	INTEGER8	ro	yes	Modes of Operation Display
6063h	0	INTEGER32	ro	yes	Position actual value (increments)
6064h	0	INTEGER32	ro	yes	Position actual value (position units)
6065h	0	UNSIGNED32	rw	—	Following error window
6067h	0	UNSIGNED32	rw	—	Position window
6068h	0	UNSIGNED16	rw	—	Position window time
606Ch	0	INTEGER32	ro	yes	Velocity actual value
6071h	0	INTEGER16	rw	yes	Target torque
6077h	0	INTEGER16	ro	yes	Torque actual value
607Ah	0	INTEGER32	rw	yes	Target position
607Ch	0	INTEGER32	rw	—	Reference offset
607Fh	0	UNSIGNED32	rw	—	Max profile velocity
6080h	0	UNSIGNED32	rw	—	Max motor speed
6081h	0	UNSIGNED32	rw	yes	Profile Velocity
6083h	0	UNSIGNED32	rw	yes	Profile Acceleration
6084h	0	UNSIGNED32	rw	yes	Profile Deceleration
6085h	0	UNSIGNED32	rw	—	Quick stop deceleration
6086h	0	INTEGER16	rw	yes	Motion profile type
6089h	0	INTEGER8	rw	—	Position Notation index
608Ah	0	UNSIGNED8	rw	—	Position Dimension index
608Bh	0	INTEGER8	rw	—	Velocity Notation index
608Ch	0	UNSIGNED8	rw	—	Velocity Dimension index
608Dh	0	INTEGER8	rw	—	Acceleration Notation index
608Eh	0	UNSIGNED8	rw	—	Acceleration Dimension index
608Fh		ARRAY			Position encoder resolution
608Fh	0	UNSIGNED8	ro	—	Number of entries
608Fh	1	UNSIGNED32	rw	—	Encoder increments
608Fh	2	UNSIGNED32	rw	—	Motor revolutions
6090h		ARRAY			Velocity encoder resolution
6090h	0	UNSIGNED8	ro	—	Number of entries
6090h	1	UNSIGNED32	rw	—	Encoder increments per second
6090h	2	UNSIGNED32	rw	—	Motor revolutions per second
6091h		ARRAY			Gear ratio
6091h	0	UNSIGNED8	ro	—	Number of entries

Index	Sub-Index	Data Type	Access	PDO mapp.	Description
6091h	1	UNSIGNED32	rw	—	Motor revolutions
6091h	2	UNSIGNED32	rw	—	Shaft revolutions
6092h		ARRAY			Feed constant
6092h	0	UNSIGNED8	ro	—	Number of entries
6092h	1	UNSIGNED32	rw	—	Feed
6092h	2	UNSIGNED32	rw	—	Shaft revolutions
6093h		ARRAY			Position factor
6093h	0	UNSIGNED8	ro	—	Number of entries
6093h	1	UNSIGNED32	rw	—	Numerator
6093h	2	UNSIGNED32	rw	—	Feed constant
6094h		ARRAY			Velocity encoder factor
6094h	0	UNSIGNED8	ro	—	Number of entries
6094h	1	UNSIGNED32	rw	—	Numerator
6094h	2	UNSIGNED32	rw	—	Denominator
6097h		ARRAY			Acceleration factor
6097h	0	UNSIGNED8	ro		Number of entries
6097h	1	UNSIGNED32	rw	—	Numerator
6097h	2	UNSIGNED32	rw	—	Denominator
6098h	0	INTEGER8	rw	—	Homing type
6099h		ARRAY			Homing velocity
6099h	0	UNSIGNED8	ro	—	Number of entries
6099h	1	UNSIGNED32	rw	—	Speed while searching for limit switch
6099h	2	UNSIGNED32	rw	—	Speed while searching for zero mark
609Ah	0	UNSIGNED32	rw	—	Homing acceleration
60C0h	0	INTEGER8	rw	—	Interpolation sub mode select
60C1h		ARRAY			Interpolation data record
60C1h	0	UNSIGNED8	ro	—	Number of entries
60C1h	1	INTEGER32	rw	yes	x1, first parameter of ip function
60C2h		RECORD			Interpolation time period
60C2h	0	UNSIGNED8	ro	—	Number of entries
60C2h	1	UNSIGNED8	rw	—	Interpolation time units
60C2h	2	INTEGER16	rw	—	Interpolation time index
60C3h		ARRAY			Interpolation sync definition
60C3h	0	UNSIGNED8	ro	—	Number of entries
60C3h	1	UNSIGNED8	rw	—	Synchronize on group
60C3h	2	UNSIGNED8	rw	—	ip sync every n event
60C4h		RECORD			Interpolation data configuration
60C4h	0	UNSIGNED8	ro	—	Number of entries
60C4h	1	UNSIGNED32	ro	—	Maximum buffer size
60C4h	2	UNSIGNED32	rw	—	Actual buffer size
60C4h	3	UNSIGNED8	rw	—	Buffer organization

Index	Sub-Index	Data Type	Access	PDO mapp.	Description
60C4h	4	UNSIGNED16	rw	—	Buffer position
60C4h	5	UNSIGNED8	wo	—	Size of data record
60C5h	0	UNSIGNED32	rw	—	Max acceleration /deceleration
60C4h	6	UNSIGNED8	wo	—	Buffer clear
60FDh	0	UNSIGNED32	ro	yes	Digital inputs
60FFh	0	INTEGER32	rw	yes	Target velocity
6502h	0	UNSIGNED32	ro	—	Supported drive modes

6. OBJECTS

6. 1 SDO 1A00-1A03H: 1ST - 4TH TXPDO MAPPING PARAMETER (DS301)

Index	**$1A00_h$ - $1A03_h$ for TXPDO 1 .. 4**
Name	Transmit PDO mapping
Object Code	RECORD
Data Type	PDO Mapping
Category	mandatory

Subindex	**0**
Name	Number of mapped application objects in PDO
Category	mandatory
Access	rw
PDO Mapping	not possible
Value Range	0: PDO is not active 1 - 8: PDO activated, mappings are taken only byte-wise
Default Value	PDO1: 1 PDO2: 2 PDO3: 2 PDO4: 2

Subindex	**1**
Name	PDO - mapping for the n-th application object
Category	Conditional, depends on number and size of object be mapped
Access	rw
PDO Mapping	not possible
Value Range	UNSIGNED32
Default Value	See 2.4.6, Process Data Object (PDO)

Subindex	**2**
Name	PDO - mapping for the n-th application object
Category	Conditional, depends on number and size of object be mapped
Access	rw
PDO Mapping	not possible
Value Range	UNSIGNED32
Default Value	See2.4.6, Process Data Object (PDO)

Subindex	**3**
Name	PDO - mapping for the n-th application object
Category	Conditional, depends on number and size of object be mapped
Access	rw
PDO Mapping	not possible
Value Range	UNSIGNED32
Default Value	See 2.4.6, Process Data Object (PDO)

Subindex	**4**
Name	PDO - mapping for the n-th application object
Category	Conditional, depends on number and size of object be mapped
Access	rw
PDO Mapping	not possible
Value Range	UNSIGNED32
Default Value	See 2.4.6, Process Data Object (PDO)

Subindex	**5**
Name	PDO - mapping for the n-th application object
Category	Conditional, depends on number and size of object be mapped
Access	rw
PDO Mapping	not possible
Value Range	UNSIGNED32
Default Value	See 2.4.6, Process Data Object (PDO)

Subindex	**6**
Name	PDO - mapping for the n-th application object
Category	Conditional, depends on number and size of object be mapped
Access	rw
PDO Mapping	not possible
Value Range	UNSIGNED32
Default Value	See 2.4.6, Process Data Object (PDO)

Subindex	**7**
Name	PDO - mapping for the n-th application object
Category	Conditional, depends on number and size of object be mapped
Access	rw
PDO Mapping	not possible
Value Range	UNSIGNED32
Default Value	See 2.4.6, Process Data Object (PDO)

Subindex	**8**
Name	PDO - mapping for the n-th application object
Category	Conditional, depends on number and size of object be mapped
Access	rw
PDO Mapping	not possible
Value Range	UNSIGNED32
Default Value	See 2.4.6, Process Data Object (PDO)

6. 2 SDO 100AH: MANUFACTURER SOFTWARE VERSION (DS301)

The object contains the manufacturer software version (here: the **CAN**open-part of the drive firmware).

Index	100Ah
Name	Manufacturer Software Version
Object code	VAR
Data type	Visible String
Category	Optional
Access	const
PDO mapping	not possible
Value range	0.01 ... 9.99
Default value	no

6. 3 SDO 100CH: GUARD TIME (DS301)

The arithmetical product of the SDOs 100Ch Guard Time and 100Dh Lifetime Factor is the response monitoring time. The Guard Time is given in milliseconds. The response monitoring is activated with the first Nodeguard object. If the value of the object Guard Time is set to zero, then the response monitoring is inactive.

Index	100Ch
Name	Guard Time
Object code	VAR
Data type	UNSIGNED16
Category	Conditional; mandatory, if heartbeat not supported
Access	rw
PDO mapping	not possible
Value range	UNSIGNED16
Default value	0

6. 4 SDO 100DH: LIFETIME FACTOR (DS301)

The product of Guard Time and Life Time Factor gives the life time for the nodeguarding protocol. If it's 0, the protocol is not used.

Index	100Dh
Name	Lifetime Factor
Object code	VAR
Data type	UNSIGNED8
Category	Conditional; mandatory, if heartbeat not supported
Access	rw
PDO mapping	not possible
Value range	UNSIGNED8
Default value	0

6. 5 SDO 1000H: DEVICE TYPE (DS301)

This object describes the device type (servo drive) and device functionality (DS402 drive profile).
Definition:

MSB				LSB
Additional information				Device profile number
Mode bits		Type		$402_d=192_h$
31 24		23 16		15 0

The device profile number is DS402, the type is 2 for servo amplifiers, the mode bits 28 to 31 are manufacturer specific and may be changed from its actual value of 0. A read access delivers 0x00002192 at the moment.

Index	1000_h
Name	Device type
Object code	VAR
Data type	UNSIGNED32
Category	mandatory
Access	ro
PDO mapping	not possible
Value range	UNSIGNED32
Default value	no

6. 6 SDO 1001H: ERROR REGISTER (DS301)

This object is an error register for the device. The device can map internal errors into this byte. It is a part of an Emergency object.

Index	1001h
Name	Error register
Object code	VAR
Data type	UNSIGNED8
Category	mandatory
Access	ro
PDO mapping	not possible
Value range	UNSIGNED8
Default value	no

Error reasons to be signaled: If a bit is set to 1 the specified error has occurred. The generic error is signaled at any error situation.

Bit	Description
0	generic error
1	current
2	voltage
3	temperature

Bit	Description
4	communication error (overrun, error state)
5	device profile specific
6	reserved (always 0)
7	manufacturer specific

6. 7 SDO 1002H: MANUFACTURER STATUS REGISTER (DS301)

The manufacturer status register contains important drive information.

Index 1002h
Name Manufacturer Status Register
Object code VAR
Data type UNSIGNED32
Category optional

Access ro
PDO mapping possible
Value range UNSIGNED32
Default value no

The following table shows the bit assignment for the status register:

Bit	Description
0	1 = Movement (positioning, homing) active
1	1 = reference point set
2	1 = reference switch high (home-position)
3	1 = In Position
4	1 = Position latch at input 2 (positive transition)
5	1 = Position register 0
6	1 = Position register 1
7	1 = Position register 2
8	1 = Position register 3
9	1 = Position register 4
10	1 = Initialization phase finished
11	1 = Position register 5
12	1 = Motor stand still message (threshold VEL0)
13	1 = Safety relay selected (AS-option)
14	1 = Power stage enabled
15	1 = Error state
16	1 = Homing move active
17	1 = Jog move active
18	1 = position latch at input 2 (negative transition)
19	1 = Emergency stop active
20	1 = Position latch at input 1 (positive transition)
21	1 = Position latch at input 1 (negative transition)
22	1 = Feed forward off
23	1 = Homing move finished
24	1 = one of the actual errors leads to a Coldstart of the drive, if reset
25	1 = digital input 1 set
26	1 = digital input 2 set
27	1 = digital input 3 set
28	1 = digital input 4 set
29	1 = digital input hardware enable set
30	reserved
31	reserved

6. 8 SDO 1003H: PREDEFINED ERROR FIELD (DS301)

SDO 1003h provides an error history with a maximum size of 8 entries. Subindex 0 contains the number of errors that have occurred since the last reset of the error history, either by startup of the drive or resetting the error history by writing 0 to Subindex 0. A new Emergency-message is written into Subindex 1 shifting the old entries one Subindex higher. The old content of Subindex 8 is lost. The UNSIGNED32-information written to the sub indices is defined in the field Error Code in the description of the Emergency Messages.

Index	1003h
Name	pre-defined Error Field
Object code	ARRAY
Data type	UNSIGNED32
Category	optional

Subindex	0
Description	Number of entries
Category	mandatory
Access	rw
PDO mapping	not possible
Value range	0 ... 8
Default value	0

Subindex	1
Description	Standard error field
Category	optional
Access	ro
PDO mapping	not possible
Value range	UNSIGNED32
Default value	no

Subindex	2
Description	Standard error field
Category	optional
Access	ro
PDO mapping	not possible
Value range	UNSIGNED32
Default value	no

Subindex	3
Description	Standard error field
Category	optional
Access	ro
PDO mapping	not possible
Value range	UNSIGNED32
Default value	no

Subindex	4
Description	Standard error field
Category	optional
Access	ro
PDO mapping	not possible
Value range	UNSIGNED32
Default value	no

Subindex	5
Description	Standard error field
Category	optional
Access	ro
PDO mapping	not possible
Value range	UNSIGNED32
Default value	no

Subindex	6
Description	Standard error field
Category	optional
Access	ro
PDO mapping	not possible
Value range	UNSIGNED32
Default value	no

Subindex	7
Description	Standard error field
Category	optional
Access	ro
PDO mapping	not possible
Value range	UNSIGNED32
Default value	no

Subindex	8
Description	Standard error field
Category	optional
Access	ro
PDO mapping	not possible
Value range	UNSIGNED32
Default value	no

6. 9 SDO 1005H: COB-ID OF THE SYNC MESSAGE (DS301)

This object defines the COB-Id of the synchronization object (SYNC).

Index 1005h
Name COB-ID for the SYNC message
Object code VAR
Data type UNSIGNED32
Category Conditional

Access rw
PDO mapping not possible
Value range UNSIGNED32
Default value no

Bit coded information:

Bit	Value	Meaning
31 (MSB)	X	—
30	0	Device not generate SYNC message
	1	Device generates SYNC message
29	0	11 Bit ID (**CAN** 2.0A)
	1	29 Bit ID (**CAN** 2.0B)
28 ... 11	X	if Bit 29=1 => Bit 11 ... 28 of 29-bit SYNC COB-ID
	0	if Bit 29=0
10 ... 0 (LSB)	X	Bit 0 ... 10 of SYNC COB-ID

The device does not support the generation of SYNC-messages and only the 11-bit IDs. So the bits 11 to 30 are always 0.

6. 10 SDO 1006H: COMMUNICATION CYCLE PERIOD (DS301)

This object can be used to define the period (in µs) for the transmission of the SYNC telegram.

Index 1006h
Name Period of the communication cycle
Object code VAR
Data type UNSIGNED32
Category O

Access rw
PDO mapping not possible
Value range UNSIGNED32
Default value 00_h

6. 11 SDO 1008H: MANUFACTURER DEVICE NAME (DS301)

The device name consists of four ASCII characters in the form S2*xx*, where *xx* is the power stage current.

Index	1008h
Name	Manufacturer Device Name
Object code	VAR
Data type	Visible String
Category	Optional
Access	const
PDO mapping	not possible
Value range	S301 - S3xx
Default value	N/A

6. 12 SDO 1009H: MANUFACTURER HARDWARE VERSION

The object gives the layout version of the drive.

Index	1009h
Name	Manufacturer hardware version
Object code	VAR
Data type	Visible String
Category	Optional
Access	const
PDO mapping	not possible
Value range	-
Default value	N/A

6. 13 SDO 1010H: STORE PARAMETERS (DS301)

This object supports the saving of parameters to a flash EEProM. Only the Subindex 1 for saving of all parameters, which can also be saved in the parameter files via the GUI, is supported.

Index	1010h
Name	Store parameters
Object code	ARRAY
Data type	UNSIGNED32
Category	optional
Subindex	0
Name	Number of entries
Object code	VAR
Data type	UNSIGNED8
Category	mandatory
Access	ro
PDO Mapping	not possible
Value range	1
Default value	1
Subindex	1
Name	Save all parameters
Object code	VAR
Data type	UNSIGNED32
Category	mandatory
Access	rw
PDO Mapping	not possible
Value range	UNSIGNED32
Default value	1

Data definition:

Bit number	Value	Meaning
31 ... 2	0	reserved (=0)
1	0	Device does not save parameters autonomously
	1	Device does save parameters autonomously
0	0	Device does not save parameters on command
	1	Device does not save parameters on command

By read access to sub-index 1 the drive provides information about its storage functionality.

This drive provides a constant value of 1 by read access, i.e., all parameters can be saved by writing to SDO 1010 sub 1. In general the drive doesn't save parameters autonomously with the exception of e.g. the special treatment of the homing of multi-turn absolute encoders.

Storing of parameters is only done if a special signature ("save") is written to Subindex 1. "save" is equivalent to the unsigned32 - number 65766173_h.

6. 14 SDO 1011H: RESTORE DEFAULT PARAMETERS

With this object, the default values of parameters according to the communication or device profile are restored. The S200 allows you to restore all default values.

Index	1011h
Name	Restore default parameters
Object code	ARRAY
Data type	UNSIGNED32
Category	optional

Subindex	0
Name	Number of entries
Entry Category	mandatory
Access	ro
PDO Mapping	not possible
Default value	1

Subindex	1
Name	Restore all default parameters
Entry Category	mandatory
Access	rw
PDO Mapping	not possible
Value range	UNSIGNED32
Default value	1 (device restores parameter)

Restoring of parameters is only done, if a special signature (“load”) is written to subindex 1. “load” has to be transmitted as unsigned32 - number $64616F6C_h$.

6. 15 SDO 1014H: COB-ID FOR EMERGENCY MESSAGE (DS301)

This object defines the COB-ID of the Emergency message.

Index	1014h
Name	COB-ID emergency message
Object code	VAR
Data type	UNSIGNED32
Category	conditional; mandatory, if Emergency is supported

Access	rw
PDO mapping	not possible
Value range	UNSIGNED32
Default value	80_h + Node - ID

6. 16 SDO 1016H: CONSUMER HEARTBEAT TIME

The consumer heartbeat time defines the expected heartbeat cycle time and has to be higher than the corresponding producer heartbeat time configured on the device producing this heartbeat. Monitoring starts after the reception of the first heartbeat. If the consumer heartbeat time is 0 the corresponding entry is not used. The time is defined in milliseconds.

Index	1016h
Name	Consumer heartbeat time
Object code	ARRAY
Data type	UNSIGNED32
Category	optional

Subindex	0
Description	Number of entries
Category	mandatory
Access	ro
PDO Mapping	not possible
Value range	1
Default value	1

Subindex	1
Description	Consumer heartbeat time
Category	mandatory
Access	rw
PDO Mapping	not possible
Value range	unsigned 32
Default value	no

Definition of the entry value of sub-index 1

	MSB					LSB
Value	Reserved (value: 00)		Node-ID		Heartbeat time	
Encoded as	-		UNSIGNED8		UNSIGNED16	
Bit	31	24	23	16	15	0

6. 17 SDO 1017H: PRODUCER HEARTBEAT TIME

The producer heartbeat time defines the cycle time of the heartbeat in ms. If it's 0, it is not used.

Index	1017h
Name	Producer heartbeat time
Object code	VAR
Data type	UNSIGNED16
Category	conditional; mandatory, if guarding is not supported

Access	rw
PDO mapping	not possible
Value range	UNSIGNED16
Default value	0

6. 18 SDO 1018H: IDENTITY OBJECT (DS301)

The Identity Object contains general device information.

Index	1018h
Name	Identity Object
Object code	RECORD
Data type	Identity
Category	mandatory

Subindex	0
Description	Number of entries
Category	mandatory
Access	ro
PDO mapping	not possible
Value range	1 ... 4
Default value	4

Subindex 1 is a unique number for a device manufacturer.

Subindex	1
Description	Vendor ID
Category	mandatory
Access	ro
PDO mapping	not possible
Value range	UNSIGNED32
Default value	0x6AH ()

Subindex 2 contains the general device number (300) plus an information about DC-bus-voltage and current class.

Subindex	2
Description	Product Code
Category	optional
Access	ro
PDO mapping	not possible
Value range	301 ... 346
Default value	no

Subindex 3 consists of two revision numbers:
- the major revision number in the upper word containing the **CAN**-version
- the minor revision number containing the general firmware version

E.g. a value of 0x0022 0079 means **CAN**-version 0.34 and firmware version 1.21.

Subindex	3
Description	Revision Number
Category	optional
Access	ro
PDO mapping	not possible
Value range	UNSIGNED32
Default value	no

Subindex 4 gives the serial number of the drive.

Subindex	4
Description	Serial Number
Category	optional
Access	ro
PDO mapping	not possible
Value range	UNSIGNED32
Default value	no

6. 19 SDO 1026H: OS PROMPT

The OS prompt is used to build an ASCII - communication channel to the drive.

Index	1026h
Name	OS Prompt
Object code	ARRAY
Data type	UNSIGNED8
Category	optional

Subindex	0
Description	Number of entries
Category	mandatory
Access	ro
PDO mapping	not possible
Value range	2
Default value	2

Subindex 1 is used to send one character to the drive.

Subindex	1
Description	StdIn
Category	mandatory
Access	ro
PDO mapping	possible
Value range	UNSIGNED8
Default value	—

Subindex 2 is used to receive one character from the drive.

Subindex	2
Description	StdOut
Category	mandatory
Access	w
PDO mapping	possible
Value range	UNSIGNED8
Default value	0

6. 20 SDO 1400-1403H: 1ST - 4TH RXPDO COMMUNICATION PARAMETER (DS301)

Index	1400h ... 1403h for RXPDO 1 ... 4
Name	receive PDO parameter
Object code	RECORD
Data type	PDO CommPar
Category	mandatory
Subindex	**0**
Name	number of entries
Category	mandatory
Access	ro
PDO Mapping	not possible
Value Range	2
Default Value	2

Subindex	**1**
Name	COB-ID used by PDO
Category	mandatory
Access	rw
PDO Mapping	not possible
Value Range	UNSIGNED32
Default Value	Index 1400_h: 200_h + Node-ID Index 1401_h: 300_h + Node-ID Index 1402_h: 400_h + Node ID Index 1403_h: 500_h + Node-ID

Subindex	**2**
Name	transmission type
Category	mandatory
Access	rw
PDO Mapping	not possible
Value Range	UNSIGNED8
Default Value	FF_h

Subindex 1 contains the COB-Id of the PDO as a bit coded information:

Bit-Number	Value	Meaning
31	0	PDO exists / is valid
	1	PDO does not exist / is not valid
30	0	RTR allowed on this PDO, not supported
	1	RTR not allowed on this PDO, not supported
29	0	11 bit-ID (**CAN** 2.0A)
	1	29 bit-ID (**CAN** 2.0B), not supported
28 .. 11	X	Identifier-bits with 29 bit-ID, not relevant
10 .. 0	X	Bits 10-0 of COB-ID

Subindex 2 contains the transmission type of the PDO. There are two ways of setting:

— the value FFh or 255 for event-triggered PDO, which is directly interpreted by reception and taken into actions,

— values from 0 to 240, which cause a SYNC-telegram-controlled interpretation of the PDO contents. Values of 1 to 240 mean, that 0 to 239 SYNC-telegrams are ignored, before one is interpreted. The value 0 means, that only the next SYNC-telegram is interpreted.

6. 21 SDO 1600-1603H: 1ST - 4TH RXPDO MAPPING PARAMETER (DS301)

Index	**1600_h - 1603_h for RXPDO 1 .. 4**
Name	Receive PDO mapping
Object Code	RECORD
Data Type	PDO Mapping
Category	mandatory

Subindex	**0**
Name	Number of entries
Category	mandatory
Access	rw
PDO Mapping	not possible
Value Range	0: PDO is not active 1 - 8: PDO activated, mappings are taken only byte-wise
Default Value	PDO1: 1 PDO2: 2 PDO3: 2 PDO4: 2

Subindex	**1**
Name	PDO - mapping for the n-th application object
Category	Conditional, depends on number and size of object be mapped
Access	rw
PDO Mapping	not possible
Value Range	UNSIGNED32
Default Value	See below

Subindex	**2**
Name	PDO - mapping for the n-th application object
Category	Conditional, depends on number and size of object be mapped
Access	rw
PDO Mapping	not possible
Value Range	UNSIGNED32
Default Value	See 2.4.6, Process Data Object (PDO)

Subindex	**3**
Name	PDO - mapping for the n-th application object
Category	Conditional, depends on number and size of object be mapped
Access	rw
PDO Mapping	not possible
Value Range	UNSIGNED32
Default Value	See below

Subindex	**4**
Name	PDO - mapping for the n-th application object
Category	Conditional, depends on number and size of object be mapped
Access	rw
PDO Mapping	not possible
Value Range	UNSIGNED32
Default Value	2.4.6, Process Data Object (PDO)

Subindex	**5**
Name	PDO - mapping for the n-th application object
Category	Conditional, depends on number and size of object be mapped
Access	rw
PDO Mapping	not possible
Value Range	UNSIGNED32
Default Value	See 2.4.6, Process Data Object (PDO)

Subindex	**6**
Name	PDO - mapping for the n-th application object
Category	Conditional, depends on number and size of object be mapped
Access	rw
PDO Mapping	not possible
Value Range	UNSIGNED32
Default Value	See 2.4.6, Process Data Object (PDO)

Subindex	**7**
Name	PDO - mapping for the n-th application object
Category	Conditional, depends on number and size of object be mapped
Access	rw
PDO Mapping	not possible
Value Range	UNSIGNED32
Default Value	See 2.4.6, Process Data Object (PDO)

Subindex	**8**
Name	PDO - mapping for the n-th application object
Category	Conditional, depends on number and size of object be mapped
Access	rw
PDO Mapping	not possible
Value Range	UNSIGNED32
Default Value	See 2.4.6, Process Data Object (PDO)

6. 22 SDO 1800-1803H: 1ST - 4TH TXPDO COMMUNICATION PARAMETER (DS301)

Index	1800_h ... 1803h for TXPDO 1 ... 4
Name	transmit PDO parameter
Object code	RECORD
Data type	PDO CommPar
Category	mandatory

Subindex	**0**
Name	Number of entries
Category	mandatory
Access	ro
PDO Mapping	not possible
Value Range	5
Default Value	5

Subindex	**1**	
Name	COB-ID used by PDO	
Category	mandatory	
Access	rw	
PDO Mapping	not possible	
Value Range	UNSIGNED32	
Default Value	Index 1800_h: 180_h + Node-ID	Index 1801_h: 280_h + Node-ID
	Index 1802_h: 380_h + Node-ID	Index 1803_h: 480_h + Node-ID

Subindex	**2**
Name	Transmission type
Category	mandatory
Access	rw
PDO Mapping	not possible
Value Range	UNSIGNED8
Default Value	FF_h

Subindex	**3**
Name	Inhibit time
Category	optional
Access	rw
PDO Mapping	not possible
Value Range	UNSIGNED16 (n*1/10ms)
Default Value	0_h

Subindex	**4**
Name	Reserved
Category	optional
Access	rw
PDO Mapping	not possible
Value Range	0
Default Value	0

Subindex	**5**
Name	Event timer
Category	optional
Access	rw
PDO Mapping	not possible
Value Range	UNSIGNED16 (0=not used, n*1/10ms)
Default Value	0h

Subindex 1 contains the COB-Id of the PDO as a bit coded information:

Bit-Number	Value	Meaning
31	0	PDO exists / is valid
	1	PDO does not exist / is not valid
30	0	RTR allowed on this PDO, not supported
	1	RTR not allowed on this PDO, not supported
29	0	11 bit-ID (**CAN** 2.0A)
	1	29 bit-ID (**CAN** 2.0B), not supported
28 .. 11	X	Identifier-bits with 29 bit-ID, not relevant
10 .. 0	X	Bits 10-0 of COB-ID

Subindex 2 contains the transmission type of the PDO. There are two ways of setting:

— a value of FFh or 255 for an event-triggered PDO, which is sent immediately after a change in the mapped application objects. Setting of sub-index 3 or 5 has an influence on the sending of a PDO. With sub-index 3 you can configure, in which minimal time the so configured Transmit-PDOs are sent, if PDO-data contents change (reduction of bus-load). With sub-index 5 (event time) a timer is used, which is reset with every event-triggered sending of this PDO. If there is no change of the PDO-content in this time, the PDO is sent caused by this timer event.

— values from 0 to 240 cause a SYNC-Telegram controlled sending of the PDO. Values of 1 to 240 define how often the SYNC-telegram leads to a sending of a PDO. The value 0 means, that only the next SYNC-telegram leads to a sending of the so configured PDOs.

6. 23 SDO 2000H: MANUFACTURER WARNINGS

This object provides information about drive internal warnings.

Index	2000h
Name	Manufacturer warnings
Object code	VAR
Data type	UNSIGNED32
Category	optional
Access	ro
PDO mapping	possible
Value range	UNSIGND32
Default value	—

Bit coded warnings:

Bit Number	Meaning
0	n01 - I2t - threshold exhausted
1	n02 - Regen power reached preset regen power limit
2	n03* - Contouring error exceeded preset limit
3	n04* - Nodeguarding monitoring has been activated
4	n05 - Mains supply phase missing
5	n06* - Position fall below software limit switch 1
6	n07* - Position exceeded software limit switch 2
7	n08 -Faulty motion task
8	n09 -No reference point at start of motion task
9	n10* -PSTOP limit-switch activated
10	n11* -NSTOP limit-switch activated
11	Reserved
12	n13* - Expansion card not operating correctly
13	Reserved
14	n15 - Table error fault according to speed/current table (with INXMODE 35)
15	n16 - Summarized warning for n17 to n31
16	n17 - **CAN**-Sync is not logged in (with SYNCSRC = 3)

**these warning messages result in a controlled shut-down of the drive (braking by emergency stop ramp)*

6. 24 SDO 2040H: GEARING FACTOR FOR ELECTRONIC GEARING

This object defines the gearing factor for the electronic gearing between a master and a slave drive, which are connected via roD. These objects are relevant only for the OPMODE 4 resp. the **CAN**open-mode 0x84.

Index	2040h
Name	Electronic gearing factor
Object code	RECORD
Data type	UNSIGNED32
Category	optional

Subindex	0
Description	Number of entries
Category	optional
Access	ro
PDO mapping	not possible
Value range	2
Default value	2

Subindex 1 is related to the master input signal depending on selected feedback and GEARMODE.

Subindex	1
Description	Gearing Input
Category	optional
Access	rw
PDO mapping	possible
Value range	INTEGER32
Default value	1024

Subindex 2 gives the movement of the slave in dependency of the master pulses.

Subindex	2
Description	Gearing Output
Category	optional
Access	rw
PDO mapping	possible
Value range	UNSIGN32ED
Default value	1

6. 25 SDO 2080H: MOTION TASK FOR PROFILE POSITION MODE

This object is an extension to the profile position mode. If the value of the object is not 0, the addressed motion task will be started with the next rising flank of the "New set point" bit of the control word (bit 4), if the bit "Change Set Immediately" (bit 5) is set. After the motion task is started, the value of the object will be reset automatically to 0.

Index	2080h
Name	Motion tasks in PP-Mode
Object code	VAR
Data type	UNSIGNED16
Category	optional
Access	rw
PDO mapping	possible
Value range	1 ... 300
Default value	0

6. 26 SDO 2081H: ACTIVE MOTION TASK DISPLAY

This object shows the last motion task, which has been started in the drive. Motion tasks numbers from 1 to 200 show Flash-EEProM motion tasks, numbers from 201 to 300 show RAM-motion tasks. If there is no value in object 2080h and a motion task is started via the new-set point /set point acknowledge mechanism of the profile position mode, motion task 0 is used and shown.

If you start a set of stored motion tasks (bit 3 of the motion task control word O_C, SDO 35B9 sub1 set), the active motion task will be shown in this object.

Index	2081h
Name	Active motion task display
Object code	VAR
Data type	UNSIGNED16
Category	optional
Access	rw
PDO mapping	possible
Value range	1 ... 300
Default value	0

6. 27 SDO 2082H: COPY MOTION TASKS

With the help of this object motion tasks can be copied in the drive. The motion task addressed in the low word is copied to the motion task addressed in the high word.

EEProM motion task between 1 and 200 can be written only if the power stage is disabled!

Index	2082h
Name	Copy motion tasks
Object code	VAR
Data type	UNSIGNED32
Category	optional
Access	wo
PDO mapping	not possible
Value range	High Word: 0 .. 300, low word: 0 .. 300
Default value	-

6. 28 SDO 2083: DELETE MOTION TASKS

This object gives the possibility to delete all Flash-EEProM motion tasks. This action is only been taken, if a special signature (“prom”) is written. “prom” has to be transmitted as unsigned32 - number $6D6F7270_h$.

Deletion is only possible if the power stage is disabled and the NMT-state is PREOPERATIONAL.

Index	2083h
Name	Delete motion tasks
Object code	VAR
Data type	UNSIGNED32
Category	optional
Access	wo
PDO mapping	not possible
Value range	UNSIGNED32
Default value	-

6. 29 SDO>3500H MANUFACTURER SPECIFIC OBJECT CHANNEL

The Object Dictionary has been expanded beyond Index 3500h (reserved object range 3500h ... 3CD0h) for all Device Objects that can be described in up to 4 bytes of user data.

This range can be dynamically extended, i.e., if extensions are made, new device parameters that fulfill the above-mentioned format are **automatically** added to the table for the core firmware. SDO 3500_h (Subindex 01_h, read) can be used to show the total number of objects in the Object Channel.

Each object in this range is described with the aid of 8 Sub-indices. The structure is built up as follows:

Index	**> 3500_h**
Name	Object-dependent
Object code	VAR
Data type	RECORD
Subindex	**0**
Description	number of entries
Unit	—
Access	—
PDO mapping	not possible
Data type	UNSIGNED8
Value range	0 ... 28-1
EEProM	—
Default value	—
Subindex	**1**
Description	read/write a parameter
Unit	see corresponding ModBus parameter
Access	see corresponding ModBus parameter
PDO mapping	not possible
Data type	see corresponding ModBus parameter, transmission always as INTEGER32
Value range	see corresponding ModBus parameter
EEProM	see Sub -index 4
Default value	see corresponding ModBus parameter
Subindex	**2**
Description	read lower limit value
Unit	see corresponding ModBus parameter
Access	ro
PDO mapping	not possible
Data type	see corresponding ModBus parameter
Value range	see corresponding ModBus parameter
EEProM	—
Default value	—

Subindex	**3**
Description	read upper limit value
Unit	see corresponding ModBus parameter
Access	ro
PDO mapping	not possible
Data type	see corresponding ModBus parameter
Value range	see corresponding ModBus parameter
EEProM	—
Default value	—

Subindex	**4**
Description	read the default value
Unit	see corresponding ModBus parameter
Access	ro
PDO mapping	not possible
Data type	see corresponding ModBus parameter
Value range	see corresponding ModBus parameter
EEProM	—
Default value	—

Subindex	**5**
Description	read the parameter format
Unit	—
Access	ro
PDO mapping	not possible
Data type	see corresponding ModBus parameter
Value range	see corresponding ModBus parameter
EEProM	—
Default value	—

Possible parameter formats:

0	UNSIGNED8	7	Reserved
1	INTEGER8	8	Reserved
2	UNSIGNED16	9	Reserved
3	INTEGER16	10	Reserved
4	UNSIGNED32		
5	INTEGER32		
6	FLOAT32		

Subindex	**6**
Description	read the parameter attributes
Unit	—
Access	ro
PDO mapping	not possible
Data type	UNSIGNED32
Value range	0 ... 131072-1
EEProM	—
Default value	—

Description:

0x00000008	Variable is read only and must not be written to over the bus
0x00000020	Variable is volatile (NOT saved in the EEProM
0x00000040	Variable is a function, not a value
0x00008000	After an alteration the variable must be saved and the controller must be reset
0x00010000	Controller must be disabled before variable can be written to

Subindex	**7**
Description	reserved
Unit	—
Access	ro
PDO mapping	not possible
Data type	UNSIGNED32
Value range	0 ... 2^{32}-1
EEProM	—
Default value	—

6. 30 *SDO 4000H: EXECUTE COMMAND*

This SDO triggers the execution of S200 specific commands. The command engine accepts only one command at a time and rejects commands while being busy and returns the error code EC_COMMANDBUSY (FF040000h).

Index	**4000_h**
Name	Execute Command
Object code	VAR
Data type	RECORD

Subindex	**0**
Description	Number of entries
Unit	—
Access	—
PDO mapping	not possible
Data type	UNSIGNED8
Value range	7
EEProM	—
Default value	—

Returns the number of entries (sub-indices).

Subindex	**1**
Description	Command status
Unit	None
Access	ro
PDO mapping	not possible
Data type	UNSIGNED 8
Value range	0…
EEProM	—
Default value	—

Read the status of the last issued command. The command execution is in progress, if the returned value is non-zero.

Subindex	**2**
Description	Clear Motion Task
Unit	--
Access	wo
PDO mapping	not possible
Data type	UNSIGNED8
Value range	0…200
EEProM	—
Default value	—

Clears the motion task addressed by the number.

Subindex	**3**
Description	Clear All Motion Tasks
Unit	--
Access	wo
PDO mapping	not possible
Data type	USIGNED 8
Value range	0…255 (value has no affect!)
EEProM	—
Default value	—

Subindex	**4**
Description	Default Configuration Parameters
Unit	—
Access	R?W
PDO mapping	not possible
Data type	UNSIGNED 8
Value range	0…255 (Value has no effect)
EEProM	—
Default value	—

Subindex **5**
Description Store Configuration Parameters
Unit —
Access wo
PDO mapping not possible
Data type UNSIGNED 8
Value range 0…255 (Value has no effect)
EEProM —
Default value —

Subindex **6**
Description Restore Configuration Parameters
Unit —
Access wo
PDO mapping not possible
Data type UNSIGNED 8
Value range 0…255 (Value has no effect)
EEProM —
Default value —

Subindex **7**
Description Cold Start Drive
Unit —
Access wo
PDO mapping not possible
Data type UNSIGNED 8
Value range 0…255 (Value has no effect)
EEProM —
Default value —

Resets the drive and reboots the firmware.

6. 31 *SDO 60FD: DIGITAL INPUTS (DS402)*

This index defines simple digital inputs for drives. The bits 0 to 2 can be supported by the drive, if the needed function is configured to the digital inputs with the ASCII - commands INxMODE (x may be 1 to 4), e.g. IN3MODE = 2, PSTOP - function, see ASCII - manual).

Index 60FDh
Name Digital inputs
Object code VAR
Data type UNSIGNED32
Category optional

Access ro
PDO mapping possible
Value range UNSIGNED32
Default value 0

31 16	15 4	3	2	1	0
Manufacturer specific	Interlock	Interlock	Home switch	Pos. limit switch	Neg. limit switch

MSB LSB

The switch must be "active high".

6. 32 SDO 60FFH: TARGET VELOCITY (DS402)

The speed set point (*target velocity*) represents the set point for the ramp generator. The scaling of this value depends on the factor *velocity encoder resolution* (SDO 6094_h).

Index	**$60FF_h$**
Name	target velocity
Object code	VAR
Data type	INTEGER32
Mode	pv
Access	rw
PDO mapping	possible
Unit	increments
Value range	(-2^{31}) ... $(2^{31}-1)$
Default value	—
EEProM	no

6. 33 SDO 6040H: CONTROL WORD (DS402)

The control commands are built up from the logical combination of the bits in the control word and external signals (e.g., enable output stage). The definitions of the bits are shown below:

Index	**6040_h**
Name	Control word
Object code	VAR
Data type	UNSIGNED16
Mode	all
Access	rw
PDO mapping	possible
Unit	—
Value range	0 ... 65535
EEProM	no
Default value	0

Bit assignment im control word

Bit	Name
0	Switch on
1	Disable Voltage
2	Quick Stop
3	Enable Operation
4	Operation mode specific
5	Operation mode specific
6	Operation mode specific
7	Reset Fault (only effective for faults)

Bit	Name
8	Pause/halt
9	reserved
10	reserved
11	reserved
12	reserved
13	Manufacturer-specific
14	Manufacturer-specific
15	Manufacturer-specific

Commands in the control word

Command	Bit 7 Fault Reset	Bit 3 Enable Operation	Bit 2 Quick Stop	Bit 1 Disable Voltage	Bit 0 Switch on	Transitions
Shutdown	X	X	1	1	0	2, 6, 8
Switch on	X	X	1	1	1	3
Disable Voltage	X	X	X	0	X	7, 9, 10, 12
Quick Stop	X	X	0	1	X	7, 10, 11
Disable Operation	X	0	1	1	1	5
Enable Operation	X	1	1	1	1	4, 16
Fault Reset	1	X	X	X	X	15

Bits marked by an X are irrelevant.

Mode-dependent bits in the control word

The following table shows the mode-dependent bits in the control word. Only manufacturer-specific modes are supported at present. The individual modes are set by SDO 6060_h *Modes of operation*.

Operation mode	No.	Bit 4	Bit 5	Bit 6
Position	88_h	reserved	reserved	reserved
Digital speed	80_h	reserved	reserved	reserved
Digital current	82_h	reserved	reserved	reserved
Analog speed	81_h	reserved	reserved	reserved
Analog current	83_h	reserved	reserved	reserved
Profile Position Mode (pp)	01_h	new_set_point	change_set_immediately	absolute / relative
Profile Velocity Mode (pv)	03_h	reserved	reserved	reserved
Profile Torque Mode (tq)	04_h	reserved	reserved	reserved
Homing Mode (hm)	06_h	homing_operation_start	reserved	reserved
Interpolated Position Mode (ip)	07_h		reserved	reserved

Description of the remaining bits in the control word

The remaining bits in the control word are described below.

Bit 8 Pause If Bit 8 is set, then the drive halts (pauses) in all modes. The setpoints (speed for homing or jogging, motion task number, setpoints for digital mode) for the individual modes are retained.

Bit 9,10 These bits are reserved for the drive profile (DS402).

Bit 13, 14, 15 These bits are manufacturer-specific, and reserved at present.

6. 34 SDO 6041H: STATUS WORD (DS402)

The momentary state of the status machine can be read out with the aid of the status word.

Index	**6041_h**
Name	Status word
Object code	VAR
Data type	UNSIGNED16
Mode	all
Access	rw
PDO mapping	possible
Unit	—
Value range	0 ... 65535
EEProM	yes
Default value	0

Bit assignment in the status word

Bit	Name
0	Ready to switch on
1	Switched on
2	Operation enable
3	Fault
4	Voltage enabled
5	Quick stop
6	Switch on disabled
7	Warning

Bit	Name
8	Manufacturer-specific (reserved)
9	Remote (always 1)
10	Target reached
11	Internal limit active (in preparation)
12	Operation mode specific (reserved)
13	Operation mode specific (reserved)
14	Manufacturer-specific (reserved)
15	Manufacturer-specific (reserved)

States of the status machine

State	Bit 6 switch on disable	Bit 5 Quick stop	Bit 3 fault	Bit 2 operation enable	Bit 1 switched on	Bit 0 ready to switch on
Not ready to switch on	0	X	0	0	0	0
Switch on disabled	1	X	0	0	0	0
Ready to switch on	0	1	0	0	0	1
Switched on	0	1	0	0	1	1
Operation enabled	0	1	0	1	1	1
Fault	0	X	1	0	0	0
Fault reaction active	0	X	1	1	1	1
Quick stop active	0	0	0	1	1	1

Bits marked by X are irrelevant

Description of the remaining bits in the status word

Bit 4: voltage_enabled The DC-link voltage is present if this bit is set.

Bit 7: warning There are several possible reasons for Bit 7 being set and this warning being produced. The reason for this warning can be revealed by using the SDO 20subindex manufacturer warnings.

Bit 9: remote is always set to 1, i.e. the drive can always communicate and be influenced via the RS232 - interface.

Bit 10: target_reached This is set when the drive has reached the target position.

Bit 11: internal_limit_active is not supported at present

6. 35 SDO 606CH: VELOCITY ACTUAL VALUE (DS402)

The *object velocity actual value* represents the actual speed. The scaling of the value depends on the factor *velocity encoder resolution* (SDO 6094_h).

Index	**$606C_h$**
Name	velocity actual value
Object code	VAR
Data type	INTEGER32
Mode	pv
Access	ro
PDO mapping	possible
Unit	velocity units
Value range	(-2^{31}) ... $(2^{31}-1)$
Default value	—
EEProM	no

6. 36 SDO 6060H: MODES OF OPERATION (DS402)

This object is used to set the mode, which can be read out by SDO 6061h. Two types of operating mode can be distinguished:

Manufacturer-Specific Operating Modes

These modes of operation have been optimized to the functionality of the equipment.

Operating Modes As Per CANopen Drive Profile DS402

These operating modes are defined in the **CAN**open drive profile DS402.

After the mode has been changed, the corresponding set point must be set once more (for instance, the homing velocity in the mode homing_setpoint). If the position or jogging mode is stored, then the Homing mode is set after a *RESET* of the servo amplifier.

WARNING

Never change the mode while the motor is running! The drive could do unexpected movements.
When the amplifier is enabled, a mode change is only permissible at zero speed. Set the speed set point to 0 before changing over.

CAUTION

An operating mode only becomes valid when it can be read by SDO 6061$_h$.

Index	6060h
Name	Mode of operation
Object code	VAR
Data type	INTEGER8
Category	mandatory
Access	rw
PDO mapping	possible
Value range	80h ... 88h, 1, 3, 4, 6, 7
Default value	—

Supported modes (negative values are manufacturer specific modes):

Value (hex)	Mode
80	Digital velocity control mode
81	Analogue velocity control mode
82	Digital current control mode
83	Analog current control mode
84	Electronic gearing
85	Reserved
86	Reserved
87	Reserved
88	Motion task mode
1	Profile position mode
3	Profile velocity mode
4	Profile torque mode
6	Homing mode
7	Interpolated position mode

6. 37 SDO 6061H: MODES OF OPERATION DISPLAY (DS402)

This object can be used to read the mode that is set by SDO 6060_h. An operating mode only becomes valid when it can be read by SDO 6061_h (see also SDO 6060_h).

Index	**6061_h**
Name	Mode of operation display
Object code	VAR
Data type	INTEGER8
Category	mandatory
Access	ro
PDO mapping	possible
Value range	80_h ... 88_h, 1, 3, 4, 6, 7
Default value	—

6. 38 SDO 6063H: POSITION ACTUAL VALUE (DS402)

The object *position actual value* provides the momentary actual position in increments. The resolution is defined with SDO 608F as power-of-two number (see *PRBASE* command).

Index	**6063_h**
Name	position actual value
Object code	VAR
Data type	INTEGER32
Mode	pc, pp
Access	rw
PDO mapping	possible
Unit	increments (1 turn = 2^{PRBASE})
Value range	(-2^{31}) ... $(2^{31}-1)$
Default value	2^{20}
EEProM	no

6. 39 SDO 6064H: POSITION ACTUAL VALUE (DS402)

The object *position actual value* provides the actual position. The resolution can be altered by the gearing factors of the position controller (SDO 6092).

Index	**6064_h**
Name	Position actual value
Object code	VAR
Data type	INTEGER32
Mode	pc, pp
Access	rw
PDO mapping	possible
Unit	position units
Value range	(-2^{31}) ... $(2^{31}-1)$
Default value	—
EEProM	no

6. 40 SDO 6065H: FOLLOWING ERROR WINDOW

The *following error window* defines a range of tolerated position values symmetrically to the position demand value. A following error might occur when a drive is blocked, unreachable profile velocity occurs, or at wrong closed loop coefficients. If the value of the following error window is 0, the following control is switched off.

Index	**6065_h**
Name	Following error window
Object code	VAR
Data type	UNSIGNED32
Category	optional
Access	rw
PDO mapping	possible
Value range	UNSIGNED32
Default value	¼ of a motor revolution

6. 41 SDO 6067H: POSITION WINDOW (DS402)

The position window defines a symmetrical range of accepted positions relative to the target position. If the actual value of the position encoder is within the position window, the target position is considered reached. The status word bit *Target reached* goes to 1.

Index	**6067_h**
Name	Position window
Object code	VAR
Data type	UNSIGNED32
Category	optional
Access	rw
PDO mapping	possible
Value range	UNSIGNED32
Default value	4000 position units

6. 42 SDO 6068H: POSITION WINDOW TIME (DS402)

When the *actual position* is within the position window during the defined position window time which is given in multiples of milliseconds, the corresponding bit 10 *target reached* in the status word is set to one.

Index	**6068_h**
Name	Position window time
Object code	VAR
Data type	UNSIGNED16
Category	optional
Access	rw
PDO mapping	possible
Value range	UNSIGNED16
Default value	no

6. 43 SDO 6071H: TARGET TORQUE (DS402)

This parameter is the input value for the torque controller in profile torque mode and the value is given per thousand of rated torque.

Index	**6071_h**
Name	Target torque
Object code	VAR
Data type	INTEGER16
Category	conditional; mandatory, if tq supported
Access	rw
PDO mapping	possible
Value range	INTEGER16
Default value	0

6. 44 SDO 6073H: MAX CURRENT (DS402)

This value represents the maximum permissible torque creating current in the motor and is given per thousand of rated current.

Index	**6073_h**
Name	Max current
Object code	VAR
Data type	UNSIGNED16
Category	optional
Access	rw
PDO mapping	possible
Value range	UNSIGNED16
Default value	0

6. 45 SDO 6077H: TORQUE ACTUAL VALUE (DS402)

The torque actual value corresponds to the instantaneous torque in the drive motor. The value is given per thousand of rated torque.

Index	**6077_h**
Name	Torque actual value
Object code	VAR
Data type	INTEGER16
Category	optional
Access	ro
PDO mapping	possible
Value range	INTEGER16
Default value	0

6. 46 SDO 608Ah: Position Dimension Index (DS402)

Index $608A_h$
Name position dimension index
Object code VAR
Data type UNSIGNED8

Category mandatory
Access rw
PDO mapping not possible
Value range UNSIGNED8
Default value 0

The *position dimension index* defines the SI-units of the used position set points. Relationship between the SDO-values and the manufacturer-specific parameter **PUNIT** is:

Value of SDO608Ah	ASCII parameter PUNIT	SI unit
1	9...1	m
0	0	Manufacturer specific increments

The parameter **PUNIT** can be stored in the drive. The values of SDO 6089h and 608Ah are initialized by that parameter.

6. 47 SDO 608Bh: Velocity Notation Index (DS402)

Index 608Bh
Name velocity notation index
Object code VAR
Data type INTEGER8

Category mandatory
Access rw
PDO mapping not possible
Value range INTEGER8
Default value 0

The *velocity notation index* scales velocity set points, which units are defined with the „velocity dimension index“ as SI-units, in powers of ten.

Relationship between the SDO-values and the parameter VUNIT:

Value of SDO$608B_h$	ASCII parameter VUNIT	Scaling
0	0	1
0	1	1
0	5	1
0	6	1
FD_h	7	10^{-3}
FD_h	8	10^{-3}

6. 48 SDO 608Ch: Velocity Dimension Index (DS402)

Index	**608Ch**
Name	velocity dimension index
Object code	VAR
Data type	UNSIGNED8
Category	mandatory
Access	rw
PDO mapping	not possible
Value range	UNSIGNED8
Default value	0

The *velocity dimension index* defines the SI-unit of the used velocity set points.
Relationship between the SDO-values and the manufacturer-specific parameter VUNIT:

Value of SDO608C$_h$	ASCII parameter VUNIT	SI unit
A6$_h$	0	m/s
A4$_h$	1	turn/min
A6$_h$	5	m/s
A7$_h$	6	m/min
A6$_h$	7	m/s
A7$_h$	8	m/min

The parameter VUNIT can be stored in the drive. The values for SDOs 608Bh and 608Ch are initialized by this parameter. Only the described values for VUNIT are possible with the profile DS402.

6. 49 SDO 608Dh: Acceleration Notation Index (DS402)

Index	**608D$_h$**
Name	acceleration notation index
Object code	VAR
Data type	INTEGER8
Category	mandatory
Access	rw
PDO mapping	not possible
Value range	INTEGER8
Default value	0

The *acceleration notation index* scales acceleration set points, which units are defined with the„acceleration dimension index“ as SI-units, in powers of ten.

Relationship between the SDO-values and the parameter ACCUNIT:

Value of SDO608D$_h$	ASCII parameter ACCUNIT	Scaling
0	1,5	1
FA$_h$	3	10-6
FDh	4	10-3

6. 50 SDO 608Eh: Acceleration Dimension Index (DS402)

Index	**$608E_h$**
Name	acceleration dimension index
Object code	VAR
Data type	UNSIGNED8
Category	mandatory
Access	rw
PDO mapping	not possible
Value range	UNSIGNED8
Default value	AEh

The *acceleration dimension index* defines the SI-unit of the used acceleration set points.

Relationship between the SDO-values and the manufacturer-specific parameter ACCUNIT:

Value of SDO$608E_h$	ASCII parameter ACCUNIT	SI unit
AE_h	1	rad/s²
55_h	3, 4, 5	m/s

The parameter ACCUNIT can be stored in the drive. The values for SDOs 608Dh and 608Eh are initialized by this parameter. Only the described values for ACCUNIT are possible with the profile DS 402.

6. 51 SDO 608FH: POSITION ENCODER RESOLUTION (DS402)

The position encoder resolution defines the ratio of encoder increments per motor revolution. This object is used in the same way for SDO 6090 (velocity encoder resolution).

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

Index	**608Fh**
Name	Position encoder resolution
Object Code	ARRAY
Data Type	UNSIGNED 32
Category	optional

Subindex	**0**
Name	Number of entries
Category	mandatory
Access	ro
PDO Mapping	not possible
Value Range	2
Default Value	2

Subindex	**1**
Name	Encoder increments
Category	mandatory
Access	rw
PDO Mapping	possible
Value Range	UNSIGNED 32
Default Value	2^20

Subindex	**2**
Name	Motor revolutions
Category	mandatory
Access	rw
PDO Mapping	possible
Value Range	UNSIGNED 32
Default Value	1

6. 52 SDO 6089H: POSITION NOTATION INDEX (DS402)

Index	**6089_h**
Name	position notation index
Object code	VAR
Data type	INTEGER8
Category	mandatory
Access	rw
PDO mapping	not possible
Value range	INTEGER8
Default value	0

The *position notation index* scales position set points, which units are defined with the *position dimension index* in SI-units, in powers of ten.

Relationship between the values for SDO 6089 and the manufacturer specific parameter PUNIT:

Value of $SDO6089_h$	ASCII parameter PUNIT	Scaling
FF_h	1	10-1
FE_h	2	10-2
FD_h	3	10-3
FC_h	4	10-4
FB_h	5	10-5
FA_h	6	10-6
$F9_h$	7	10-7
$F8_h$	8	10-8
$F7_h$	9	10-9
0	0	1

6. 53 SDO 6091H: GEAR RATIO (DS402)

The gear ratio defines the ratio of motor shaft revolution per driving shaft revolutions.

$$\text{gear ratio} = \frac{\text{motor shaft revolutions}}{\text{driving shaft revolutions}}$$

Index	**6091h**
Name	Gear ratio
Object Code	ARRAY
Data Type	UNSIGNED 32
Category	optional

Subindex	**0**
Name	Number of entries
Category	mandatory
Access	ro
PDO Mapping	not possible
Value Range	2
Default Value	2

Subindex	**1**
Name	Motor revolutions
Category	mandatory
Access	rw
PDO Mapping	possible
Value Range	UNSIGNED 32
Default Value	1

Subindex	**2**
Name	Shaft revolutions
Category	mandatory
Access	rw
PDO Mapping	possible
Value Range	UNSIGNED 32
Default Value	1

6. 54 SDO 6092H: FEED CONSTANT (DS402)

The feed constant defines the ratio of feed in position units per driving shaft revolutions. This includes the gear, if present.

$$\text{feed constant} = \frac{\text{feed}}{\text{driving shaft revolutions}}$$

Index	**6092h**
Name	Feed constant
Object Code	ARRAY
Data Type	UNSIGNED 32
Category	optional

Subindex	**0**
Name	Number of entries
Category	mandatory
Access	ro
PDO Mapping	not possible
Value Range	2
Default Value	2

Subindex	**1**
Name	Feed
Category	mandatory
Access	rw
PDO Mapping	possible
Value Range	UNSIGNED 32
Default Value	1

Subindex	**2**
Name	Shaft revolutions
Category	mandatory
Access	rw
PDO Mapping	possible
Value Range	UNSIGNED 32
Default Value	1

6. 55 SDO 6093H: POSITION FACTOR (DS402)

The position factor converts the desired position (in position units) into the internal format (in increments). These values are calculated via the SDOs 608F and 6091.

$$\text{position factor} = \frac{\text{position encoder resolution} * \text{gear ratio}}{\text{feed constant}}$$

Index	**6093h**
Name	Position factor
Object Code	ARRAY
Data Type	UNSIGNED 32
Category	optional

Subindex	**0**
Name	Number of entries
Category	mandatory
Access	ro
PDO Mapping	not possible
Value Range	2
Default Value	2

Subindex	**1**
Name	Numerator (position encoder resolution * gear ratio)
Category	mandatory
Access	rw
PDO Mapping	possible
Value Range	UNSIGNED 32
Default Value	1

Subindex	**2**
Name	Feed constant
Category	mandatory
Access	rw
PDO Mapping	possible
Value Range	UNSIGNED 32
Default Value	1

6. 56 SDO 6097H: ACCELERATION FACTOR (DS402)

The acceleration factor converts the acceleration (in acceleration units / s) into the internal format (in increments / s). This factor is actually calculated from SDO 6093 and readable only.

$$\text{acceleration factor} = \frac{\text{velocity unit} * \text{velocity encoder factor}}{\text{acceleration unit} * \text{second}}$$

Index	**6097_h**
Name	Acceleration factor
Object Code	ARRAY
Data Type	UNSIGNED 32
Category	optional

Subindex	**0**
Name	Number of entries
Category	mandatory
Access	ro
PDO Mapping	not possible
Value Range	2
Default Value	2

Subindex	**1**
Name	Numerator (velocity unit * velocity encoder factor)
Category	mandatory
Access	ro
PDO Mapping	possible
Value Range	UNSIGNED 32
Default Value	1

Subindex	**2**
Name	Divisor (acceleration unit * second)
Category	mandatory
Access	ro
PDO Mapping	possible
Value Range	UNSIGNED 32
Default Value	1

6. 57 SDO 6502: SUPPORTED DRIVE MODES (DS402)

A drive can support more than one and several distinct modes of operation. This object gives an overview of the implemented operating modes in the device. This object is read only.

Index 6502h
Name Supported drive modes
Object code VAR
Data type UNSIGNED32
Category optional

Access ro
PDO mapping possible
Value range UNSIGNED32
Default value 0x6D (ip hm tq pv pp)

31 16	15 7	6	5	4	3	2	1	0
Manufacturer specific	Reserved	ip	hm	Reserved	tq	pv	vl	pp
MSB								LSB

7. EXAMPLES

Note: For all the following examples a single drive with address set to 3 will be represented in the communication examples (COB-ID 603). It will be presumed that the bus is functionally communicating with proper serial connections and settings.

7. 1 *POWER UP MESSAGE*

COB-ID	Data	Comments
703	00	Broadcast Message

7. 2 *EXAMPLE: ENABLING THE DRIVE FOR OPERATION (OPERATING THE STATUS/STATE MACHINE)*

The status machine must be run through a series of states to enable the drive. It is not possible to enable the drive without these steps. The Hardware Enable input must also be active before the drive will put power to the motor. After the S200 Position Node is powered on and allowed to initialise (D or E in the display or 703 broadcast) communication via SDOs can be initiated. The state of the status machine can be obtained through the query of Object 6041 Sub Index 0. The following data would be communicated on the CAN bus when reading the Control Word Status (6041):

	COB-ID	Control byte	Index		Sub-index	Data	Comment
			Low byte	High byte			
To Drive	603	40	41	60	00_h	40 00 00 00	Request Status
From Drive	583	4B	41	60	00_h	50 02 00 00	response telegram

Enabling the drive through CANOpen is done by a sequential writing of data to the Control word (Object 6040 Sub 0). If this is successful there will be a positive acknowledgement in the SDO reply (control byte 0 in the data field = 60_h). The sequence is to write 0x06, 0x07, 0x0F; in that order.

	COB-ID	Control byte	Index		Sub-index	Data	Comment
			Low byte	High byte			
To Drive	603	2B	40	60	00	06 00 00 00	Shut down
From Drive	583	60	40	60	00	00 00 00 00	response telegram
To Drive	603	2B	40	60	00	07 00 00 00	Switch on
From Drive	583	60	40	60	00	00 00 00 00	response telegram
To Drive	603	2B	40	60	00	0F 00 00 00	Enable for Operation
From Drive	583	60	40	60	00	00 00 00 00	

Control word = 0x000F Meaning: Bit 0, Bit 1, Bit 2, Bit 3 set ⇒ Enable Operation

The new status can then be queried again, and returns the following result:

COB-ID	Control byte	Index		Sub-index	Data	Comment
		Low byte	High byte			
603	40	41	60	00_h	—	query status
583	4B	41	60	00_h	37 02 00 00	response telegram

Status = 0x0237 Meaning: Bits 9 and 2 set :Operation Enabled, Remote

7. 3 EXAMPLE: JOG MODE VIA PDO

It is useful to disable unused PDOs. In Operation Mode "Digital Velocity" a digital speed setpoint is transmitted via RXPDO. Actual position and actual speed is read via a TXPDO triggered by SYNC.

COB-ID	Control byte	Index		Sub-index	Data	Comment
		Low byte	High byte			
603	2F	60	60	00_h	03 00 00 00	Mode of operation “Profile Velocity”
583	60	60	60	00_h	00 00 00 00	response telegram
603	2F	00	16	00_h	00 00 00 00	delete entries for the first RXPDO
583	60	00	16	00_h	00 00 00 00	response telegram
603	23	00	16	01_h	20 00 FF 60	mapping RXPDO1, Object 60FF, Sub-Index 0 speed setpoint, data length 32bit
583	60	00	16	01_h	00 00 00 00	response telegram
603	2F	00	16	00_h	01 00 00 00	number of mapped objects
583	60	00	16	00_h	00 00 00 00	response telegram
603	2F	00	1A	00_h	00 00 00 00	delete entries for the first TXPDO
583	60	00	1A	00_h	00 00 00 00	response telegram
603	23	00	1A	01_h	20 00 64 60	mapping TXPDO1/1, Object6064, Sub-Index 0 current position value in SI units, data length 32bit
583	60	00	1A	01_h	00 00 00 00	response telegram
603	23	00	1A	02_h	20 00 6C 60	mapping TXPDO1/2, Object606C, Sub-Index 0 current speed value, data length 32bit
583	60	00	1A	02_h	00 00 00 00	response telegram
603	2F	00	1A	00h	02 00 00 00	number of mapped objects
583	60	00	1A	00h	00 00 00 00	response telegram
603	2F	00	18	02_h	01 00 00 00	set TXPDO1 to synchronous, transmission with every SYNC
583	60	00	18	02_h	00 00 00 00	response telegram
603	23	01	18	01_h	83 02 00 80	disable TPDO2, set bit 31 (80h)
583	60	01	18	01_h	00 00 00 00	response telegram
603	23	02	18	01_h	83 03 00 80	disable TPDO3
583	60	02	18	01_h	00 00 00 00	response telegram
603	23	03	18	01_h	83 04 00 80	disabled TPDO4
583	60	03	18	01_h	00 00 00 00	response telegram
603	23	01	14	01_h	03 03 00 80	disabled RPDO2
583	60	01	14	01_h	00 00 00 00	response telegram
603	23	02	14	01_h	03 04 00 80	disabled RPDO3
583	60	02	14	01_h	00 00 00 00	response telegram
603	23	03	14	01_h	03 05 00 80	disabled RPDO4
583	60	03	14	01_h	00 00 00 00	response telegram
000					01 03	enable NMT
603	2B	40	60	00_h	06 00 00 00	shutdown
583	60	40	60	00h	00 00 00 00	response telegram
603	2B	40	60	00_h	07 00 00 00	switch on
583	60	40	60	00_h	00 00 00 00	response telegram
603	2B	40	60	00_h	0F 00 00 00	enable operation
583	60	40	60	00_h	00 00 00 00	response telegram
203					00 00 04 00	setpoint V= 240 rpm Calculated as follows: 4000h=240d RPM/ 60d Sec/Min*65536d CPR
080						send SYNC
183	FE	45	01	00_h	FE 45 01 00 00 88 41 00	Response: Position and Speed Pos.= 00 01 45 FE = 83454 [Si units]; Act Speed = (00 41 88 00h) / 17894.4dec = 240 RPM 17894.4 is the const. factor.
603	2B	40	60	00_h	0F 01 00 00	intermediate stop
583	60	40	60	00_h	00 00 00 00	response telegram

Note: Velocity units default to increments per second based on 65536 counts per revolution.

7. 4 EXAMPLE: START MOTION TASK FROM THE INTERNAL MEMORY OF S200 POSITION NODE VIA SDO

This example needs a defined motion task (can be done with the setup software) saved in the S200 Position Node Drive and homing must be done before starting absolute motion tasks.

COB-ID	Control byte	Index		Sub-index	Data	Comment
		Low byte	High byte			
603	2F	60	60	00_h	01 00 00 00	Mode of operation= position
583	60	60	60	00_h	00 00 00 00	
603	2B	40	60	00_h	06 00 00 00	Shutdown
583	60	40	60	00_h	00 00 00 00	
603	2B	40	60	00_h	07 00 00 00	Switch On
583	60	40	60	00_h	00 00 00 00	
603	2B	40	60	00h	0F 00 00 00	Enable Operation
583	60	40	60	00h	00 00 00 00	
603	2B	80	20	00h	03 00 00 00	Select motion task 3
583	60	80	20	00h	00 00 00 00	
603	2B	40	60	00h	3F 00 00 00	Start with new SETPOINT and CHANGE_SET_IMMEDIATELY
583	60	40	60	00h	00 00 00 00	
603	40	81	20	00h	00 00 00 00	Read active motion task response: Motion task 3 in operation
583	4B	81	20	00h	03 00 00 00	
603	2B	40	60	00h	0F 00 00 00	Reset Start Bit
583	60	40	60	00h	00 00 00 00	
603	2B	40	60	00h	3F 00 00 00	Do move again
583	60	40	60	00h	00 00 00 00	

Sales and Service

Danaher Motion is committed to quality customer service. Our products are available world-wide through an extensive authorized distributor network. To serve in the most effective way, please contact your local sales representative for assistance. If you are unaware of your local sales representative, please contact us.

Europe
Danaher Motion Customer Service Europe
Email: support@danahermotion.net
Phone: +49(0)203 9979 9
Fax: +49(0)203 9979 155
Web: www.DanaherMotion.net

North America
Danaher Motion Customer Service North America
Email: customer.support@danahermotion.com
Phone: 1-540-633-3400
Fax: 1-540-639-4162
Web: www.DanaherMotion.com

DANAHER MOTION

Helping you build a better machine faster.