AKD™

CAN-BUS Communication

Edition: F, March 2012
Valid for Hardware Revision D
Part Number 903-200004-00
Original Documentation

CANopen®

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

Record of Document Revisions:

Revision	Remarks
- ,11/2009	Beta launch version
-, 12/2009	Objects 2018&60FE added, Object dictionary, formatting updated
A, 04/2010	Termination connector "optional", several new objects, Object dictionary split
B, 07/2010	Part number added, several new objects, object dictionary expanded
C, 01/2011	HW Rev. C, new objects, object dictionary expanded
D, 04/2011	Object dictionary updated, baudrate setup
E, 10/2011	Cover layout & error table & object dictionary updated, objects 3474 & 3475 & 3496 & 6091 added
F, 03/2012	Touch Probe objects 60B8 to 60BD & 60D0 added, object 2071 & 2077 added, PVT interpolation added, 60C0 & 60C1 & 60C4 & 6041 bit 9 updated, object dictionary updated, error codes updated, object 1011h added

Hardware Revision (HR)

AKD HR	AKD PDMM HR	Firmware	WorkBench	KAS IDE	Remarks
A	-	01.03.xx.yyy	1.3.0.zzzzz	-	AKD Start revision
C	-	01.05.xx.yyy	1.5.0.zzzzz	-	STO certified, PROFINET RT released
D	DA	01.06.xx.yyy	1.6.0.zzzzz	2.5.0.zzzzz	Control board revision 9, AKD PDMM Start revision

Current patents:

US Patent 5,646,496 (used in control card R/D and 1 Vp-p feedback interface)
US Patent 5,162,798 (used in control card R/D)
US Patent 6,118,241 (used in control card simple dynamic braking)

Technical changes which improve the performance of the device may be made without prior notice!

Table of Contents

This page intentionally left blank.

1 General

1.1 About this Manual

This manual, *AKD CAN-Bus Communication*, This manual describes the installation, setup, range of functions, and software protocol for the CANopen AKD product series. All AKD CANopen drives have built-in CANopen functionality; therefore an additional option card is not required.

A digital version of this manual (pdf format) is available on the CD-ROM included with your drive. Manual updates can be downloaded from the Kollmorgen™ website.

Related documents for the AKD series include:

- *AKD Quick Start* (also provided in hard copy). This guide provides instructions for initial drive setup and connection to a network.
- *AKD Installation Manual* (also provided in hard copy for EU customers). This manual provides instructions for installation and drive setup.
- *AKD Users Manual*. This manual describes how to use your drive in common applications. It also provides tips for maximizing your system performance with the AKD.
- *AKD Parameter and Command Reference Guide*. This guide provides documentation for the parameters and commands used to program the AKD.
- *Accessories Manual*. This manual provides documentation for accessories like cables and regen resistors used with AKD. Regional versions of this manual exist.

Additional documentation:

- CAN Application (CAL) for Industrial Applications (publisher CiA e.V.)
- Draft Standards 301 (from Version 4.0), 402 (publisher CiA e.V.)
- CAN Specification Version 2.0 (publisher CiA e.V.)
- ISO 11898 ... Controller Area Network (CAN) for high-speed communication

1.2 Target Group

This manual addresses personnel with the following qualifications:

- Installation: only by electrically qualified personnel.
- Setup : only by qualified personnel with extensive knowledge of electrical engineering and drive technology
- Programming: Software developers, project-planners

The qualified personnel must know and observe the following standards:

- ISO 12100, IEC 60364 and IEC 60664
- National accident prevention regulations

⚠ WARNING **During operation there are deadly hazards, with the possibility of death, severe injury or material damage. The operator must ensure that the safety instructions in this manual are followed. The operator must ensure that all personnel responsible for working with the servo drive have read and understand the manual.**

1.3 Symbols used

Warning Symbols

Symbol	Indication
⚠ DANGER	Indicates a hazardous situation which, if not avoided, will result in death or serious injury.
⚠ WARNING	Indicates a hazardous situation which, if not avoided, could result in death or serious injury.
⚠ CAUTION	Indicates a hazardous situation which, if not avoided, could result in minor or moderate injury.
NOTICE	Indicates situations which, if not avoided, could result in property damage.
NOTE	This is not a safety symbol. This symbol indicates important notes.

Drawing symbols

Symbol	Description	Symbol	Description
	Signal ground		Diode
	Chassis ground		Relays
	Protective earth		Relays switch off delayed
	Resistor		Normal open contact
	Fuse		Normal closed contact

1.4 Abbreviations used

Abbreviation	Meaning
BTB/RTO	Ready to operate (standby)
COB	Communication Object
COB-ID	Communication Object Identifier
EEPROM	Electrically erasable/programmable memory
EMC	Electromagnetic compatibility
EMCY	Emergency Objects
ISO	International Standardization Organization
km	1000 m
LED	Light-emitting diode
LSB	Low significant Byte (or Bit)
MSB	Main significant Byte (or Bit)
MB	Megabyte
NMT	Network Management Objects
NSTOP	Limit switch for negative (left) rotation
PC	Personal Computer
PDO	Process Data Object
PSTOP	Limit switch for positive (right) rotation
RAM	Volatile memory
ROD	Incremental position encoder
RXPDO	Receive PDO
SDO	Service Data Object
SYNC	Synchronization Objects
TXPDO	Transmit PDO

2 Safety

2.1 Safety Instructions

⚠ DANGER	**During operation there are deadly hazards, with the possibility of death, severe injury or material damage. Do not open or touch the equipment during operation. Keep all covers and cabinet doors closed during operation. Touching the equipment is allowed during installation and commissioning for properly qualified persons only.** • **During operation, drives may have uncovered live components, depending on their level of enclosure protection.** • **Control and power connections may be live, even though the motor is not rotating.** • **Drives may have hot surfaces during operation. Heat sink can reach temperatures above 80°C.**
⚠ WARNING	**Electronic equipment is basically not failure-proof. The user is responsible for ensuring that, in the event of a failure of the drive, the drive is set to a state that is safe for both machinery and personnel, for instance with the aid of a mechanical brake.** **Drives with drives and CANopen expansion cards are remote-controlled machines. They can start to move at any time without previous warning. Take appropriate measures to ensure that the operating and service personnel is aware of this danger.** **Implement appropriate protective measures to ensure that any unintended start-up of the machines cannot result in dangerous situations for personnel or machinery. Software limit-switches are not a substitute for the hardware limit-switches in the machine.**
NOTICE	**Install the drive as described in the *Installation Manual*. The wiring for the analog setpoint input and the positioning interface, as shown in the wiring diagram in the *Installation Manual*, is not required. Never break any of the electrical connections to the drive while it is live. This action can result in destruction of the electronics**

2.2 Use As Directed

Drives are components that are built into electrical plants or machines and can only be operated as integral components of these plants or machines. The manufacturer of the machine used with a drive must generate a risk assessment for the machine and take appropriate measures to ensure that unforeseen movements cannot cause personnel injury or property damage.

Please observe the chapters "Use as directed" and "Prohibited use" in the *AKD Installation Manual*.

The CANopen interface serves only for the connection of the *AKD* to a master via the CAN bus.

2.3 Prohibited Use

Other use than that described in chapter "Use as directed" is not intended and can lead to personnel injuries and equipment damage. The drive may not be used with a machine that does not comply with appropriate national directives or standards. The use of the drive in the following environments is also prohibited:

- potentially explosive areas
- environments with corrosive and/or electrically conductive acids, alkaline solutions, oils, vapors, dusts
- ships or offshore applications

3 Installation and Setup

3.1 Safety Instructions

⚠ DANGER	Never undo any electrical connections to the drive while it is live. There is a danger of electrical arcing with damage to contacts and serious personal injury. Wait at least seven minutes after disconnecting the drive from the main supply power before touching potentially live sections of the equipment (e.g. contacts) or undoing any connections. Capacitors can still have dangerous voltages present up to 7 minutes after switching off the supply power. To be sure, measure the voltage in the DC Bus link and wait until it has fallen below 40 V. Control and power connections can still be live, even if the motor is not rotating.
⚠ WARNING	Electronic equipment is basically not failure-proof. The user is responsible for ensuring that, in the event of a failure of the drive, the drive is set to a state that is safe for both machinery and personnel, for instance with the aid of a mechanical brake. Drives with drives and CANopen are remote-controlled machines. They can start to move at any time without previous warning. Take appropriate measures to ensure that the operating and service personnel is aware of this danger. Implement appropriate protective measures to ensure that any unintended start-up of the machines cannot result in dangerous situations for personnel or machinery. Software limit-switches are not a substitute for the hardware limit-switches in the machine.
NOTICE	Install the drive as described in the *Installation Manual*. The wiring for the analog setpoint input and the positioning interface, as shown in the wiring diagram in the *Installation Manual*, is not required. Never break any of the electrical connections to the drive while it is live. This action can result in destruction of the electronics.
NOTICE	The drive's status must be monitored by the PLC to acknowledge critical situations. Wire the FAULT contact in series into the emergency stop circuit of the installation. The emergency stop circuit must operate the supply contactor.

NOTE	It is permissible to use the setup software to alter the settings of the drive. Any other alterations will invalidate the warranty.
NOTE	Because of the internal representation of the position-control parameters, the position controller can only be operated if the final limit speed of the drive does not exceed: **rotary** at sinusoidal² commutation: 7500 rpm at trapezoidal commutation: 12000 rpm. **linear** at sinusoidal² commutation: 4 m/s at trapezoidal commutation: 6.25 m/s
NOTE	All the data on resolution, step size, positioning accuracy etc. refer to calculatory values. Non-linearities in the mechanism (backlash, flexing, etc.) are not taken into account. If the final limit speed of the motor must be altered, then all the parameters that were previously entered for position control and motion blocks must be adapted.

3.2 CAN-Bus Interface (X12/X13)

Two 6-pin RJ-12 connectors X12/X13 are used for CAN-Bus connection.

Conn.	Pin	Signal	Conn.	Pin	Signal
X12	1	Internal Termination Resistor	X13	1	Internal Termination Resistor
X12	2	CAN Shield	X13	2	CAN Shield
X12	3	CANH in	X13	3	CANH out
X12	4	CANL in	X13	4	CANL out
X12	5	GND	X13	5	GND
X12	6	Internal Termination Resistor	X13	6	Internal Termination Resistor

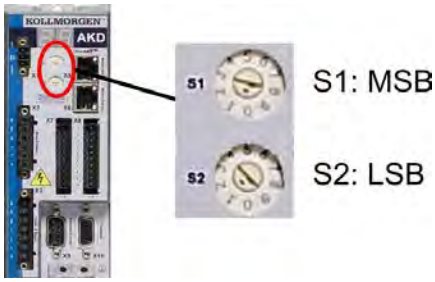
3.2.1 CAN-Bus activation with AKD-CC models

AKD-CC drive models are Drives, which support EtherCAT and CAN fieldbus types within one common software. These CC drive models allow selecting a fieldbus support by setting the DRV.TYPE parameter to a certain value. CC drive models are delivered with EtherCAT set active.
To activate CANopen, the DRV.TYPE parameter must be changed

1. by software: connect the PC to the AKD and change the parameter DRV.TYPE in the WorkBench terminal screen (see DRV.TYPE parameter documentation) or
2. by hardware: with the rotary switches S1 & S2 at the front and the button B1 on the top side of the Drive.

The following steps are needed for changing the fieldbus type from EtherCAT to CAN with the rotary switches.

1. Set the rotary switches on the front side of the AKD to the value of 89.



Set S1 to 8 and S2 to 9

2. Press the button B1 for about 3 seconds (starts DRV.NVSAVE).

The seven segment display shows **Cn** during the process of changing DRV.TYPE to CAN.
Do not switch off the 24[V] power supply while the seven segment shows Cn!

3. Wait until the seven segment display goes back to the original state, no the drive is prepared for CAN.
4. Power cycle the drive by switching the 24 V power supply **off** and then **on** again.

NOTE	**The seven segment display shows Er (Error) in case that the DRV.TYPE instruction failed. In this case please power cycle the drive and contact the Kollmorgen™ customer support for further help.**

3.2.2 Baudrate for CAN-Bus

The user can decide to use a fixed baud rate or an auto baud detection algorithm for the startup behaviour of the drive. The transmission rate can be set via the parameter **FBUS.PARAM01**. The parameter FBUS.PARAM01 can either be set via WorkBench or via a special mechanism with the rotary switches in the AKD front.

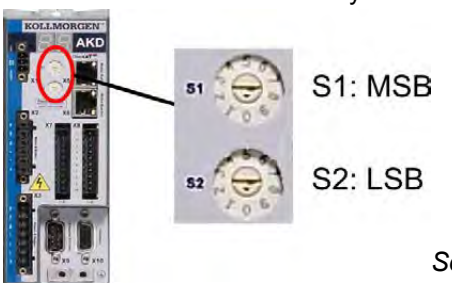
Baudrate [kBit/s]	FBUS.PARAM01	Upper rotary switch S1	Lower rotary switch S2
auto	0	9	0
125	125	9	1
250	250	9	2
500	500	9	3
1000	1000	9	4

In case of a fix baud rate, the drive sends the boot up message with the baud rate saved in the drive's non volatile memory after a power cycle. In case of auto baud detection, the drive listens for a valid CAN frame on the bus. When a valid frame is received, the drive sends the boot up message with the measured bit time. Afterwards the baud rate can either be stored to non volatile memory via object 1010 sub 1, or the auto baud mechanism is used always.

NOTE **For reliable auto baud detection, it is recommended to use suitable cabling of the CAN-Bus (two terminators, GND connection etc.). Spikes or other noise effects on the CAN-Bus can disturb the measurement. The drive needs to be disabled, if auto baud is in use.**

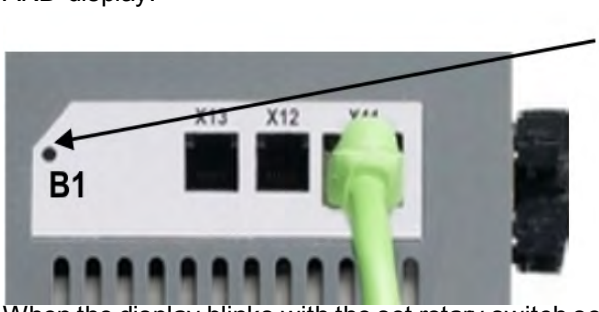
For setting the baudrate with rotary switches, follow the procedure below (drive state disabled):

1. Disable the drive. Set the rotary switches to one of the addresses 90 to 94 (see above table).



Set S1 to 9 and S2 to either 0 or 4

2. Push the button B1 on the AKDfor at least 3 seconds until the rotary switch setting is displayed on the AKD-display.



3. When the display blinks with the set rotary switch setting stop pushing B1 and wait until the blinking stops. During that time the parameter FBUS.PARAM01 is set to the new value and all parameters are stored to the non volatile memory. The new setting will be taken with the next power-up of the drive.

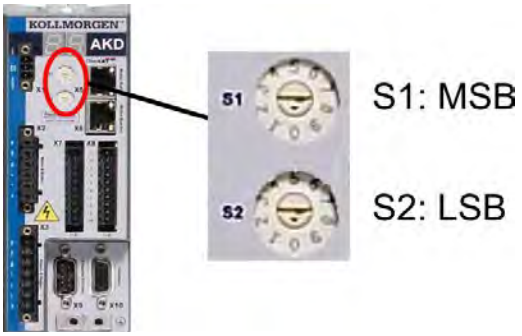
If an error occurred, the following error messages will flash 5 times:

- E1 - Drive is enabled
- E2 - Non-volatile storage of the new setting failed
- E3 - Invalid rotary switch selection

3.2.3 Node Address for CAN-Bus

NOTE **After changing the node address, you must turn off the 24 V auxiliary supply for the drive and then turn it on again.**

During setup, use the rotary switches on the AKD front panel to preset the station address for communication.



The rotary switches on the front of the AKD (S1&S2) correspond to the CAN node address.

The S1&S2 switches also correspond to the IP address setting of the drive. Both CAN and IP network address schemes have to be configured to account for this dependence if both TCP/IP and CAN networks are running at the same time in an application. Example:

S1 (MSB)	S2 (LSB)	CAN address	IP address
4	5	45	192.168.0.45

The IP address setting can be decoupled from the Rotary switches using settings in the drive. Use Settings -> Fieldbus-> TCP/IP to adjust these settings.

3.2.4 CAN-Bus Termination

The last bus device on both ends of the CAN-Bus system must have termination resistors. The AKD has built-in 132 ohms resistors that can be activated by connecting pins 1 and 6. An optional termination plug is available for AKD (*P-AKD-CAN-TERM*). The optional termination plug is an RJ-12 connector with an enclosed wire jumper between pins 1&6. A plug should be inserted into the X13 connector of the last drive in the CAN network.

NOTE **Remove the termination connector if the AKD is not the last CAN-Bus device and use X13 for connecting the next CAN node.**

3.2.5 CAN-Bus Cable

To meet ISO 11898, a bus cable with a characteristic impedance of 120 ohms should be used. The maximum usable cable length for reliable communication decreases with increasing transmission speed. As a guide, you can use the following values which Kollmorgen™ has measured; however, these values are not assured limits:

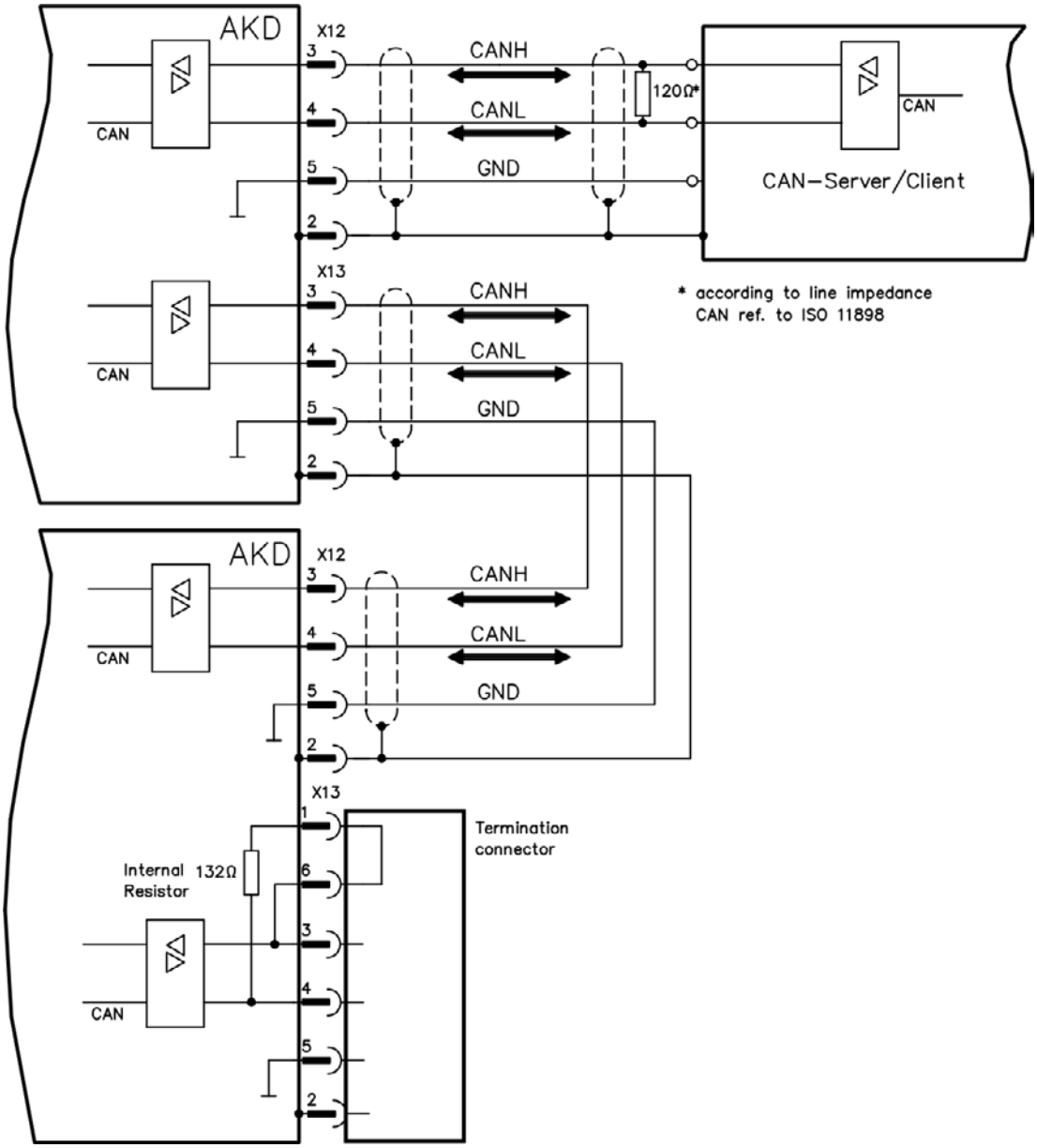
- Characteristic impedance: 100–120 ohms
- Cable capacitance max.: 60 nF/km
- Lead loop resistance: 159.8 ohms/km

Cable length, depending on the transmission rate:

Transmission Rate (kBaud)	Maximum Cable Length (m)
1,000	10
500	70
250	115

Lower cable capacitance (max. 30 nF/km) and lower lead resistance (loop resistance, 115 ohms/1000m) make it possible to achieve greater distances.
(Characteristic impedance 150 ± 5 ohms requires terminating resistor 150 ± 5 ohms).

3.2.6 CAN-Bus Wiring



3.3 Guide to Setup

⚠ WARNING	**Only professional personnel with extensive knowledge of control and drive technology are allowed to setup the drive.**
⚠ CAUTION	**Make sure that any unintended movement of the drive cannot endanger machinery or personnel.**

1. Check assembly/installation. Check that all the safety instructions in the product manual for the drive and this manual have been observed and implemented. Check the setting for the station address and baud rate.
2. Connect PC,start WorkBench. Use the setup software WorkBench to set the parameters for the drive.
3. Setup basic functions. Start up the basic functions of the drive and optimize the current, speed and position controllers. This section of the setup is described in the in the online help of the setup software.
4. Save parameters. When the parameters have been optimized, save them in the drive.
5. Start up communication. The altered parameters will only become effective after a reboot (switch off 24V and switch on again). Adjust the transmission rate of the AKD to match the master.
6. Test communication. Check for the bootup-message, when you switch on the drive. Do an SDO read access on index 0x1000 subindex 0 (DeviceType).
7. Setup position controller. Setup the position controller, as described in the WorkBenchonline help.

4 CANopen Basics

4.1 Basic Features implemented by CANopen

It is assumed that the basic operating functions of the communication profile are known and available as reference documentation. When working with the position controller that is integrated in AKD, the following functions are available:

4.1.1 Setup and general functions:

- Homing, set reference point
- Provision of a digital setpoint for speed and torque control
- Support of the following modes of the CANopen Profile DS402:
  - Profile position mode
  - Homing mode
  - Profile torque mode
  - Interpolated position mode
  - Profile velocity mode
  - Cyclic synchronous position mode

4.1.2 Positioning functions:

- Execution of a motion task from the motion block memory of the drive
- Execution of a direct motion task
- Absolute trajectory, ip-Mode or csp-Mode

4.1.3 Data transfer functions:

- Transmit a motion task to the drive's motion block memory. A motion task consists of these elements:
  - Position setpoint (absolute task) or path setpoint (relative task)
  - Speed setpoint
  - 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 drive
- Read actual values
- Read the error register (Emergency error codes)
- Read the status register
- Read/write control parameters

4.2 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 drive: 125 (default), 250, 500 and 1000 kbit/s
- The baudrate is set with the AKD - parameter FBUS.PARAM01. It gets effective by saving this parameter to NVRAM and re-starting the drive.

4.3 Response to 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 AKD, 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 702. If the output stage is enabled at the moment when this fault occurs, the output stage is disabled.

4.4 Important Configuration Parameters

FBUS.PARAM01 siehe "Transmission Rate and Procedure"

FBUS.PARAM02 0 - no PLL used for synchronization
1 - PLL used for synchronized modes, IP (7), CSP (8), generates a warning n125, when PLL is unlocked

FBUS.PARAM04 0 - arrival of SYNC-messages in cyclic-synchronized application is not supervised
1 - arrival of SYNC-messages in cyclic-synchronized application is supervised (after 3 missing SYNC-telegrams the fault F125 is generated)

FBUS.PARAM05 description for bits 0 to 3 as in AKD - command reference

Bit 0
1: Faults can only be reset using DS402 control word bit 7.
0 = 0: The reset can also be done via telnet or digital input and the DS402 state machine reflects this condition.

Bit 1
1: The state of the hardware enable does not change the state machine state Operation Enable.
0: If the state Operation Enable or Switched on is active it falls back to the state switched On Disabled, if the Hardware enable goes to 0.

Bit 2
1: Workbench/Telnet can not software enable the drive, when CANopen/EtherCAT are Operational.
0: Workbench/Telnet can software enable the drive.

Bit 3
1: DS402-state machine is not influenced, if the software-enable is taken away via Telnet.
0: DS402-state machine is influenced, if the software-enable is taken away via Telnet.

Bit 4
1: Scaling is done via special DS402 - objects (independent on units)
0: Scaling for position, velocity and acceleration objects is done via UNIT parameters

Bit 5 used in EtherCAT, reserved for CAN

Bit 6
1: Bit 0 of parameter MT.CNTL (object 35D9 sub 0) can be accesse
0: Bit 0 of parameter MT.CNTL (object 35D9 sub 0) is exclusively used for DS402 controlwordd

5 CANopen Communication Profile

5.1 General Description of CAN

This chapter describes the basic services and communication objects of the CANopen communication profile DS 301, which are used in the AKD.

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

The transmission method that is used here 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 that are provided by the Application Layer make it possible to implement an application that is spread across the network. These service elements are described in the CAN Application Layer (CAL) for Industrial Applications.
The communication profile CANopen 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 to 8byte (Data-COB)
	0byte (Remote-COB)
CRC	Cyclic Redundancy Check
ACK	Acknowledge slot
EOM	End of message

5.2 Construction of the 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 (drive's CAN-bus address, range 1 to 127; is set up in WorkBench or the drive,)

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

NOTE **If an invalid station number (=0) is set, then the module will be set internally to 1.**

The following tables show the default values for the COB Identifier after switching on the drive. 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	0	—
SYNC	0001	128	80	(1005)
TIME	0010	256	100	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	81..FF	—	high
TPDO 1	0011	385..511	181..1FF	1800	
RPDO 1	0100	513..639	201..27F	1400	
TPDO 2	0101	641..767	281..2FF	1801	
RPDO 2	0110	769..895	301..37F	1401	
TPDO 3	0110	897..1023	381..3FF	1802	
RPDO 3	1000	1025..1151	401..47F	1402	
TPDO 4	1001	1153..1279	481..4FF	1803	
RPDO 4	1010	1281..1407	501..57F	1403	
SDO (tx*)	1011	1409..1535	581..5FF		
SDO (rx*)	1100	1537..1663	601..67F		
Nodeguard	1110	1793..1919	701..77F	(100E)	low

*tx = direction of transmission: AKD => Master
rx = direction of transmission: Master => AKD

5.3 Definition of the Used Data Types

This chapter 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 (a.k.a. Intel format) is used for numerical data types (see also: DS301 Application Layer "General Description of Data Types and Encoding Rules").

5.3.1 Basic data types

5.3.1.1 Unsigned Integer

Data in the basic data type UNSIGNEDn define exclusively positive integers.
The value range is from 0 to 2^n-1. The bit sequence b = b_0 to b_{n-1} defines the value
UNSIGNEDn(b) = b_{n-1} 2^{n-1} + to + b_1 2^1 + b_0 2^0

Example: the value 266 = 10Ah is transmitted in the data type UNSIGNED16, in the form of two octets (1st octet = 0Ah, 2nd octet = 01h).

Transmission syntax for the data type UNSIGNEDn

Octet number	1.	2.	3.	4.
UNSIGNED8	b_7 to b_0			
UNSIGNED16	b_7 to b_0	b_{15} to b_8		
UNSIGNED24	b_7 to b_0	b_{15} to b_8	b_{23} to b_{16}	
UNSIGNED32	b_7 to b_0	b_{15} to b_8	b_{23} to b_{16}	b_{31} to b_{24}
UNSIGNED40	b_7 to b_0	b_{15} to b_8	b_{23} to b_{16}	b_{31} to b_{24}
UNSIGNED48	b_7 to b_0	b_{15} to b_8	b_{23} to b_{16}	b_{31} to b_{24}
UNSIGNED56	b_7 to b_0	b_{15} to b_8	b_{23} to b_{16}	b_{31} to b_{24}
UNSIGNED64	b_7 to b_0	b_{15} to b_8	b_{23} to b_{16}	b_{31} to b_{24}

Octet number	5.	6.	7.	8.
UNSIGNED8				
UNSIGNED16				
UNSIGNED24				
UNSIGNED32				
UNSIGNED40	b_{39} to b_{32}			
UNSIGNED48	b_{39} to b_{32}	b_{47} to b_{40}		
UNSIGNED56	b_{39} to b_{32}	b_{47} to b_{40}	b_{55} to b_{48}	
UNSIGNED64	b_{39} to b_{32}	b_{47} to b_{40}	b_{55} to b_{48}	b_{63} to b_{56}

5.3.1.2 Signed Integer

Data in the basic data type INTEGERn define both positive and negative integers.
The value range is from -2^{n-1}-1 to 2^{n-1}-1. The bit sequence b = b_0 to b_{n-1} defines the value
INTEGERn(b) = b_{n-2} 2^{n-2} + to + 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 = FEF6h is transmitted in the data type INTEGER16, in the form of two octets (1st octet = F6h, 2nd octet = FEh).

Transmission syntax for the data type INTEGERn

Octet number	1.	2.	3.	4.
INTEGER8	b_7 to b_0			
INTEGER16	b_7 to b_0	b_{15} to b_8		
INTEGER24	b_7 to b_0	b_{15} to b_8	b_{23} to b_{16}	
INTEGER32	b_7 to b_0	b_{15} to b_8	b_{23} to b_{16}	b_{31} to b_{24}
INTEGER40	b_7 to b_0	b_{15} to b_8	b_{23} to b_{16}	b_{31} to b_{24}
INTEGER48	b_7 to b_0	b_{15} to b_8	b_{23} to b_{16}	b_{31} to b_{24}
INTEGER56	b_7 to b_0	b_{15} to b_8	b_{23} to b_{16}	b_{31} to b_{24}
INTEGER64	b_7 to b_0	b_{15} to b_8	b_{23} to b_{16}	b_{31} to b_{24}

Octet number	5.	6.	7.	8.
INTEGER8				
INTEGER16				
INTEGER24				
INTEGER32				
INTEGER40	b_{39} to b_{32}			
INTEGER48	b_{39} to b_{32}	b_{47} to b_{40}		
INTEGER56	b_{39} to b_{32}	b_{47} to b_{40}	b_{55} to b_{48}	
INTEGER64	b_{39} to b_{32}	b_{47} to b_{40}	b_{55} to b_{48}	b_{63} to b_{56}

5.3.2 Mixed data types

Mixed data types combine basic data types (INTEGERn, UNSIGNEDn, 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.

5.3.3 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.

5.3.3.1 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

5.3.3.2 Visible String

The data type VISIBLE_STRING can be defined with the data type UNSIGNED8 or the data type ARRAY. Permissible values are 00h and the range from 20h to 7Eh. 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

5.4 Communication Objects

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

- Unconfirmed services PDO
- Confirmed services SDO

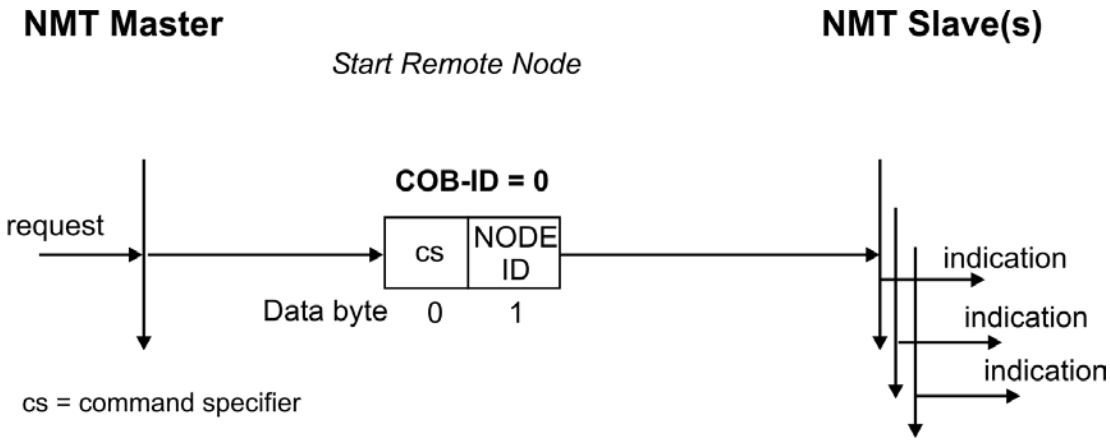
All services require faultless operation of the Data Link and Physical Layer.

AKD 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/Heartbeat

5.4.1 Network Management Objects (NMT)

The followind diagram describes the NMT telegram:



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. I.e. the PDOs of the drive are enabled for operation. From this moment, transmit-PDOs will be transmitted under event-control, and cyclical process data operation can commence.

cs = 2, stop remote node:
Stops the CAN node, I.e. the drive no longer responds to any received PDOs or transmits any PDOs.

5.4.2 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 appendix from page . You can use the SYNC object to start motion task of several axes simultaneously for example.

5.4.3 Time-Stamp Object (TIME)

This communication object is not supported by the AKD.

5.4.4 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 the Chapter *""CANopen Emergency Messages and Error Codes" (=> p. 42)"*. The last 10 Emergency error codes can be read via object 1003.

5.4.4.1 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:

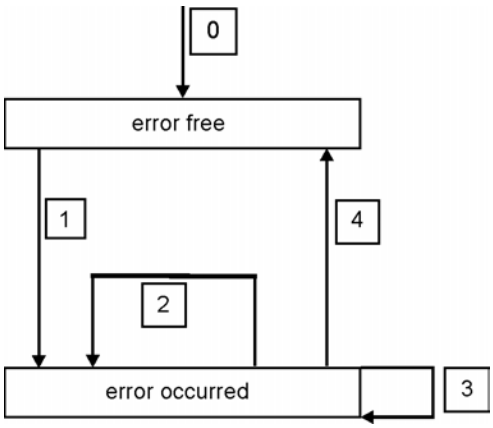
Transition 0: After initialization, the error-free status is taken up if no errors are detected.
No error signal is generated in this condition.

Transition1: The AKD 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).

Transition2: One error has been reset, but not all. The EMCY telegram contains error code 0000 and the error register indicates the remaining errors that are present. The manufacture-specific area is set to zero.

Transition3: A new error has occurred. The AKD remains in the error status and transmits an EMCY. Object with the corresponding error code. The new error code is entered into bytes 0 and 1.

Transition4: All errors have been reset. The EMCY telegram contains the error code 0000,
The error register does not indicate any other errors. The manufacture-specific area is set to zero.



5.4.4.2 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	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 Emergency Error Code (0000) and Byte 2 indicates if a possible further error is present. If the error register contains 00, 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 in the section Emergency Messages. The error register is defined through object 1001. Error register´.

5.4.5 Service Data Objects (SDO)

SDOs are used to implement access to the Object Dictionary. The SDOs are required for parametrerization 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 AKD:

- 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 appendix from page .

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

5.4.5.1 Composition of the Service Data Object

An SDO consists of the following components:

Byte	1	2	3	4	5	6	7	8
Content	R/W	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. A description of the complete Object Dictionary for AKD=> p. 117. Data exchange with the AKD 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 must be transmitted in the control byte.

The drive 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 successfull, 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 => p. 42

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 + Subindex, d: 4 bytes data field
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

Client	<= <= <= <= <= <=<= <= <=<= <= <=<= <= <=<= <= <=<= <= <=															**Server**
	Byte 1								**2**	**3**	**4**	**5**	**6**	**7**	**8**	
confirm	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**								response
<=	scs=3			X					min			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.
For example: 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 to 8):

These components are used for the exchange of user data. In read-request telegrams to the AKD they are set to 0. They have no content in a write confirmation from the AKD if the transfer was successful, but if the write operation was faulty they contain an error => p. 42.

5.4.5.2 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).

5.4.5.3 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).

5.4.5.4 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).

5.4.5.5 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).

5.4.5.6 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
0504 0000h	SDO timeout
0504 0001h	Command specifier invalid
0504 0002h	SDO segmented: invalid blocksize
0504 0004h	SDO segmented: invalid block CRC
0504 0005h	SDO segmented: out of memory
0601 0001h	Attempted read access to a write-only object
0601 0002h	Attempted write access to a read-only object
0602 0000h	Object does not exist in Object Dictionary
0604 0041h	Object cannot be mapped to a PDO
0604 0042h	Size and number of mapped objects exceed permissible PDO length
0604 0043h	General parameter incompatibility
0606 0000h	SDO hardware fault
0607 0010h	Data type incompatible, length of service parameter is incompatible
0609 0011h	Subindex does not exist
0609 0030h	Outside value range for the parameter (only for write access)
0609 0031h	Parameter value too high
0609 0032h	Parameter value too low
0800 0020h	Data cannot be transmitted or saved
0800 0022h	Data cannot be transmitted or saved because of device status

Abort Codes not listed above are reserved.

5.4.6 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 setpoints and ROD feedback, digital speed setpoints 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 AKD. 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 appendix.
Basically, two types of PDOs can be distinguished, depending on the direction of transmission:

- Transmit-PDOs (TPDOs) (AKD => Master)
  The TPDOs transmit data from AKD to control system (for example actual value objects, instrument status).
- Receive-PDOs (RPDOs) (Master =>AKD)
  The RPDOs receive data from control system to AKD (for example setpoints).

AKD supports four independent PDO channels for each direction of transmission. The channels are labeled by the channel numbers 1 to 4.
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 (objects 1400h to 1403h, 1800h to 1803h).

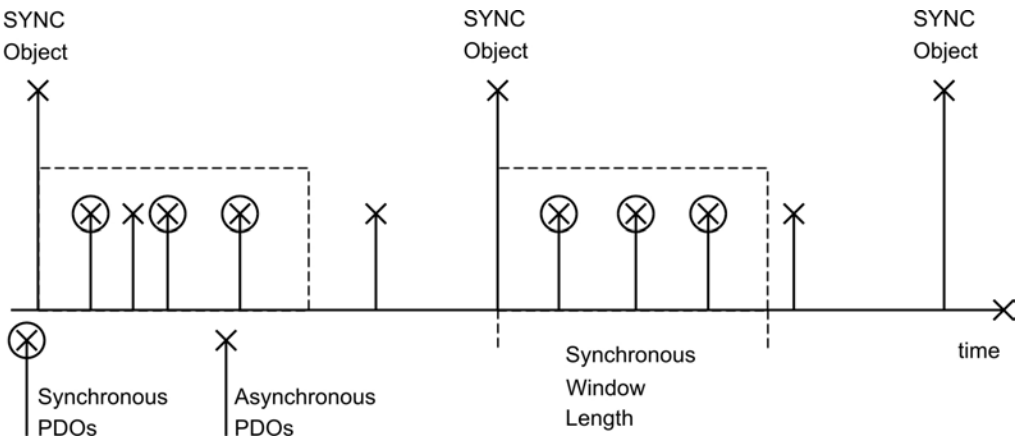
5.4.6.1 Transmission modes

The following PDO transmission modes are distinguished:

- 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.



5.4.6.2 Trigger modes

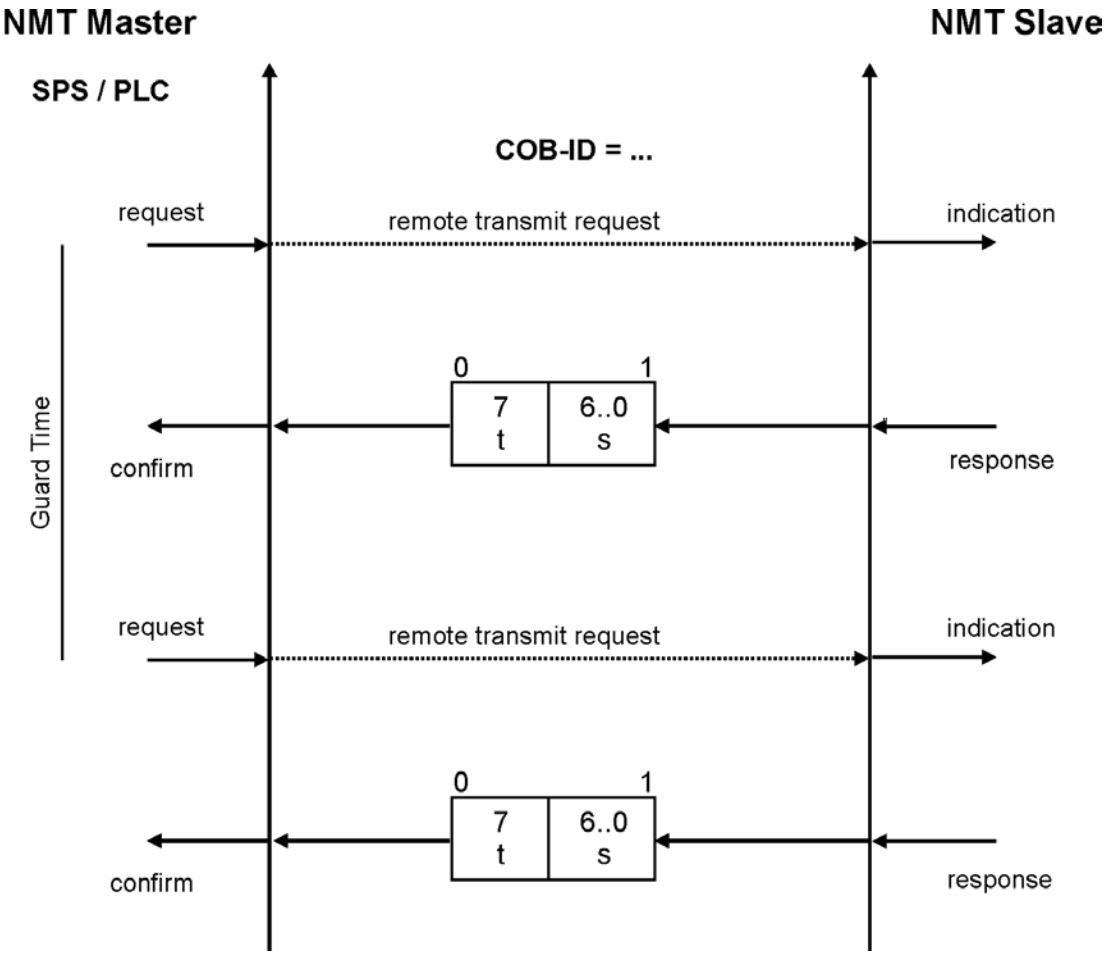
Three different trigger modes are distinguished:

- **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, Subindex 03h)
- **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.

5.4.7 Nodeguard

The Node Guarding protocol is a functional monitoring for the drive. It requires that the drive is accessed at regular intervals by the CANopen master.
The maximum time interval that is permitted between two Nodeguard telegrams is given by the product of the Guard Time (Object 100Ch) and the Life Time Factor (Object 100Dh). 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 objects 100Ch and 100Dh, then fault F129 (response monitoring) appears on the drive, the drive is braked to a stop, and any other movement is prevented.

The time sequence for node guarding is as shown below:



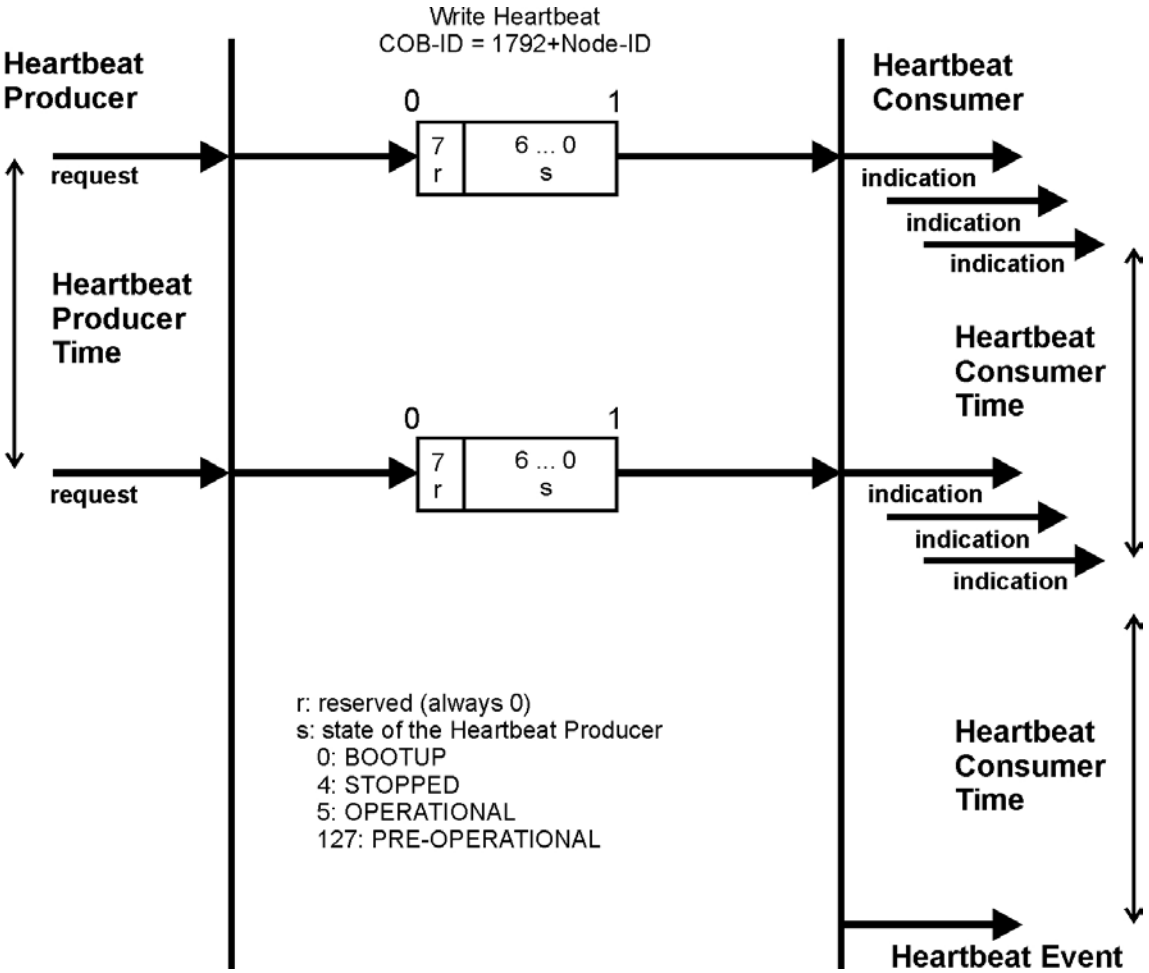
t = toggle Bit, changes its status with every slave telegram
s = status of the NMT slave status machine

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

5.4.8 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 Object 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 will be generated.

Heartbeat protocol:



6 CANopen Drive Profile

6.1 CANopen Emergency Messages and Error Codes

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.
Error numbers from 0000h to 7FFFh are defined in the communication or drive profile. Error numbers from FF00h to FFFFh have manufacturer-specific definitions. The following table describes the various error codes:

Error Code	Fault/Warning Code	Description
0x0000	0	Emergency error free
0x1080	-	General Warning
0x1081	-	General Error
0x3110	F523	Bus over voltage
0x3120	F247	Bus under voltage
0x3130	F503	Phase failure
0x3180	n503	Warning: Supply missing phase
0x3210	F501	DC link over-voltage
0x3220	F502	DC link under-voltage
0x3280	n502	Warning: Bus under voltage.
0x3281	n521	Warning: Dynamic Braking I2T.
0x3282	F519	Regen short circuit.
0x4210	F234	Excess temperature, device (control board)
0x4310	F235	Excess temperature, drive (heat sink)
0x4380	F236	Power temperature sensor 2 high
0x4381	F237	Power temperature sensor 3 high.
0x4382	F535	
0x4390	n234	Warning: Control temperature sensor 1 high.
0x4391	n235	Warning: Power temperature sensor 1 high.
0x4392	n236	Warning: Power temperature sensor 2 high.
0x4393	n237	Warning: Power temperature sensor 3 high.
0x4394	n240	Warning: Control temperature sensor 1 low.
0x4395	n241	Warning: Power temperature sensor 1 low.
0x4396	n242	Warning: Power temperature sensor 2 low.
0x4397	n243	Warning: Control temperature sensor 1 low.
0x4398	F240	Control temperature sensor 1 low.
0x4399	F241	Power temperature sensor 1 low.
0x439A	F242	Power temperature sensor 2 low.
0x439B	F243	Power temperature sensor 3 low.
0x5113	F512	5V0 under voltage
0x5114	F505	1V2 under voltage
0x5115	F507	2V5 under voltage
0x5116	F509	3V3 under voltage
0x5117	F514	+12V0 under voltage
0x5118	F516	-12V0 under voltage
0x5119	F518	Analog 3V3 under voltage
0x5180	F504	1V2 over voltage
0x5181	F506	2V5 over voltage

Error Code	Fault/Warning Code	Description
0x5182	F508	3V3 over voltage
0x5183	F510	5V0 over voltage
0x5184	F513	+12V0 over voltage
0x5185	F515	-12V0 over voltage
0x5186	F517	Analog 3V3 over voltage
0x5510	F201	Internal RAM failed.
0x5530	F105	Hardware memory, non-volatile memory stamp invalid.
0x5580	F106	Hardware memory, non-volatile memory data
0x5581	F202	Hardware memory, external Ram for resident firmware failed
0x5582	F203	Hardware memory, code integrity failed for resident firmware
0x5583	F102	Hardware memory, resident firmware failed
0x5584	F103	Hardware memory, resident FPGA failed
0x5585	F104	Hardware memory, operational FPGA failed
0x6380	F532	Drive motor parameters setup incomplete.
0x7180	F301	Motor overheat
0x7182	F305	Motor open circuit
0x7183	F306	Motor short circuit
0x7184	F307	Motor brake applied
0x7185	F436	
0x7186	n301	Warning: Motor overheated.
0x7187	F308	Voltage exceeds motor rating.
0x7303	F426	Resolver 1 fault
0x7305	F417	Incremental sensor 1 fault
0x7380	F402	Feedback 1 analogue fault
0x7381	F403	Feedback 1 EnDat communication fault
0x7382	F404	Feedback 1 illegal hall
0x7383	F405	Feedback 1 BiSS watchdog
0x7384	F406	Feedback 1 BiSS multi cycle
0x7385	F407	Feedback 1 BiSS sensor
0x7386	F408	Feedback 1 SFD configuration
0x7387	F409	Feedback 1 SFD UART overrun
0x7388	F410	Feedback 1 SFD UART frame
0x7389	F412	Feedback 1 SFD UART parity
0x738A	F413	Feedback 1 SFD transfer timeout
0x738B	F415	Feedback 1 SFD mult. corrupt position
0x738C	F416	Feedback 1 SFD Transfer incomplete
0x738D	F418	Feedback 1 power supply fault
0x738E	F401	Feedback 1 failed to set feedback
0x7390	n414	Warning: SFD single corrupted position.
0x7391	F419	Encoder init procedure failed
0x7392	F534	Failed to read motor parameters from feedback device.
0x73A0	F424	Feedback 2 resolver amplitude low
0x73A1	F425	Feedback 2 resolver amplitude high
0x73A2	F425	Feedback 2 resolver fault
0x73A3	F427	Feedback 2 analogue low

Error Code	Fault/Warning Code	Description
0x73A4	F428	Feedback 2 analogue high
0x73A5	F429	Feedback 2 incremental low
0x73A6	F430	Feedback 2 incremental high
0x73A7	F431	Feedback 2 halls
0x73A8	F432	Feedback 2 communication
0x73A9	-	Reserved
0x73AA	-	Reserved
0x73C0	F473	Wake and Shake. Insufficient movement
0x73C1	F475	Wake and Shake. Excess movement.
0x73C2	F476	Wake and Shake. Fine-coarse delta too large.
0x73C3	F478	Wake and Shake. Overspeed.
0x73C4	F479	Wake and Shake. Loop angle delta too large.
0x73C5	F482	Commutation not initialized
0x73C6	F483	Motor U phase missing.
0x73C7	F484	Motor V phase missing.
0x73C8	F485	Motor W phase missing.
0x73C9	n478	Warning: Wake and Shake. Overspeed.
0x73CA	n479	Warning: Wake and Shake. Loop angle delta too large.
0x8130	F129	Life Guard Error or Heartbeat Error
0x8180	n702	Warning: Fieldbus communication lost.
0x8280	F601	
0x8311	F304	Excess torque
0x8331	F524	Torque fault
0x8380	n524	Warning: Drive foldback
0x8381	n304	Warning: Motor foldback
0x8382	n309	Warning:
0x8480	F302	Velocity overspeed
0x8482	F480	Fieldbus command velocity too high
0x8481	F703	Emergency timeout occurred while axis should disable
0x8483	F481	Fieldbus command velocity too low.
0x8580	F107	Software limit switch, positive
0x8581	F108	Software limit switch, negative
0x8582	N107	Warning: Positive software position limit is exceeded.
0x8583	n108	Warning: Negative software position limit is exceeded.
0x8584	n704	Warning: PVT buffer overflow
0x8585	n705	Warning: PVT buffer underflow
0x8586	n127	Warning: Scale factor of PVT velocity command over range.
0x8611	F439	Following error
0x8684	n123	Warning: Motion global warning
0x8685	F138	Instability during autotune
0x8686	F151	Not enough distance to move; Motion Exception
0x8687	F152	Not enough distance to move; Following Motion Exception
0x8688	F153	Velocity Limit Violation, Exceeding Max Limit
0x8689	F154	Following Motion Failed; Check Motion Parameters
0x868a	F156	Target Position crossed due to Stop command

Error Code	Fault/Warning Code	Description
0x86a0	F157	Homing Index pulse not found
0x86a1	F158	Homing Reference Switch not found
0x86a2	F159	Failed to set motion task parameters
0x86a3	F160	Motion Task Activation Failed
0x86a4	F161	Homing Procedure Failed
0x86a5	F139	Target Position Over Short due to invalid Motion task activation.
0x86a6	n163	Warning: MT.NUM exceeds limit.
0x86a7	n164	Warning: Motion task is not initialized.
0x86a8	n165	Warning: Motion task target position is out.
0x86a9	n167	Warning:
0x86aa	n168	Warning: Invalid bit combination in the motion task control word.
0x86ab	n169	Warning: 1:1 profile cannot be triggered on the fly.
0x86ac	n170	Warning: Customer profile table is not initialized.
0x86ad	n171	Warning:
0x86ae	n172	Warning:
0x86B0	F438	Following error (numeric)
0x8780	F125	Fieldbus synchronization lost
0x8781	n125	Warning: Fieldbus PLL unlocked.
0x8AF0	F137	Homing and Feedback mismatch
0x8AF1	n140	Warning:
0xFF00	F701	Fieldbus runtime fault
0xFF01	F702	Fieldbus communication lost
0xFF02	F529	Iu offset limit exceeded
0xFF03	F530	Iv offset limit exceeded
0xFF04	F521	Stored energy reached critical point
0xFF05	F527	Iu detection stuck
0xFF06	F528	Iv detection stuck
0xFF07	F525	Control output over current
0xFF08	F526	Current sensor short circuit
0xFF09	F128	Axis dpoles
0xFF0A	F531	Power stage fault
0xFF0B	F602	Safe torque off
0xFF0C	F131	Emulated encoder line break.
0xFF0D	F130	Secondary feedback supply over current.
0xFF0E	F134	Secondary feedback illegal state.
0xFF0F	F245	External fault.
0xFF10	n414	Warning: SFD single corrupted position.
0xFF11	F101	Not compatible Firmware
0xFF12	n439	Warning: Following error (user)
0xFF13	n438	Warning: Following error (numeric)
0xFF14	n102	Warning: Operational FPGA is not a default FPGA.
0xFF15	n101	Warning: The FPGA is a laboratory FPGA
0xFF16	n602	Warning: Safe torque off.

6.2 General Definitions

This chapter describes objects with a general validity (e.g. Object 1000h Device Type). The next section explains the free configuration of Process Data Objects ("free mapping").

6.2.1 General objects

6.2.1.1 Object 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	402d=192h
31 24	23 16	15 0

The device profile number is DS402, the type is 2 for drives, the mode bits 28 to 31 are manufacturer specific and may be changed from its actual value of 0. A read access delivers 0x00020192 at the moment.

Index	1000h
Name	device type
Object code	VAR
Data type	UNSIGNED32
Category	mandatory

Access	R/O
PDO mapping	not possible
Value range	UNSIGNED32
Default value	no

6.2.1.2 Object 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	R/O
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	Bit	Description
0	generic error	4	communication error (overrun, error state)
1	current	5	device profile specific
2	voltage	6	reserved (always 0)
3	temperature	7	manufacturer specific

6.2.1.3 Object 1002h: Manufacturer Status Register (DS301)

The manufacturer status register contains important drive informations.

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

Access	R/O
PDO mapping	possible
Value range	UNSIGNED32
Default value	no

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

Bit	Description	Bit	Description
0	1 = Movement (positioning, homing) active	16	1 = Homing move active
1	reference position set	17	reserved
2	1 = reference switch high (home-position)	18	reserved
3	1 = In Position	19	1 = Emergency stop active
4	reserved	20	reserved
5	reserved	21	reserved
6	reserved	22	reserved
7	Active Disabel activated	23	1 = Homing move finished
8	Warning active	24	Power stage deactivating
9	1 = target velocity reached (pp- or pv-Mode)	25	1 = digital input 1 set
10	reserved	26	1 = digital input 2 set
11	1 = Homing error	27	1 = digital input 3 set
12	reserved	28	1 = digital input 4 set
13	1 = Safe Torque Off selected	29	1 = digital input hardware enable set
14	1 = Power stage enabled	30	1 = Wake and Shake action is required
15	1 = Error state	31	Braking, 1 = set points not accepted

6.2.1.4 Object 1003h: Predefined Error Field (DS301)

The object 1003h provides an error history with a maximum size of 10 entries.

Subindex 0 contains the number of errors which have occured 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 subindizes is defined in the field Error Code in the description of the Emergency Messages (=> p. 42).

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

Subindex	**0**
Description	Number of entries
Data type	UNSIGNED8
Category	mandatory
Access	R/W
PDO mapping	not possible
Value range	0 to 10
Default value	0

Subindex	**1 to 10**
Description	Standard error field (=> p. 42)
Category	optional
Access	R/O
PDO mapping	not possible
Value range	UNSIGNED32
Default value	no

6.2.1.5 Object 1005h: COB-ID of the SYNC Message (DS301)

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

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

Access	R/W
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 to 11	X	—
	0	if Bit 29=0
10 to 0 (LSB)	X	Bit 0 to 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.

6.2.1.6 Object 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	R/W
PDO mapping	not possible
Value range	UNSIGNED32
Default value	00h

6.2.1.7 Object 1008h: Manufacturer Device Name (DS301)

The device name consists of four ASCII characters in the form Yzzz, whereby Y stands for the mains voltage (L, M, H or U, e.g. H for High Voltage) zzz stands for 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	
Default value	no

6.2.1.8 Object 1009h: Manufacturer Hardware Version

This object will be supported in the future.

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	no

6.2.1.9 Object 100Ah: Manufacturer Software Version (DS301)

The object contains the manufacturer software version (here: the CANopen-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 to 9.99
Default value	no

6.2.1.10 Object 100Ch: Guard Time (DS301)Response monitoring

The arithmetical product of the Objects 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	R/W
PDO mapping	not possible
Value range	UNSIGNED16
Default value	0

6.2.1.11 Object 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	R/W
PDO mapping	not possible
Value range	UNSIGNED8
Default value	0

6.2.1.12 Object 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 (DRV.NVSAVE)
Object code	ARRAY
Data type	UNSIGNED32
Category	optional

Subindex	**0**
Name	number of entries
Object code	VAR
Data type	UNSIGNED8
Category	mandatory
Access	R/O
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	R/W
PDO Mapping	not possible
Value range	UNSIGNED32
Default value	1

Data definition:

Bit	Value	Meaning
31 to 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 subindex 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 Object 1010 sub 1. In general the drive does not save parameters autonomously with the exception of e.g. the special treatment of the homing of multiturn 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 65766173h.

6.2.1.13 Object 1011h: Restore Default Parameters DS301

With this object the default values of parameters according to the communication or device profile are restored. The AKD gives the possibility to restore all default values.

Index	1011h
Name	restore default parameters (DRV.NVLOAD)
Object code	ARRAY
Data type	UNSIGNED32
Category	optional

Subindex	0
Name	number of entries
Object code	VAR
Data type	UNSIGNED8
Category	mandatory
Access	R/O
PDO Mapping	not possible
Value range	1
Default value	1

Subindex	1
Name	restore all default parameters
Object code	VAR
Data type	UNSIGNED32
Category	mandatory
Access	R/W
PDO Mapping	not possible
Value range	UNSIGNED32
Default value	1 (device restores parameter)

Restoring of parameters will only be done, if a special signature ("load") is written to subindex 1. "load" has to be transmitted as unsigned32 - number 64616F6Ch.

6.2.1.14 Object 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	R/O
PDO mapping	not possible
Value range	UNSIGNED32
Default value	80h + Node - ID

6.2.1.15 Object 1016h: Consumer Heartbeat Time

The consumer heartbeat time defines the expected heartbeat cycle time (ms) and must 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 ms the corresponding entry is not used.

Index	**1016h**
Name	consumer heartbeat time
Object code	ARRAY
Data type	UNSIGNED32
Category	optional
Subindex	**0**
Description	number of entries
Data type	UNSIGNED8
Category	mandatory
Access	R/O
PDO Mapping	not possible
Value range	1
Default value	1
Subindex	**1**
Description	Consumer heartbeat time
Category	mandatory
Access	R/W
PDO Mapping	not possible
Value range	unsigned 32
Default value	no

Definition of the entry value of Subindex 1

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

6.2.1.16 Object 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	R/W
PDO mapping	not possible
Value range	UNSIGNED16
Default value	0

6.2.1.17 Object 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
Data type	UNSIGNED8
Category	mandatory
Access	R/O
PDO mapping	not possible
Value range	1 to 4
Default value	4

Subindex 1 is a unique number for a device manufacturer.

Subindex	1
Description	Vendor ID
Category	mandatory
Access	R/O
PDO mapping	not possible
Value range	UNSIGNED32
Default value	0x6Ah (Danaher Motion)

Subindex 2 contains four ASCII - characters, which determine the voltage range and current class of the device. The voltage range is one character L, M or H for low, medium and high voltage. The next three characters are showing the continuos current of the drive.

Subindex	**2**
Description	Product Code
Category	optional
Access	R/O
PDO mapping	not possible
Value range	e.g. M006 for an MV6 drive
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 is not used in the AKD. The firmware version can be retrieved as a string via object 0x100A or as numbers voa object 0x2018 subindex 1 to 4.

E.g. a value of 0x0014 0000 means CAN-version 0.20.

Subindex	**3**
Description	Revision Number
Category	optional
Access	R/O
PDO mapping	not possible
Value range	UNSIGNED32
Default value	no

Subindex 4 gives the serial number of the drive. This number contains the following information in it:

- bits 0..14: Board serial number (production in week of year)
- bits 15..20: week of production
- bits 21..24: year of production - 2009
- bits 25..31: ASCII-code of MFR-ID

Subindex	**4**
Description	Serial Number
Category	optional
Access	R/O
PDO mapping	not possible
Value range	UNSIGNED32
Default value	no

6.2.1.18 Object 1026h: OS Prompt

The OS prompt is used to build up 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
Data type	UNSIGNED8
Category	mandatory
Access	R/O
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	W
PDO mapping	not possible
Value range	UNSIGNED8
Default value	—

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

Subindex	2
Description	StdOut
Category	mandatory
Access	R/O
PDO mapping	not possible
Value range	UNSIGNED8
Default value	0

6.2.2 Manufacturer specific objects

6.2.2.1 Object 2014-2017h: 1st-4th Mask 1 to 4 for Transmit-PDO

In order to reduce the bus loading with event-triggered PDOs, masking can be used to switch off the monitoring for individual bits in the PDO. In this way it can be arranged, for instance, that actual position values are only signaled once per turn.
This Object masks the PDO-channels 1 to 4. If only two bytes have been defined in a PDO, then it masks just two bytes, although 4 bytes of mask information have been transmitted.
An activated bit in the mask means that monitoring is active for the corresponding bit in the PDO.

Index	**2014h** **2015h** **2016h** **2017h**
Name	tx_mask 1 to 4
Object code	ARRAY
Data type	UNSIGNED32

Subindex	**1**
Description	tx_mask1 to 4_low
Mode	independent
Access	R/W
PDO mapping	not possible
Unit	—
Value range	UNSIGNED32
Default value	FFFFFFFFh

Subindex	**2**
Description	tx_mask1 to 4_high
Mode	independent
Access	R/W
PDO mapping	not possible
Unit	—
Value range	UNSIGNED32
Default value	FFFFFFFFh

6.2.2.2 Object 2018h: Firmware Version

This object gives all information regarding the firmware version.
Example: Firmware version M_01_00_01_005 would show the numbers 1, 0, 1, 5 in the sub-indices 1 to 4.

Index	2018h
Name	firmware version
Object code	ARRAY
Data type	UNSIGNED16

Subindex	1
Description	major version
Mode	independent
Access	R/O
PDO mapping	not possible
Unit	—
Value range	UNSIGNED16
Default value	0

Subindex	2
Description	minor version
Mode	independent
Access	R/O
PDO mapping	not possible
Unit	—
Value range	UNSIGNED16
Default value	0

Subindex	3
Description	revision
Mode	independent
Access	R/O
PDO mapping	not possible
Unit	—
Value range	UNSIGNED16
Default value	0

Subindex	4
Description	branch revision
Mode	independent
Access	R/O
PDO mapping	not possible
Unit	—
Value range	UNSIGNED16
Default value	0

6.2.2.3 Object 2026h: ASCII Channel

This object is used to build up an ASCII - communication channel to the drive with 4-byte ASCII-strings.

Index	2026h
Name	ASCII Channel
Object code	ARRAY
Data type	Visible String
Category	optional

Subindex	0
Description	Number of entries
Data type	UNSIGNED8
Category	mandatory
Access	R/O
PDO mapping	not possible
Value range	2
Default value	2

Subindex 1 is used to send four ASCII-characters to the drive.

Subindex	1
Description	Command
Category	mandatory
Access	wo
PDO mapping	no
Value range	Visible String
Default value	—

Subindex 2 is used to receive four characters from the drive.

Subindex	2
Description	Response
Category	mandatory
Access	R/O
PDO mapping	no
Value range	Visible String
Default value	-

6.2.2.4 Object 20A0h: Latch position 1, positive edge

This object is used to output the position or a time, depending on CAP0.MODE, at which the first positive edge occurred on a signal, which can be configured with the command CAP0.TRIGGER. The latch enable must be active for that purpose(see object 20A4 and 20A5). With CAP0.MODE = 3 the latched position of the encoder index pulse is transferred via this object.

Index	**20A0h**
Name	Latch position 1, positive edge, CAP0.PLFB Time capture, CAP0.T
Object code	VAR
Data type	INTEGER32
Category	optional

Access	R/O
PDO mapping	possible
Value range	INTEGER32
Default value	0

6.2.2.5 Object 20A1h: Latch position 1, negative edge

This object is used to output the position or a time, depending on CAP0.MODE, at which the first negative edge occurred on a signal, which can be configured with the command CAP0.TRIGGER. The latch enable must be active for that purpose(see object 20A4 and 20A5).

Index	**20A1h**
Name	Latch position 1, negative edge, CAP0.PLFB Time capture, CAP0.T
Object code	VAR
Data type	INTEGER32
Category	optional

Access	R/O
PDO mapping	possible
Value range	INTEGER32
Default value	0

6.2.2.6 Object 20A2h: Latch position 2, positive edge

This object is used to output the position or a time, depending on CAP1.MODE, at which the first positive edge occurred on a signal, which can be configured with the command CAP1.TRIGGER. The latch enable must be active for that purpose(see object 20A4 and 20A5).

Index	**20A2h**
Name	Latch position 2, positive edge, CAP1.PLFB Time capture, CAP1.T
Object code	VAR
Data type	INTEGER32
Category	optional

Access	R/O
PDO mapping	possible
Value range	INTEGER32
Default value	0

6.2.2.7 Object 20A3h: Latch position 2, negative edge

This object is used to output the position or a time, depending on CAP0.MODE, at which the first negative edge occurred on a signal, which can be configured with the command CAP0.TRIGGER. The latch enable must be active for that purpose(see object 20A4 and 20A5).

Index	**20A3h**
Name	Latch position 2, negative edge, CAP1.PLFB Time capture, CAP1.T
Object code	VAR
Data type	INTEGER32
Category	optional

Access	R/O
PDO mapping	possible
Value range	INTEGER32
Default value	0

6.2.2.8 Object 20A4h: Latch Control Register

The latch control register is used to enable the latch monitoring of the capture engines 0 and 1. The latch is enabled with a 1 signal and disabled with a 0 signal. Whether or not a latch event has occurred can be recognised by the latch status register (object 20A5).

Index	**20A4h**
Name	Latch Control Register
Object code	VAR
Data type	UNSIGNED16
Category	optional

Access	rww
PDO mapping	possible
Value range	0 to 15
Default value	0

Bit	Value (bin)	Value (hex)	Description
0	00000000 00000001	xx01	Enable extern latch 1 (positive rise)
1	00000000 00000010	xx02	Enable extern latch 1 (negative rise)
2	00000000 00000100	xx04	Enable extern latch 2 (positive rise)
3	00000000 00001000	xx08	Enable extern latch 2 (negative rise)
4	00000000 00010000	xx10	Enable latch of encoder index pulse
5 to 7			Reserve
8	00000001 00000000	01xx	Read external latch 1 (positive rise)
9	00000010 00000000	02xx	Read external latch 1 (negative rise)
10	00000011 00000000	03xx	Read external latch 2 (positive rise)
11	00000100 00000000	04xx	Read external latch 2 (negative rise)
12	00000101 00000000	05xx	Read latched position of encoder index pulse
13 to 15			Reserve

6.2.2.9 Object 20A5h: Latch Status Register

The latch status register is used to look for the states of the capture engines 0 and 1.

Index	**20A5h**
Name	Latch Status Register
Object code	VAR
Data type	UNSIGNED16
Category	optional

Access	rwr
PDO mapping	possible
Value range	-
Default value	0

Bit	Value (bin)	Value (hex)	Description
0	00000000 00000001	zz01	External latch 1 valid (positive rise)
1	00000000 00000010	zz02	External latch 1 valid (negative rise)
2	00000000 00000100	zz04	External latch 2 valid (positive rise)
3	00000000 00001000	zz08	External latch 2 valid (negative rise)
4	00000000 00010000	z10	Latched position of encoder index pulse valid (positive rise)
5 to 7			Reserve
8 to 11	00000001 00000000	z1zz	Acknowledge value external latch 1 (positive rise)
	00000010 00000000	z2zz	Acknowledge value external latch 1 (negative rise)
	00000011 00000000	z3zz	Acknowledge value external latch 2 (positive rise)
	00000100 00000000	z4zz	Acknowledge value external latch 2 (negative rise)
	00000101 00000000	z5zz	Acknowledge value of latched position of encoder index pulse (positive rise)
12 to 15	00010000 00000000	1zzz	State Digital Input 4
	00100000 00000000	2zzz	State Digital Input 3
	01000000 00000000	4zzz	State Digital Input 2
	10000000 00000000	8zzz	State Digital Input 1

6.2.2.10 Object 20A6h: Latch position 1, positive or negative edge

This object is used to output the position or a time, depending on CAP0.MODE, at which the first positive or negative edge occurred on a signal, which can be configured with the command CAP0.TRIGGER.

The latch enable must be active for that purpose (see object 20A4 and 20A5).

Index	**20A6h**
Name	Latch position 1, positive or negative, CAP0.PLFB
Object code	VAR
Data type	INTEGER32
Category	optional

Access	ro
PDO mapping	possible
Value range	INTEGER32
Default value	0

6.2.2.11 Object 20B8h: Reset of changed input information

This object is used in PDOs to reset the state change information for the digital inputs shown in the Bits 24 to 30 in the object 60FD. Bit 0 to 6 are used to reset the information of the digital input 1 to 7.

Index	**20B8h**
Name	Reset of changed input information
Object code	VAR
Data type	UNSIGNED16
Category	optional
Access	rw
PDO mapping	possible
Value range	UNSIGNED16
Default value	0

6.2.2.12 Object 3474h: Parameters for digital inputs

This set of objects is used to set extended parameters for some digital input functions. The parameters can be used for different DINx.MODEs. Therefore thescaling might be different or no scaling is used at all.

Two subindices build an access object to one of these parameters, because they are 64-bit numbers internally, e.g. object 3474 sub 1 gives access to the low 32 bits of DIN1.PARAM whereas 3474 sub 8 gives access to the high 32 bits.

If access to the whole 64 bit number is needed the higher bits must be written first. The access to the lower 32 bits then writes the parameter. If the to be written value fits into 32 bit, only the lower part needs to be written. The most-significant bit is then taken as sign-bit for the number.

Index	**3474h**
Name	DINx.PARAM
Object code	Array
Data type	UNSIGNED32
Category	optional

Subindex	**0**
Description	number of entries
Category	mandatory
Access	R/O
PDO mapping	not possible
Value range	0xE
Default value	0xE

Subindex	**1 to 7**
Description	DINx.PARAM low 32 bits, x = 1 .. 7
Category	optional
Access	R/W
PDO mapping	not possible
Value range	UNSIGNED32
Default value	0

Subindex	**8 to 0xE**
Description	DINx.PARAM high 32 bits, x = 1 .. 7
Category	optional
Access	R/W
PDO mapping	not possible
Value range	UNSIGNED32
Default value	0

6.2.2.13 Object 3475h: Parameters for digital outputs

This set of objects is used to set extended parameters for some digital output functions. The parameters can be used for different DOUTx.MODEs. Therefore the scaling might be different or no scaling is used at all.

Two subindices build an access object to one of these parameters, because they are 64-bit numbers internally, e.g. object 3475 sub 1 gives access to the low 32 bits of DOUT1.PARAM whereas 3475 sub 3 gives access to the high 32 bits.

If access to the whole 64 bit number is needed the higher bits must be written first. The access to the lower 32 bits then writes the parameter. If the to be written value fits into 32 bit, only the lower part needs to be written. The most-significant bit is then taken as sign-bit for the number.

Index	**3475h**
Name	DOUTx.PARAM
Object code	Array
Data type	UNSIGNED32
Category	optional

Subindex	**0**
Description	number of entries
Category	mandatory
Access	R/O
PDO mapping	not possible
Value range	0x4
Default value	0x4

Subindex	**1 to 2**
Description	DOUTx.PARAM low 32 bits, x = 1 .. 2
Category	optional
Access	R/W
PDO mapping	not possible
Value range	UNSIGNED32
Default value	0

Subindex	**3 to 4**
Description	DOUTx.PARAM high 32 bits, x = 1 .. 2
Category	optional
Access	R/W
PDO mapping	not possible
Value range	UNSIGNED32
Default value	0

6.2.2.14 Object 3496h: Fieldbus synchronization parameters

This set of objects is used to set or read parameters for the fieldbus synchronization used in the interpolated position mode (7) and the cyclic-modes 8 etc. The synchronization between a fieldbus master and the AKD is similar in all the supported fieldbus systems.

The AKD internal 16[kHz] interrupt function is responsible for calling the PLL function. This PLL function is called once per fieldbus cycle (set by object 60C2 sub 1 and 2).If the fieldbus sample period is for example 1[ms], the PLL code is called every 16th time of the 16[kHz] IRQ of the AKD.

Once in a fieldbus sample the SYNC-telegram must arrive, which resets a PLL counter in the Drive. After some time the already mentioned PLL function is called and reads back the time from that PLL counter.

Depending on the measured time the PLL function extends (in case that the measured time is too low) or lowers (in case that the measured time is too high) the sample time of the upcoming 16[kHz] tasks for one fieldbus sample by a selectable value (object 3496 sub 4) in order to move the PLL function closer to the expected distance (object 3496 sub 1).

Beside the objects mentioned here the parameter FBUS.SAMPLEPERIOD is important, which is set by object 60C2 sub 1 and 2.This setting is required in order to share the fieldbus sample time with the slave. This information is e.g. needed for being able to call the AKD internal PLL function once per fieldbus sample.

Index	**3496h**
Name	FBUS synchronization parameters
Object code	Array
Data type	UNSIGNED32
Category	optional

Subindex	**0**
Description	number of entries
Category	mandatory
Access	R/O
PDO mapping	not possible
Value range	0x4
Default value	0x4

Subindex	**1**
Description	FBUS.SYNCDIST
Category	optional
Access	R/W
PDO mapping	not possible
Value range	UNSIGNED32
Default value	320000 [ns]

Sub 1 is the expected time distance in nano seconds between clearing the PLL counter and calling the PLL function.

Subindex	**2**
Description	FBUS.SYNCACT
Category	optional
Access	R/W
PDO mapping	not possible
Value range	UNSIGNED32
Default value	320000 [ns]

Sub 2 is the actual time distance in nano seconds between clearing the PLL counter and calling the PLL function.

Subindex	3
Description	FBUS.SYNCWND
Category	optional
Access	R/W
PDO mapping	not possible
Value range	UNSIGNED32
Default value	70000 [ns]

Sub 3 is a window, which is used in order to consider the drive as being synchronized. The AKD is considered as synchronized in the following case:

FBUS.SYNCDIST – FBUS.SYNCWND < FBUS.SYNCACT < FBUS.SYNCDIST + FBUS.SYNCWND

Subindex	4
Description	FBUS.COMPTIME
Category	optional
Access	R/W
PDO mapping	not possible
Value range	UNSIGNED32
Default value	150 [ns]

Sub 4 value indicates the time, which is used for extending or lowering the sample rate of the AKD internal 16[kHz] IRQ, which is responsible for calling the PLL function. The default sample time is 32 * 1/16[kHz] = 2[ms].

The sample time of the AKD high prior interrupt is determined by
62.5[µs] – FBUS.COMPTIME if FBUS.SYNCACT > FBUS.SYNCDIST.

The sample time of the AKD high prior interrupt is determined by
62.5[µs] + FBUS.COMPTIME if FBUS.SYNCACT < FBUS.SYNCDIST.

6.2.3 Profile specific objects

6.2.3.1 Object 60B8h: Touch probe function

This object indicates the configured function of the touch probe.

Index	**60B8h**
Name	Touch probe function
Object code	Variable
Data type	UNSIGNED16
Category	optional
Access	R/W
PDO Mapping	yes
Value range	UNSIGNED16
Default value	0

Definition of the possible functions:

Bit	Value	Meaning
0	0	Switch off touch probe 1
	1	Enable touch probe 1
1	0	Trigger first event
	1	Continuous
3, 2	00b	Trigger with touch probe 1 input
	01b	Trigger with zero impulse signal or position encoder
	10b	Touch probe source as defined in object 60D0h, sub-index 01h
	11b	reserved
4	0	Switch off sampling at positive edge of touch probe 1
	1	Enable sampling at positive edge of touch probe 1
5	0	Switch off sampling at negative edge of touch probe 1
	1	Enable sampling at negative edge of touch probe 1
6, 7	-	User-defined (e.g. for testing)
8	0	Switch off touch probe 2
	1	Enable touch probe 2
9	0	Trigger first event
	1	continuous
11, 10	00b	Trigger with touch probe 2 input
	01b	Trigger with zero impulse signal or position encoder
	10b	Touch probe source as defined in object 60D0h, sub-index 02h
	11b	reserved
12	0	Switch off sampling at positive edge of touch probe 2
	1	Enable sampling at positive edge of touch probe 2
13	0	Switch off sampling at negative edge of touch probe 2
	1	Enable sampling at negative edge of touch probe 2
14, 15	-	User-defined (e.g. for testing)

6.2.3.2 Object 60B9h: Touch probe status

This object indicates the status of the touch probe.

Index	**60B9h**
Name	Touch probe status
Object code	Variable
Data type	UNSIGNED16
Category	optional
Access	R/O
PDO Mapping	yes
Value range	UNSIGNED16
Default value	0

Definition of the status:

Bit	Value	Meaning
0	0	Touch probe 1 is switched off
	1	Touch probe 1 is enabled
1	0	Touch probe 1 no positive edge value stored
	1	Touch probe 1 positive edge position stored
2	0	Touch probe 1 no negative edge value stored
	1	Touch probe 1 negative edge position stored
3 to 5	0	reserved
6, 7	-	User-defined (e.g. for testing)
8	0	Touch probe 2 is switched off
	1	Touch probe 2 is enabled
9	0	Touch probe 2 no positive edge value stored
	1	Touch probe 2 positive edge position stored
10	0	Touch probe 2 no negative edge value stored
	1	Touch probe2 negative edge position stored
11 to 13	0	reserved
14, 15	-	User-defined (e.g. for testing)

6.2.3.3 Object 60BAh: Touch probe 1 positive edge

This object provides the position value of the touch probe 1 at positive edge.

Index	60BAh
Name	Touch probe 1 positive edge
Object code	Variable
Data type	INTEGER32
Category	optional
Access	R/O
PDO Mapping	yes
Value range	INTEGER32
Default value	no

6.2.3.4 Object 60BBh: Touch probe 1 negative edge

This object provides the position value of the touch probe 1 at negative edge.

Index	60BBh
Name	Touch probe 1 negative edge
Object code	Variable
Data type	INTEGER32
Category	optional
Access	R/O
PDO Mapping	yes
Value range	INTEGER32
Default value	no

6.2.3.5 Object 60BCh: Touch probe 2 positive edge

This object provides the position value of the touch probe 2 at positive edge.

Index	60BCh
Name	Touch probe 2 positive edge
Object code	Variable
Data type	INTEGER32
Category	optional
Access	R/O
PDO Mapping	yes
Value range	INTEGER32
Default value	no

6.2.3.6 Object 60BDh: Touch probe 2 negative edge

This object provides the position value of the touch probe 2 at negative edge.

Index	60BDh
Name	Touch probe 2 negative edge
Object code	Variable
Data type	INTEGER32
Category	optional
Access	R/O
PDO Mapping	yes
Value range	INTEGER32
Default value	no

6.2.3.7 Object 60D0h: Touch probe source

This object provides the source of the touch probe function, when the dedicated bits 2/3 or 10/11 of the touch probe function (object 60B8h) are set accordingly.

Index	60D0h
Name	Touch probe source
Object code	Array
Data type	Integer 16
Category	optional

Subindex	0
Description	Highest sub-index supported
Category	mandatory
Access	R/O
PDO mapping	not possible
Value range	2
Default value	2

Subindex	1
Description	Touch probe 1 source
Category	mandatory
Access	R/W
PDO mapping	not possible
Value range	-11 to -1, 1 to 5
Default value	1

Subindex	2
Description	Touch probe 2 source
Category	mandatory
Access	R/W
PDO mapping	not possible
Value range	-11 to -1, 1 to 5
Default value	1

6.2.3.8 Object 60FDh: Digital inputs (DS402)

This index defines simple digital inputs for drives. The manufacturer bits 16 to 22 are used to mirror the digital inputs 1 to 7. The manufacturer bits 24 to 30 are used to show the change of the state of the digital inputs 1 to 7.

Index	**60FDh**
Name	digital inputs
Object code	VAR
Data type	UNSIGNED32
Category	optional
Access	R/O
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

6.2.3.9 Object 60FEh: Digital outputs (DS402)

This index defines simple digital outputs for drives. The manufacturer bits 16 and 17 are used to mirror the digital outputs 1 and 2.

Index	**60FEh**
Name	digital outputs
Object code	Array
Data type	UNSIGNED32
Category	optional

Subindex	**0**
Description	number of entries
Category	mandatory
Access	R/O
PDO mapping	not possible
Value range	2
Default value	2

Subindex	**1**
Description	physical outputs
Category	mandatory
Access	R/W
PDO mapping	possible
Value range	UNSIGNED32
Default value	0

Subindex	**2**
Description	bit mask
Category	optional
Access	R/W
PDO mapping	not possible
Value range	UNSIGNED32
Default value	0

31 18	17	16	15 1	0
manufacturer specific	DOUT2	DOUT1	reserved	set brake
MSB				LSB

6.2.3.10 Object 6502h: Supported drive modes (DS402)

A drive can support more then 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	R/O
PDO mapping	not possible
Value range	UNSIGNED32
Default value	0x65 (ip hm 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

6.3 PDO Configuration

PDOs are used for process data communication. There are two types of PDOs: Receive PDOs (RPDOs) and transmit PDOs (TPDOs).
The content of the PDOs is pre-defined (see descriptions on pages => p. 78 and => p. 81). If the data content is not appropriate for a special application the data objects in the PDOs can be remapped freely.

One data entry in the PDOs looks like this:

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. Stop possible transmission of the PDO.

COB-ID	Control byte	Index		Sub-index	Data	Comment
		Low byte	High byte			
601	23	´00	18	01h	81 01 00 C0	Switch-off COB-Id

2. Delete the actual mapping of the PDO by writing a 0 to the subindex 0 of the mapping Object.

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

3. Build the mapping with object dictionary objects (see page => p. 117)) which are mappable, e.g.

COB-ID	Control byte	Index		Sub-index	Data	Comment
		Low byte	High byte			
601	23	00	1A	01h	10 00 41 60	1st entry: CANopen statusword with 16 bits
601	23	00	1A	02h	20 00 02 10	2nd entry: Manufacturer status with 32 bits

3. Write the number of mapped objects to subindex 0 of the mapping Object.

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

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

6.3.1 Receive PDOs (RXPDO)

Four Receive PDOs can be configured in the drive:

- - configuration of the communication (Objects 1400-1403h)
- - configuration of the PDO-contents (mapping, Objects 1600-1603h)

6.3.1.1 Objects 1400-1403h: 1st - 4th RXPDO communication parameter (DS301)

Index	**1400h to 1403h for RXPDO 1 to 4**
Name	receive PDO parameter
Object code	RECORD
Data type	PDO CommPar
Category	mandatory

Defined sub-indices

Subindex	**0**
Name	number of entries
Data type	UNSIGNED8
Category	mandatory
Access	R/O
PDO Mapping	not possible
Value Range	2
Default Value	2

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

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

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

Subindex	2
Name	transmission type
Category	mandatory
Access	R/W
PDO Mapping	not possible
Value Range	UNSIGNED8
Default Value	FFh

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.

6.3.1.2 Objects 1600-1603h: 1st - 4th RXPDO mapping parameter (DS301)

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

Subindex	0
Name	number of entries
Data type	UNSIGNED8
Category	mandatory
Access	R/W
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 - 8
Name	PDO - mapping for the n-th application object
Category	Conditional, depends on number and size of object be mapped
Access	R/W
PDO Mapping	not possible
Value Range	UNSIGNED32
Default Value	See below

6.3.1.3 Default RXPDO definition

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)

6.3.2 Transmit PDOs (TXPDO)

Four Transmit PDOs can be configured in the drive:

- - configuration of the communication (Objects 1800-1803h)
- - configuration of the PDO-contents (mapping, Objects 1A00-1A03h)

6.3.2.1 Objects 1800-1803h: 1st - 4th TXPDO communication parameter (DS301)

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

Subindex	**0**
Name	number of entries
Data type	UNSIGNED8
Category	mandatory
Access	R/O
PDO Mapping	not possible
Value Range	5
Default Value	5

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

Subindex	**2**
Name	transmission type
Category	mandatory
Access	R/W
PDO Mapping	not possible
Value Range	UNSIGNED8
Default Value	FFh

Subindex	**3**
Name	inhibit time
Category	optional
Access	R/W
PDO Mapping	not possible
Value Range	UNSIGNED16 (n*1/10ms)
Default Value	0h

Subindex	4
Name	reserved
Category	optional
Access	R/W
PDO Mapping	not possible
Value Range	0
Default Value	0

Subindex	5
Name	event timer
Category	optional
Access	R/W
PDO Mapping	not possible
Value Range	UNSIGNED16 (0=not used, ms)
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 to 11	X	Identifier-bits with 29 bit-ID, not relevant
10 to 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 255d for an event-triggered PDO, which is sent immediately after a change in the mapped application objects. Setting of Subindex 3 or 5 has an influence on the sending of a PDO. With Subindex 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 Subindex 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 from 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.

6.3.2.2 Objects 1A00-1A03h: 1st - 4th TXPDO mapping parameter (DS301)

Index	1A00h - 1A03h 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
Data type	UNSIGNED8
Category	mandatory
Access	R/W
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 - 8
Name	PDO - mapping for the n-th application object
Category	Conditional, depends on number and size of object be mapped
Access	R/W
PDO Mapping	not possible
Value Range	UNSIGNED32
Default Value	See below

6.3.2.3 Default TXPDO definition

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

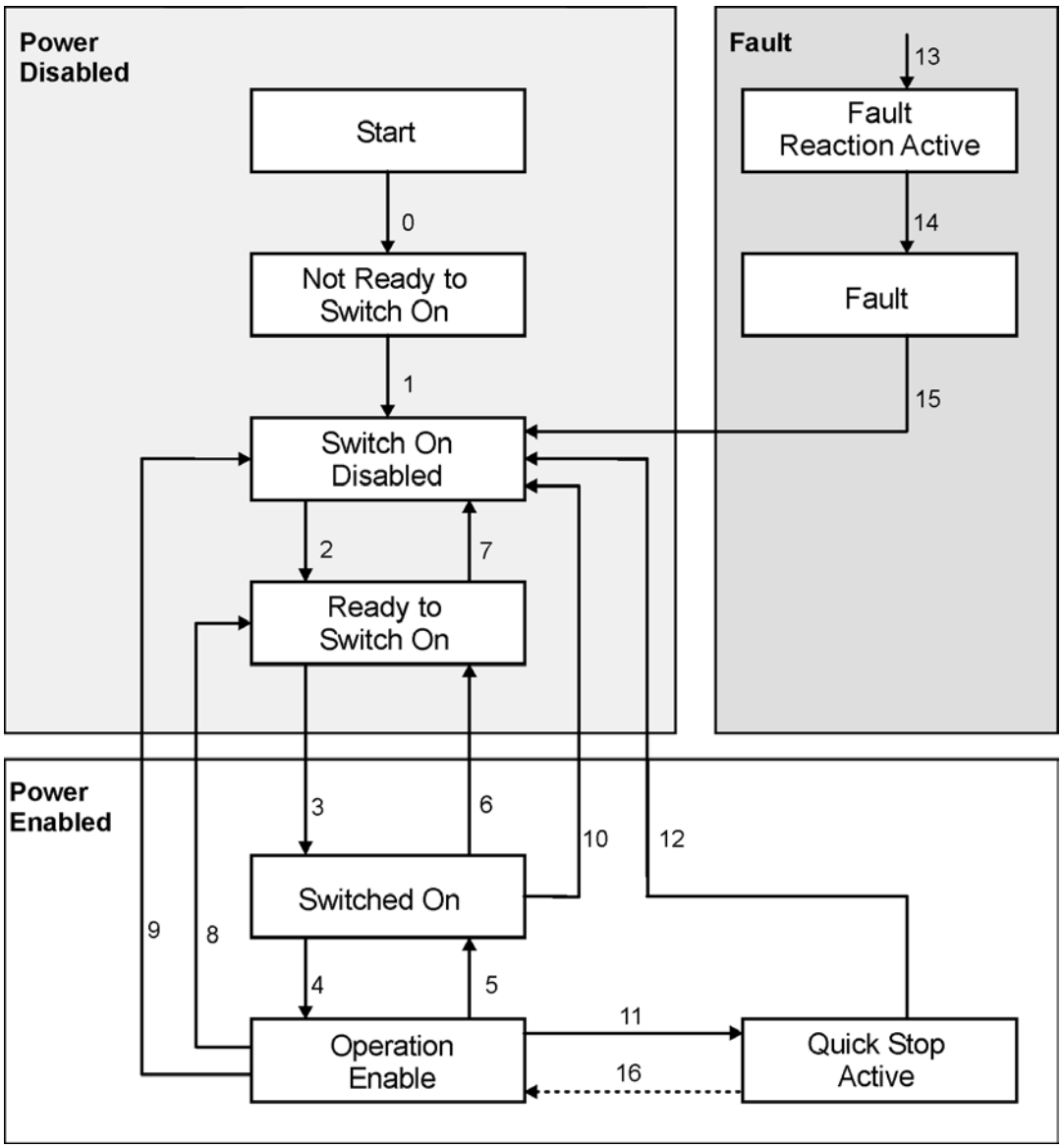
TXPDO 4:

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

6.4 Device Control (dc)

The device control of the AKD can be used to carry out all the motion functions in the corresponding modes. The control of the AKD is implemented through a mode-dependent status machine. The status machine is controlled through the control word (=> p. 88).
The mode setting is made through the object "Modes of Operation" (=> p. 91). The states of the status machine can be revealed by using the status word (=> p. 89).

6.4.1 Status Machine (DS402)



6.4.1.1 States of the Status Machine

State	Description
Not Ready for Switch On	AKD is not ready to switch on, there is no operational readiness (BTB/RTO) signaled from the controller program.
Switch On Disable	AKD 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 not 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.

6.4.1.2 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. AKD 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)	Output stage is disabled. Drive has no torque.
11	Bit 2 is canceled (Quick Stop command)	Drive is stopped with the emergency braking ramp. The output stage remains enabled. Setpoints are canceled (motion block number, digital setpoint, 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 controlword must be cleared by the host
16	Bit 2 is set	Motion function is enabled again.

NOTE **If the drive is operated through the control word/status word, then no control commands may be sent through another communication channel (ASCII channel, RS232).**

6.4.2 Object Description

6.4.2.1 Object 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	**6040h**
Name	control word
Object code	VAR
Data type	UNSIGNED16
Access	R/W
PDO mapping	possible
Unit	—
Value range	0 to 65535
EEPROM	no
Default value	0

Bit assignment im control word

Bit	Name	Bit	Name
0	Switch on	8	Pause/halt
1	Disable Voltage	9	reserved
2	Quick Stop	10	reserved
3	Enable Operation	11	reserved
4	Operation mode specific	12	reserved
5	Operation mode specific	13	Manufacturer-specific
6	Operation mode specific	14	Manufacturer-specific
7	Reset Fault (only effective for faults)	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 Object 6060_h Modes of operation.

Operation mode	No.	Bit 4	Bit 5	Bit 6
Profile Position Mode (pp)	01h	new_setpoint	change_set_immediately	absolute/relative
Profile Velocity Mode (pv)	03h	reserved	reserved	reserved
Profile Torque Mode (tq)	04h	reserved	reserved	reserved
Homing Mode (hm)	06h	homing_operation_start	reserved	reserved
Interpolated Position Mode (ip)	07h	Enable Interpolation	reserved	reserved
Cyclic sync position Mode (csp)	08h	reserved	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.

6.4.2.2 Object 6041h: Status word (DS402)

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

Index	**6041h**
Name	Status word
Object code	VAR
Data type	UNSIGNED16

Access	R/W
PDO mapping	possible
Unit	—
Value range	0 to 65535
EEPROM	yes
Default value	0

Bit assignment in the status word

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

States of the status machine

State	Bit 6 switch on disabled	Bit 5 quick stop	Bit 3 fault	Bit 2 operation enabled	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 of a warning can be seen by the Error code of the Emergency message, which is sent on the bus caused by this warning.

Bit 9: The remote-bit is set by the telnet command FBUS.REMOTE. The default state is 1 indicating that the power stage shall be only controlled by the DS402 control word. For special actions via telnet like tuning or commutation finding, FBUS.REMOTEshall be set to 0 via telnet to inform the fieldbus master.

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

Bit 11: internal_limit_active This bit specifies that a movement was or is limited. In different modes, different warnings cause the bit to be set. The following assignments exist:

Mode of operation	Warnings which set Bit 11
all	n04, n06, n07, n10, n11, n14
0x1 (PP), 0x88	n03, n08, n09, n20

6.4.2.3 Object 6060h: Modes of Operation (DS402)

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

- manufacturer-specific operating modes
- operating modes as per CANopen drive profile DS402

These operating modes are defined in the CANopen drive profile DS402. After the mode has been changed, the corresponding setpoint 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 drive.

NOTE	**An operating mode only becomes valid when it can be read by Object 6061h.**
⚠ WARNING	**Never change the mode while the motor is running! The drive could move unexpectedly. When the drive is enabled, a mode change is only permissible at zero speed. Set the speed setpoint to 0 before changing over.**

Index	**6060h**
Name	mode of operation
Object code	VAR
Data type	INTEGER8

Category	mandatory
Access	R/W
PDO mapping	possible
Value range	1, 3, 4, 6, 7, 8
Default value	—

Supported modes (negative values are manufacturer specific modes):

Value (hex)	Mode
1	Profile position mode
3	Profile velocity mode
4	Profile torque mode
6	Homing mode
7	Interpolated position mode
8	Cyclic sync position mode

6.4.2.4 Object 6061h: Modes of Operation Display (DS402)

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

Index	**6061h**
Name	mode of operation display
Object code	VAR
Data type	INTEGER8
Category	mandatory
Access	R/O
PDO mapping	possible
Value range	1, 3, 4, 6, 7, 8
Default value	—

6.5 Factor Groups (fg) (DS402)

The "factor groups" define the units of position-, velocity- and acceleration setpoints. These values are converted into drive-specific parameters.

Two types of scalings can be used depending on the configuration bit 4 in FBUS.PARAM05:

1. Scaling analog to Telnet. Then you should set the UNIT.PROTARY = 3, UNIT.VROTARY = 3 and UNIT.ACCROTARY = 3.

2. Scaling only with DS402 - scaling factors independant of the scalings used via Telnet. Therefore use the settings via the objects 204C / 6091/6092.

NOTE	**The drive parameters for the unit definitions should be set as follows:** **UNIT.PROTARY= 3 (UNIT.PIN/UNIT.POUT)** **UNIT.VROTARY = 3 (UNIT.PIN/UNIT.POUT/s)** **UNIT.ACCROTARY = 3 (c UNIT.PIN/UNIT.POUT/s²)**

6.5.1 General Information

6.5.1.1 Factors

You can convert between physical dimensions and sizes, and the internal units used in the device (increments). Several factors can be implemented. This chapter describes how these factors influence the system, how they are calculated and which data are necessary to build them.

6.5.1.2 Relationship between Physical and Internal Units

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:

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

6.5.2 Objects for velocity scaling

6.5.2.1 Object 204Ch: pv scaling factor

This object shall indicate the configured numerator and denominator of the pv scaling factor. The pv scaling factor serves to modify the resolution or directing range of the specified setpoint. It is aso included in calculation of the vl velocity demand, and vl velocity actual value. It does not influence the velocity limit function and the ramp function. The value shall have no physical unit and shall be given in the range from -32 768 to +32 767, but the value of 0 shall not be used.

Index	**204Ch**
Name	pv scaling factor
Object code	ARRAY
Data type	INTEGER32
Category	optional

Subindex	**0**
Description	number of entries
Data type	UNSIGNED8
Category	mandatory
Access	R/O
PDO mapping	not possible
Value range	2
Default value	no

Subindex	**1**
Description	pv scaling factor numerator
Category	optinal
Access	R/W
PDO mapping	not possible
Value range	INTEGER32
Default value	+1

Subindex	**2**
Description	pv scaling factor denominator
Category	optional
Access	R/W
PDO mapping	not possible
Value range	INTEGER32
Default value	+1

6.5.3 Objects for position calculation

6.5.3.1 Object 608Fh: Position encoder resolution (DS402)

The position encoder resolution defines the ratio of encoder increments per motor revolution.

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

Subindex	0
Name	number of entries
Data type	UNSIGNED8
Category	mandatory
Access	R/O
PDO Mapping	not possible
Value Range	2
Default Value	2

Subindex	1
Name	Encoder increments
Category	mandatory
Access	R/W
PDO Mapping	not possible
Value Range	UNSIGNED 32
Default Value	2^20

Subindex	2
Name	Motor revolutions
Category	mandatory
Access	R/W
PDO Mapping	not possible
Value Range	UNSIGNED 32
Default Value	1

6.5.3.2 Object 6091h: Gear Ratio (DS402)

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

gear ratio = motor shaft revolutions / driving shaft revolutions

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

Subindex	**0**
Name	number of entries
Data type	UNSIGNED8
Category	mandatory
Access	R/O
PDO Mapping	not possible
Value Range	2
Default Value	2

Subindex	**1**
Name	Motor revolution
Category	mandatory
Access	R/W
PDO Mapping	not possible
Value Range	UNSIGNED 32
Default Value	1

Subindex	**2**
Name	Shaft revolutions
Category	mandatory
Access	R/W
PDO Mapping	not possible
Value Range	UNSIGNED 32
Default Value	1

6.5.3.3 Object 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.

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

Subindex	**0**
Name	number of entries
Data type	UNSIGNED8
Category	mandatory
Access	R/O
PDO Mapping	not possible
Value Range	2
Default Value	2

Subindex	**1**
Name	Feed
Category	mandatory
Access	R/W
PDO Mapping	not possible
Value Range	UNSIGNED 32
Default Value	1

Subindex	**2**
Name	Shaft revolutions
Category	mandatory
Access	R/W
PDO Mapping	not possible
Value Range	UNSIGNED 32
Default Value	1

6.6 Profile Velocity Mode (pv) (DS402)

6.6.1 General Information

The profile velocity mode enables the processing of velocity setpoints and the associated accelerations.

6.6.1.1 Objects that are defined in this section

Index	Object	Name	Type	Access
606Ch	VAR	velocity actual value	INTEGER32	R/O
60FFh	VAR	target velocity	INTEGER32	R/W

6.6.1.2 Objects that are defined in other sections

Index	Object	Name	Type	Section
6040h	VAR	control word	INTEGER16	dc (=> p. 88)
6041h	VAR	status word	UNSIGNED16	dc (=> p. 89)
6063h	VAR	position actual value*	INTEGER32	pc (=> p. 101)
6083h	VAR	profile acceleration	UNSIGNED32	pp (=> p. 114)
6084h	VAR	profile deceleration	UNSIGNED32	pp (=> p. 114)

6.6.2 Object description

6.6.2.1 Object 606Ch: velocity actual value (DS402)

The object velocity actual value represents the actual speed.

Index	**606Ch**
Name	velocity actual value, VL.FB
Object code	VAR
Data type	INTEGER32
Mode	pv
Access	R/O
PDO mapping	possible
Unit	velocity units (SDO is in user units and the PDO is in RPM
Value range	(-2^{31}) to $(2^{31}-1)$
Default value	—
EEPROM	no

6.6.2.2 Object 60FFh: target velocity (DS402)

The speed setpoint (target velocity) represents the setpoint for the ramp generator.

Index	**60FFh**
Name	target velocity, VL.CMDU
Object code	VAR
Data type	INTEGER32
Mode	pv
Access	R/W
PDO mapping	possible
Unit	increments
Value range	(-2^{31}) to $(2^{31}-1)$
Default value	—
EEPROM	no

6.7 Profile Torque Mode (tq) (DS402)

6.7.1 General Information

The profile torque mode enables the processing of torque setpoints and the associated current.

6.7.1.1 Objects that are defined in this section

Index	Object	Name	Type	Access
2071h	VAR	Target current	INTEGER32	R/W
2077h	VAR	Current actual value	INTEGER32	R/O
6071h	VAR	Target torque	INTEGER16	R/W
6073h	VAR	Max current	UNSIGNED16	R/W
6077h	VAR	Torque actual value	INTEGER16	R/O

6.7.1.2 Objects that are defined in other sections

None.

6.7.2 Object description

6.7.2.1 Object 2071h: Target Current

This parameter can be used alternatively to the DS402 parameter 6071h and is the input to the torque controller. The value is scaled in mA (milli Amperes).

Index	**2071h**
Name	Target current
Object code	VAR
Data type	INTEGER 32
Category	optional
Access	RW
PDO mapping	possible
Value range	depends on DRV.IPEAK and MOTOR.IPEAK
Default value	0

6.7.2.2 Object 2077h: Current Actual Value

This parameter can be used alternatively to the DS402 parameter 6077h. The value is scaled in mA (milli Amperes).

Index	**2077h**
Name	Current actual value
Object code	VAR
Data type	INTEGER 32
Category	optional
Access	RO
PDO mapping	possible
Value range	depends on DRV.IPEAK and MOTOR.IPEAK
Default value	0

6.7.2.3 Object 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 (1‰) of rated torque.

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

6.7.2.4 Object 6073h: Max current (DS402)

This value represents the maximum permissible torque creating current in the motor and is given per thousand (1‰) of rated current.

Index	**6073h**
Name	Max current
Object code	VAR
Data type	UNSIGNED16
Category	optional
Access	R/W
PDO mapping	not possible
Value range	UNSIGNED16
Default value	0

6.7.2.5 Object 6077h: Torque actual value (DS402)

The torque actual value corresponds to the instantaneous torque in the drive motor. The value is given per thousand (1‰) of rated torque.

Index	**6077h**
Name	Torque actual value
Object code	VAR
Data type	INTEGER16
Category	optional
Access	R/O
PDO mapping	possible
Value range	INTEGER16
Default value	0

6.8 Position Control Function (pc) (DS402)

6.8.1 General Information

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.

6.8.1.1 Objects that are defined in this section

Index	Object	Name	Type	Access
6063h	VAR	position actual value*	INTEGER32	r
6064h	VAR	position actual value	INTEGER32	r
6065h	VAR	following error window	UNSIGNED32	R/W

6.8.1.2 Objects that are defined in other sections

Index	Object	Name	Type	Section
607Ah	VAR	target position	INTEGER32	pp (=> p. 113)
607Ch	VAR	home-offset	INTEGER32	hm (=> p. 108)
607Dh	ARRAY	software position limit	INTEGER32	pp (=> p. 113)
6040h	VAR	control word	INTEGER16	dc (=> p. 88)
6041h	VAR	status word	UNSIGNED16	dc (=> p. 89)

6.8.2 Object Description

6.8.2.1 Object 6063h: position actual value* (DS402)

The object position actual value provides the momentary actual position in increments.
The resolution is defined with Object 608F as power-of-two number.

Index	**6063h**
Name	position actual value
Object code	VAR
Data type	INTEGER32
Mode	pc, pp
Access	R/W
PDO mapping	possible
Unit	increments (1 turn = 2^{PRBASE})
Value range	(-2^{31}) to (2^{31}-1)
Default value	2^{20}
EEPROM	no

6.8.2.2 Object 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 (Object 6091/6092).

Index	**6064h**
Name	position actual value, PL.FB
Object code	VAR
Data type	INTEGER32
Mode	pc, pp
Access	R/W
PDO mapping	possible
Unit	position units
Value range	(-2^{31}) to $(2^{31}-1)$
Default value	—
EEPROM	no

6.8.2.3 Object 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	**6065h**
Name	Following error window
Object code	VAR
Data type	UNSIGNED32
Category	optional
Access	R/W
PDO mapping	not possible
Value range	UNSIGNED32
Default value	0

6.8.2.4 Object 60F4h: Following error actual value (DS402)

This object returns the current value of the following error in units defined by the user.

Index	**60F4h**
Name	Following error actual value
Object code	VAR
Data type	Integer32
Category	optional
Access	R/O
PDO mapping	possible
Value range	INTEGER32
Default value	0

6.9 Interpolated Position Mode (ip) (DS402)

6.9.1 General information

The interpolated position mode is implemented in a simple, straightforward way. Single position setpoints must be transmitted in the interpolation time period and are taken over on every defined SYNC - telegram sent. A linear interpolation is used between the setpoints. Examples can be found from page => p. 151.

6.9.1.1 Objects defined in this section

Index	Object	Name	Type	Access
60C0h	VAR	Interpolation sub mode select	INTEGER16	R/W
60C1h	ARRAY	Interpolation data record	INTEGER32	R/W
60C2h	RECORD	Interpolation time period	Interpolation time period	R/W
60C4h	RECORD	Interpolation data configuration record	Interpolation data configuration record	R/W

6.9.1.2 Objects defined in other sections

6.9.2 Object description

6.9.2.1 Object 60C0h: Interpolation sub mode select

In the AKD, both linear interpolation between position setpoints and Cubic polynomial interpolation between position/velocity/time setpoints are supported.

Index	**60C0h**
Name	Interpolation sub mode select
Object code	VAR
Data type	INTEGER16
Category	optional

Access	R/W
PDO mapping	not possible
Value range	-2, -1, 0
Default value	0

Value description

Value(decimal)	Description
0	Linear interpolation with a constant time.
-1	Reserved.
-2	Cubic polynomial interpolation, which is also known as position, velocity, and time (PVT) interpolation.

6.9.2.2 Object 60C1h: Interpolation data record

In the AKD, a single setpoint (target position, Subindex 1) is supported for the linear interpolation. For cubic polynomial interpolation 3 setpoints is supported, target position, time and velocity (Subindex1 to Subindex3). After the last item of an interpolation data record is written to the devices input buffer, the pointer of the buffer is automatically incremented to the next buffer.

Index	**60C1h**
Name	Interpolation data record
Object code	ARRAY
Data type	INTEGER32
Category	optional

Subindex	**0**
Description	number of entries
Data type	UNSIGNED8
Category	mandatory
Access	R/O
PDO mapping	not possible
Value range	3
Default value	3

Subindex	**1**
Description	Interpolation target position in counts, the first parameter of interpolation function
Category	mandatory
Access	R/W
PDO mapping	possible
Value range	INTEGER32
Default value	no

Subindex	**2**
Description	Interpolation time in ms, the second parameter of interpolation function. It means the interval time until move to target PVT Point n+1. Set to zero to end the move.
Category	mandatory
Access	R/W
PDO mapping	possible
Value range	UNSIGNED32
Default value	no

Subindex	**3**
Description	Interpolation target velocity in counts/s, the third parameter of interpolation function.
Category	mandatory
Access	R/W
PDO mapping	possible
Value range	INTEGER32
Default value	no

6.9.2.3 Object 60C2h: Interpolation time period

The interpolation time period is used for the PLL (phase locked loop) synchronized position modes. The unit (subindex 1) of the time is given in $10^{\text{interpolation time index}}$ seconds.
Only multiples of 1 ms are allowed. The two values define the internal ASCII - parameter PTBASE (given in multiples of 250 Mikroseconds). Both values must be written to fix a new interpolation time period. PTBASE will only be updated then.

Index	**60C2h**
Name	Interpolation time period
Object code	RECORD
Data type	Interpolation time period record (0080h)
Category	optional

Subindex	**0**
Description	number of entries, FBUS.SAMPLEPERIOD
Category	mandatory
Access	R/O
PDO mapping	not possible
Value range	2
Default value	2

Subindex	**1**
Description	Interpolation time units
Category	mandatory
Access	R/W
PDO mapping	not possible
Value range	UNSIGNED8
Default value	1
Subindex	**2**
Description	Interpolation time index
Category	mandatory
Access	R/W
PDO mapping	not possible
Value range	INTEGER16
Default value	1

6.9.2.4 Object 60C4h: Interpolation data configuration

In the AKD, for linear interpolation, only the value 1 in Subindex 5 is possible. For cubic interpolation, Subindex 1 and Subindex 2 is possible for manage the buffer of cubic interpolation.

Index	**60C4h**
Name	Interpolation data configuration
Object code	RECORD
Data type	Interpolation data configuration record (0081h)
Category	optional

Subindex	**0**
Description	number of entries
Category	mandatory
Access	R/O
PDO mapping	not possible
Value range	6
Default value	6

Subindex	**1**
Description	Maximum buffer size
Category	mandatory
Access	R/O
PDO mapping	not possible
Value range	UNSIGNED32
Default value	10

Subindex	**2**
Description	Actual buffer size
Category	mandatory
Access	R/O
PDO mapping	possible
Value range	0 to 9
Default value	9

Subindex	**3**
Description	Buffer organization
Category	mandatory
Access	R/W
PDO mapping	not possible
Value range	UNSIGND8
Default value	0

Subindex	4
Description	Buffer position
Category	mandatory
Access	R/W
PDO mapping	not possible
Value range	UNSIGNED16
Default value	0

Subindex	5
Description	Size of data record
Category	mandatory
Access	W
PDO mapping	not possible
Value range	1 to 254
Default value	1

Subindex	6
Description	Buffer clear
Category	mandatory
Access	W
PDO mapping	not possible
Value range	UNSIGNED8
Default value	0

6.10 Homing Mode (hm) (DS402)

6.10.1 General information

This section describes the various parameters which are required to define a homing mode.

6.10.1.1 Objects that are defined in this section

Index	Object	Name	Type	Access
607Ch	VAR	HOME.P: home offset	INTEGER32	R/W
6098h	VAR	HOME.MODE, HOME.DIR:homing method	INTEGER8	R/W
6099h	ARRAY	HOME.V: homing speeds	UNSIGNED32	R/W
609Ah	VAR	HOME.ACC, HOME.DEC: homing accel./decel.	UNSIGNED32	R/W

6.10.1.2 Objects that are defined in other sections

Index	Object	Name	Type	Section
6040h	VAR	control word	INTEGER16	dc (=> p. 88)
6041h	VAR	status word	UNSIGNED16	dc (=> p. 89)

6.10.2 Object Description

6.10.2.1 Object 607Ch: homing offset (DS402)

The reference offset (home offset) is the difference between the zero position for the application and the zero point of the machine. All subsequent absolute motion tasks take account of the reference offset.

Index	**607Ch**
Name	home offset, HOME.P
Object code	VAR
Data type	INTEGER32
Mode	hm
Access	R/W
PDO mapping	not possible
Unit	user-defined
Value range	(-2^{31}) to $(2^{31}-1)$
Default value	0

6.10.2.2 Object 6098h: homing method (DS402)

Index	**6098h**
Name	homing method, HOME.MODE, HOME.DIR
Object code	VAR
Data type	INTEGER8
Mode	hm
Access	R/W
PDO mapping	not possible
Unit	position units
Value range	-128 to 127
Default value	0

Description of the homing methods

Choosing a homing method by writing a value to homing method (Object 6098h) will clearly establish:

- the homing signal (P-Stop, N-Stop, reference switch)
- the direction of actuation

and where appropriate

- the position of the index pulse.

The reference position is give by the reference offset (Object 607Ch).

A detailed description of the types of homing movement can be found in the description of WorkBench.

The following homing methods are supported:

Method as per DS402	Brief description: Homing	command
-128 to -1	reserved	—
0	reserved	—
1	homing to negative limit switch, with zeroing, negative count direction	HOME.MODE=2, HOME.DIR=0
2	homing to positive limit switch, with zeroing, positive count direction	HOME.MODE=2, HOME.DIR=1
3 to 7	not supported	—
8	homing to reference switch, with zeroing, positive count direction	HOME.MODE=5, HOME.DIR=1
9 to 11	not supported	—
12	homing to reference switch, with zeroing, negative count direction	HOME.MODE=5, HOME.DIR=0
13 to 14	not supported	—
15 to 16	reserved	—
17	homing to negative limit switch, without zeroing, negative count direction	HOME.MODE=1, HOME.DIR=0
18	homing to negative limit switch, without zeroing, positive count direction	HOME.MODE=1, HOME.DIR=1
19 to 23	not supported	—
24	homing to reference switch, without zeroing, positive count direction	HOME.MODE=4, HOME.DIR=1
25 to 27	not supported	—
28	homing to reference switch, without zeroing, negative count direction	HOME.MODE=4, HOME.DIR=0
29 to 30	not supported	—
31 to 32	reserved	—
33	homing within a single turn, negative count direction	HOME.MODE=7, HOME.DIR=0
34	homing within a single turn, positive count direction	HOME.MODE=7, HOME.DIR=1
35	set reference point at present position	HOME.MODE=0, HOME.DIR=0
36 to 127	reserved	—

6.10.2.3 Object 6099h: homing speeds (DS402)

Index	6099h
Name	homing speeds
Object code	ARRAY
Data type	UNSIGNED32

Subindex	1
Description	speed during search for switch, HOME.V
Mode	hm
Access	R/W
PDO mapping	not possible
Unit	velocity units
Value range	0 to (2^{32}-1)
Default value	equivalent 60 rpm

Subindex	2
Description	speed during search for zero, HOME.FEEDRATE
Mode	hm
Access	R/W
PDO mapping	not possible
Unit	velocity units
Value range	0 to (2^{32}-1)
Default value	1/8 * Object 6099 sub 1

6.10.2.4 Object 609Ah: homing acceleration (DS402)

Index	609Ah
Name	homing acceleration
Object code	VAR
Data type	UNSIGNED32
Mode	hm

Access	R/W
PDO mapping	not possible
Unit	acceleration units
Value range	0 to (2^{32}-1)
Default value	0

6.10.2.5 Homing Mode Sequence

The homing movement is started by setting Bit 4 (positive edge). The successful conclusion is indicated by Bit 12 in the status word (" Object 6041h: Status word (DS402)" (=> p. 89)). Bit 13 indicates that an error occurred during the homing movement. In this case, the error code must be evaluated (error register: " Object 1001h: Error register (DS301)" (=> p. 47)," Object 1003h: Predefined Error Field (DS301)" (=> p. 49), manufacturer status: " Object 1002h: Manufacturer Status Register (DS301)" (=> p. 48)).

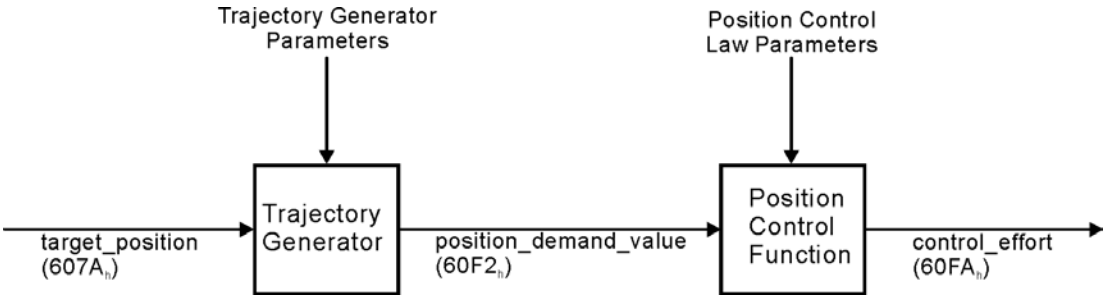
Bit 4	Meaning
0	homing inactive
0 => 1	start homing movement
1	homing active
1 => 0	interruption of homing movement

Bit 13	Bit 12	Meaning
0	0	reference point not set, or homing movement not yet finished
0	1	reference point set, homing movement finished
1	0	homing movement could not be successfully concluded (lag error)
1	1	impermissible state

6.11 Profile Position Mode (DS402)

6.11.1 General Information

The overall structure for this mode is shown in this figure:



The special handshake procedure for the control word and status word is described in "Functional Description" (=> p. 115)

6.11.1.1 Objects that are defined in this section

Index	Object	Name	Type	Access
607Ah	VAR	target position	INTEGER32	R/W
607Dh	ARRAY	software position limit	INTEGER32	R/W
6081h	VAR	profile velocity	UNSIGNED32	R/W
6083h	VAR	profile acceleration	UNSIGNED32	R/W
6084h	VAR	profile deceleration	UNSIGNED32	R/W

6.11.1.2 Objects that are defined in other sections

Index	Object	Name	Type	Section
6040h	VAR	control word	INTEGER16	dc (=> p. 88)
6041h	VAR	status word	UNSIGNED16	dc (=> p. 89)

6.11.2 Object Description

6.11.2.1 Object 607Ah: target position (DS402)

The object target position defines the target position for the drive. The target position is interpreted as a relative distance or an absolute position, depending on Bit 6 of the control word. The type of relative movement can be further defined by the manufacturer-specific parameter 35B9h Subindex 0 and 1.
The mechanical resolution is set by the gearing factors Object 6093h Subindex 1 and 2.

Index	**607Ah**
Name	target position, MT.P
Object code	VAR
Data type	INTEGER32
Mode	pp
Access	R/W
PDO mapping	possible
Unit	user-defined
Value range	$-(2^{31}-1)$ to $(2^{31}-1)$
Default value	—

6.11.2.2 Object 607Dh: Software position limit (DS402)

Software position limit contains the sub-parameters min position limit and max position limit. New target positions are checked against these limits. The limits are relative to the machine home position, which is the result of homing (including the home offset (Object 607Ch)). As default the software position limits are switched off. Changed values must be saved and the drive must be restarted to take enable the new the software limits.

Index	**607Dh**
Name	Software position limit, SWLS.LIMIT0
Object code	ARRAY
Data type	INTEGER32
Category	optional
Subindex	**0**
Description	number of entries
Category	mandatory
Access	R/O
PDO mapping	not possible
Value range	2
Default value	2
Subindex	**1**
Description	min position limit, SWLS.LIMIT0
Category	mandatory
Access	R/W
PDO mapping	not possible
Value range	INTEGER32
Default value	0 (switched off)

6.11.2.3 Object 6081h: profile velocity (DS402)

The profile velocity is the final velocity that should be reached after the acceleration phase of a motion task.

Index	**6081h**
Name	profile velocity, MT.V
Object code	VAR
Data type	UNSIGNED32
Mode	pp
Access	R/W
PDO mapping	possible
Unit	speed units
Value range	0 to (2^{32}-1)
Default value	10

6.11.2.4 Object 6083h: profile acceleration (DS402)

The acceleration ramp (profile acceleration) is given in units that are defined by the user (position units per s²). They can be transformed with the acceleration factor defined by Object 6097 sub1 & 2. The type of acceleration ramp can be selected as a linear ramp or a sin² ramp (see Object 6086h).

Index	**6083h**
Name	profile acceleration, MT.ACC
Object code	VAR
Data type	UNSIGNED32
Mode	pp
Access	R/W
PDO mapping	possible
Unit	acceleration units
Value range	0 to (2^{32}-1)
Default value	0

6.11.2.5 Object 6084h: profile deceleration (DS402)

The braking/deceleration ramp is handled in the same way as the acceleration ramp (" Object 6083h: profile acceleration (DS402)" (=> p. 114)).

Index	**6084h**
Name	profile deceleration, MT.DEC
Object code	VAR
Data type	UNSIGNED32
Mode	pp
Access	R/W
PDO mapping	possible
Unit	deceleration units
Value range	0 to (2^{32}-1)
Default value	0

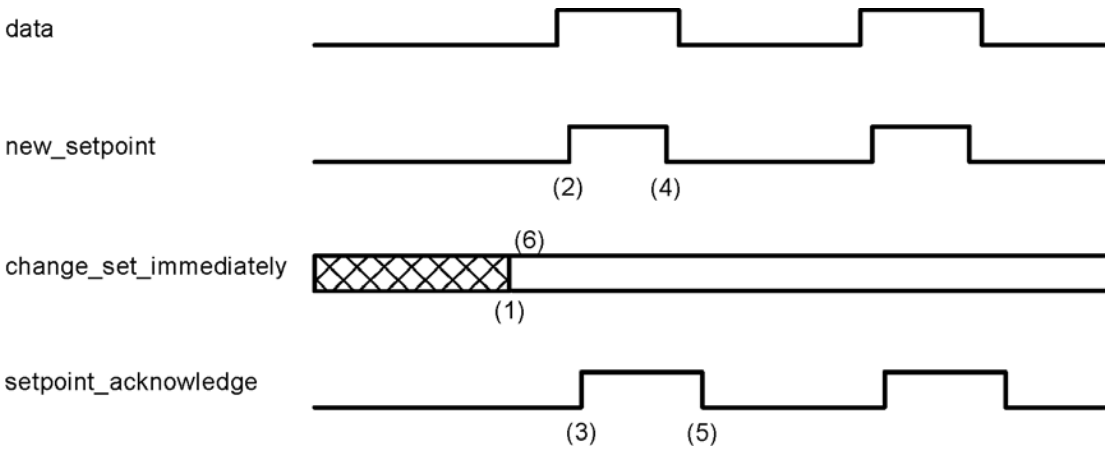
6.11.2.6 Functional Description

Two different ways to apply target positions to a drive are supported by this device profile.

Set of setpoints:
After reaching the target_position, the drive device immediately processes the next target position, which results in a move where the velocity of the drive normally is not reduced to zero after achieving a setpoint. With AKD, this is only possible if trapezoidal ramps are used.

Single setpoints:
After reaching the target_position, the drive device signals this status to a host computer and then receives a new setpoint. After reaching a target_position, the velocity is normally reduced to zero before starting a move to the next setpoint.

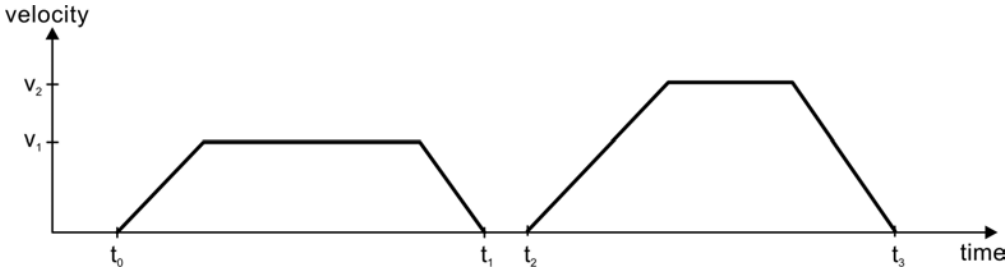
The two modes are controlled by the timing of the bits for new_setpoint and change_set_immediately in the control word, and setpoint_acknowledge in the status word. These bits allow the setting up of a request-response mechanism in order to prepare a set of setpoints while another set is still being processed in the drive unit. This minimizes reaction times within a control program on a host computer.



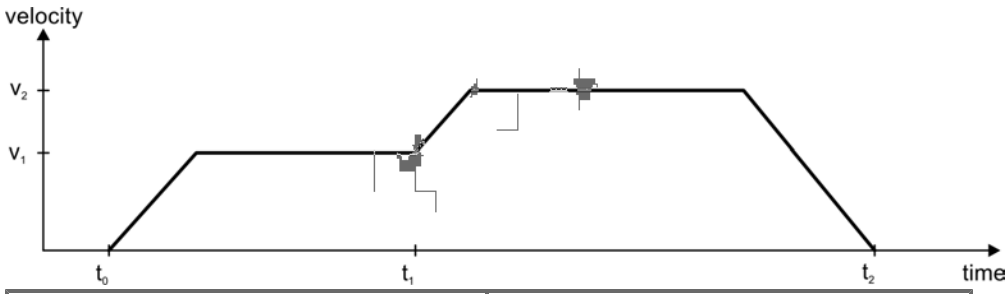
The figures show the difference between the set_of_setpoints mode and the single setpoint mode. The initial status of the bit change_set_immediately in the control word determines which mode is used. To keep these examples simple, only trapezoidal moves are used.

If the bit change_set_immediately is "0" a single setpoint is expected by the drive (1). After data is applied to the drive, a host signals that the data is valid by changing the bit new_setpoint to "1" in the control word (2). The drive responds with setpoint_acknowledge set to "1" in the status word (3) after it has recognized and buffered the new valid data. Now the host can release new_setpoint (4) and subsequently the drive will signal through setpoint_ acknowledge = "0" its ability to accept new data again (5).

In the figure below this mechanism results in a velocity of zero after ramping down to reach a target_position X1 at t1. After signaling to the host, that the setpoint has been reached as described above, the next target_position is processed at t2 and reached at t3.



With change_set_immediately set to "1" (6), the host instructs the drive to apply a new setpoint immediately after reaching the previous one. The relative timing of the other signals is unchanged. This behavior causes the drive to process the next setpoint X2 in advance, and to hold its velocity when it reaches the target_position X1 at t1. The drive then moves immediately to the next target_position X2 that has already been calculated.



Bits in the control word:		Bits in the status word:	
Bit 4	new_setpoint (positive edge!)	Bit 12	setpoint acknowledge
Bit 5	change_set_immediately	Bit 13	lag/following error
Bit 6	absolute/relative		

Notes on motion task type relative:
If Bit 6 is set, then the motion task type is relative, and activated according to the last target position or actual position. If other types of relative motion are required, these must be activated in advance through the manufacture specific object 35B9h Subindex 0 (MT.CNTL).

Notes on profile position mode:
Functional description for the profile position mode
The drive profile DS402 distinguishes between two methods of moving to a target position. These two methods are controlled by the bits for new_setpoint and change_set_immediately in the control word, and setpoint_ acknowledge in the status word. These bits can be used to prepare a motion task while another is still being carried out (handshake).

Moving to several target positions without an intermediate halt
After the target position has been reached, the drive moves immediately to the next target position. This requires that new setpoints are signaled to the drive. This is done through a positive transition of the new_setpoint bit. In this case, the setpoint_acknowledge bit must not be active (=1) in the status word (see also Handshake DS402). The velocity is not reduced to zero when the first setpoint is reached.

Moving to a single target position
The drive moves to the target position, whereby the velocity is reduced to zero. Reaching the target position is signaled by the bit for target_reached in the status word.

7 Appendix

7.1 Object Dictionary

The following table describes all objects reachable via SDO or PDO. (i.p. = in preparation).

Abbreviations:

U	= UNSIGNED	RO	= Read only
INT	= INTEGER	RW	= Read and Write
VisStr	= Visible String	WO	= Write only
		const	= Constant

7.1.1 Float Scaling

The scaling applied to objects which match floating-point parameters in WorkBench/Telnet are listed in the column "Float Scaling."

For example, index 607Ah is listed as 1:1 - this means that commanding a value of 1000 in SDO 607Ah is equivalent to entering MT.P 1000.000 in Workbench. On the other hand, index 3598h is listed as 1000:1 - this means that commanding a value of 1000 in SDO 3598h is equivalent to entering IL.KP 1.000 in Workbench.

A few parameters are listed as variable (var), because the scaling depends on other settings.

7.1.2 Communication SDOs

Index	Sub-index	Data Type	Float Scaling	Access	PDO mapp.	Description	ASCII object
1000h	0	U32		RO	no	Device type	—
1001h	0	U8		RO	no	Error register	—
1002h	0	U32		RO	yes	Manufacturer-specific status register	—
1003h		ARRAY				Pre-defined error field	—
1003h	0	U8		RW	no	Number of errors	—
1003h	1 to 10	U32		RO	no	standard error field	—
1005h	0	U32		RW	no	COB—ID SYNC message	—
1006h	0	U32		RW	no	Communication cycle period	—
1008h	0	VisStr		const	no	Manufacturer device name	—
1009h	0	VisStr		const	no	Manufacturer hardware version	—
100Ah	0	VisStr		const	no	Manufacturer software version	—
100Ch	0	U16		RW	no	Guard time	—
100Dh	0	U8		RW	no	Lifetime factor	—
1010h		ARRAY				Save parameters	—
1010h	0	U8		RO	no	Number of entries	—
1010h	1	U32		RW	no	Saves the drive parameters from the RAM to the NV.	DRV.NVSAVE
1011h		ARRAY				Load parameters	—
1011h	0	U8		RO	no	Number of entries	—
1011h	1	U32		RW	no	Restores the drive parameters from the NV to the RAM.	DRV.NVLOAD
1014h	0	U32		RW	no	COB—ID for the Emergency Object	—

Index	Sub-index	Data Type	Float Scaling	Access	PDO mapp.	Description	ASCII object
1016h		RECORD				Consumer heartbeat time	
1016h	0	U8		RO	no	Number of entries	—
1016h	1	U32		RW	no	Consumer heartbeat time	—
1017h	0	U16		RW	no	Producer heartbeat time	—
1018h		RECORD				Identity Object	—
1018h	0	U8		RO	no	Number of entries	—
1018h	1	U32		RO	no	Vendor ID	—
1018h	2	U32		RO	no	Product Code	—
1018h	3	U32		RO	no	Revision number	—
1018h	4	U32		RO	no	Serial number	
1026h		ARRAY				OS prompt	—
1026h	0	U8		RO	no	Number of entries	—
1026h	1	U8		WO	no	StdIn	—
1026h	2	U8		RO	no	StdOut	—
1400h		RECORD				RXPDO1 communication parameter	—
1400h	0	U8		RO	no	Number of entries	—
1400h	1	U32		RW	no	RXPDO1 COB — ID	—
1400h	2	U8		RW	no	Transmission type RXPDO1	—
1401h		RECORD				RXPDO2 communication parameter	—
1401h	0	U8		RO	no	Number of entries	—
1401h	1	U32		RW	no	RXPDO2 COB—ID	—
1401h	2	U8		RW	no	Transmission type RXPDO2	—
1402h		RECORD				RXPDO3 communication parameter	—
1402h	0	U8		RO	no	Number of entries	—
1402h	1	U32		RW	no	RXPDO3 COB—ID	—
1402h	2	U8		RW	no	Transmission type RXPDO3	—
1403h		RECORD				RXPDO4 communication parameter	—
1403h	0	U8		RO	no	Number of entries	—
1403h	1	U32		RW	no	RXPDO4 COB—ID	—
1403h	2	U8		RW	no	Transmission type RXPDO4	—
1600h		RECORD				RXPDO1 mapping parameter	—
1600h	0	U8		RO	no	Number of entries	—
1600h	1 to 8	U32		RW	no	Mapping for n—th application object	—
1601h		RECORD				RXPDO2 mapping parameter	—
1601h	0	U8		RO	no	Number of entries	—
1601h	1 to 8	U32		RW	no	Mapping for n—th application object	—
1602h		RECORD				RXPDO3 mapping parameter	—
1602h	0	U8		RO	no	Number of entries	—
1602h	1 to 8	U32		RW	no	Mapping for n—th application object	—
1603h		RECORD				RXPDO4 mapping parameter	—
1603h	0	U8		RO	no	Number of entries	—
1603h	1 to 8	U32		RW	no	Mapping for n—th application object	—
1800h		RECORD				TXPDO1 communication parameter	—
1800h	0	U8		RO	no	Number of entries	—

Index	Sub-index	Data Type	Float Scaling	Access	PDO mapp.	Description	ASCII object
1800h	1	U32		RW	no	TXPDO1 COB—ID	—
1800h	2	U8		RW	no	Transmission type TXPDO1	—
1800h	3	U16		RW	no	Inhibit time	—
1800h	4	U8		const	no	reserved	—
1800h	5	U16		RW	no	Event timer	—
1801h		RECORD				TXPDO2 communication parameter	—
1801h	0	U8		RO	no	Number of entries	—
1801h	1	U32		RW	no	TXPDO2 COB—ID	—
1801h	2	U8		RW	no	Transmission type TXPDO2	—
1801h	3	U16		RW	no	Inhibit time	—
1801h	4	U8		const	no	reserved	—
1801h	5	U16		RW	no	Event timer	—
1802h		RECORD				TXPDO3 communication parameter	—
1802h	0	U8		RO	no	Number of entries	—
1802h	1	U32		RW	no	TXPDO3 COB—ID	—
1802h	2	U8		RW	no	Transmission type TXPDO3	—
1802h	3	U16		RW	no	Inhibit time	—
1802h	4	U8		const	no	reserved	—
1802h	5	U16		RW	no	Event timer	—
1803h		RECORD				TXPDO4 communication parameter	—
1803h	0	U8		RO	no	Number of entries	—
1803h	1	U32		RW	no	TXPDO4 COB—ID	—
1803h	2	U8		RW	no	Transmission type TXPDO4	—
1803h	3	U16		RW	no	Inhibit time	—
1803h	4	U8		const	no	reserved	—
1803h	5	U16		RW	no	Event timer	—
1A00h		RECORD				Mapping parameter TXPDO1	—
1A00h	0	U8		RO	no	Number of entries	—
1A00h	1 to 8	U32		RW	no	Mapping for n—th application object	—
1A01h		RECORD				Mapping parameter TXPDO2	—
1A01h	0	U8		RO	no	Number of entries	—
1A01h	1 to 8	U32		RW	no	Mapping for n—th application object	—
1A02h		RECORD				Mapping parameter TXPDO3	—
1A02h	0	U8		RO	no	Number of entries	—
1A02h	1 to 8	U32		RW	no	Mapping for n—th application object	—
1A03h		RECORD				Mapping parameter TXPDO4	—
1A03h	0	U8		RO	no	Number of entries	—
1A03h	1 to 8	U32		RW	no	Mapping for n—the application object	—

7.1.3 Manufacturer specific SDOs

Index	Sub-index	Data Type	Float Scaling	Access	PDO mapp.	Description	ASCII object
2014h		ARRAY				Mask TxPDO Channel 1	—
2014h	1	U32		RW	no	Mask (Byte 0..3)	—
2014h	2	U32		RW	no	Mask (Byte 4..7)	—
2015h		ARRAY				Mask TxPDO Channel 2	—
2015h	1	U32		RW	no	Mask (Byte 0..3)	—
2015h	2	U32		RW	no	Mask (Byte 4..7)	—
2016h		ARRAY				Mask TxPDO Channel 3	—
2016h	1	U32		RW	no	Mask (Byte 0..3)	—
2016h	2	U32		RW	no	Mask (Byte 4..7)	—
2017h		ARRAY				Mask TxPDO Channel 4	—
2017h	1	U32		RW	no	Mask (Byte 0..3)	—
2017h	2	U32		RW	no	Mask (Byte 4..7)	—
2018h		ARRAY				Firmware version	—
2018h	0	U16		const	no	Number of entries	—
2018h	1	U16		const	no	Major version	—
2018h	2	U16		const	no	Minor version	—
2018h	3	U16		const	no	Revision	—
2018h	4	U16		const	no	Branch version	—
204Ch		ARRAY				pv scaling factor	—
204Ch	0	U8		RO	no	Number of entries	—
204Ch	1	INT32		RW	no	pv scaling factor numerator	—
204Ch	2	INT32		RW	no	pv scaling factor denominator	—
2050h	0	INT32	1:1	RO	no	Position, secondary feedback	DRV.HANDWHEEL
2071h	0	INT32		RW	yes	Target current	-
2077h	0	INT32		RO	yes	Current actual value	-
20A0h	0	INT32	var	RO	yes	Latch position 1, positive edge	CAP0.PLFB, CAP0.T
20A1h	0	INT32	var	RO	yes	Latch position 1, negative edge	CAP0.PLFB, CAP0.T
20A2h	0	INT32	var	RO	yes	Latch position 2, positive edge	CAP1.PLFB, CAP1.T
20A3h	0	INT32	var	RO	yes	Latch position 2, negative edge	CAP1.PLFB, CAP1.T
20A4h	0	U16		RW	yes	Latch control register	—
20A5h	0	U16		RW	yes	Latch status register	—
20A6h	0	INT32	var	RO	yes	Sets captured position value	CAP0.PLFB
20B8h	0	U16		RW	yes	Clear changed digital input information	—
3405h		ARRAY				VL.ARTYPE	—
3405h	0	U8		RO	no	Number of entries	—

Index	Sub-index	Data Type	Float Scaling	Access	PDO mapp.	Description	ASCII object
3405h	1	U8		RW	no	Calculation method for BiQuad filter 1	VL.ARTYPE1
3405h	2	U8		RW	no	Calculation method for BiQuad filter 2	VL.ARTYPE2
3405h	3	U8		RW	no	Calculation method for BiQuad filter 3	VL.ARTYPE3
3405h	4	U8		RW	no	Calculation method for BiQuad filter 4	VL.ARTYPE4
3406h		ARRAY				VL BiQuad	—
3406h	0	U8		RO	no	Number of entries	—
3406h	1	U32	1000:1	RW	no	Natural frequency of pole of anti-resonance (AR) filter 1	VL.ARPF1
3406h	2	U32	1000:1	RW	no	Natural frequency of pole of anti-resonance (AR) filter 2	VL.ARPF2
3406h	3	U32	1000:1	RW	no	Natural frequency of pole of anti-resonance (AR) filter 3	VL.ARPF3
3406h	4	U32	1000:1	RW	no	Natural frequency of pole of anti-resonance (AR) filter 4	VL.ARPF4
3406h	5	U32	1000:1	RW	no	Q of pole of anti-resonance (AR) filter 1	VL.ARPQ1
3406h	6	U32	1000:1	RW	no	Q of pole of anti-resonance (AR) filter 2	VL.ARPQ2
3406h	7	U32	1000:1	RW	no	Q of pole of anti-resonance (AR) filter 3	VL.ARPQ3
3406h	8	U32	1000:1	RW	no	Q of pole of anti-resonance (AR) filter 4	VL.ARPQ4
3406h	9	U32	1000:1	RW	no	Natural frequency of zero of anti-resonance (AR)filter 1	VL.ARZF1
3406h	A	U32	1000:1	RW	no	Natural frequency of zero of anti-resonance (AR)filter 2	VL.ARZF2
3406h	B	U32	1000:1	RW	no	Natural frequency of zero of anti-resonance (AR)filter 3	VL.ARZF3
3406h	C	U32	1000:1	RW	no	Natural frequency of zero of anti-resonance (AR)filter 4	VL.ARZF4
3406h	D	U32	1000:1	RW	no	Q of zero of anti-resonance filter 1	VL.ARZQ1
3406h	E	U32	1000:1	RW	no	Q of zero of anti-resonance filter 2	VL.ARZQ2
3406h	F	U32	1000:1	RW	no	Q of zero of anti-resonance filter 3	VL.ARZQ3

Index	Sub-index	Data Type	Float Scaling	Access	PDO mapp.	Description	ASCII object
3406h	10	U32	1000:1	RW	no	Q of zero of anti-resonance filter 4	VL.ARZQ4
3407h		STRUCT				Velocity Filter	—
3407h	0	U8		RO	no	Number of entries	—
3407h	1	INT32	1000:1	RW	no	10 Hz filtered VL.FB	VL.FBFILTER
3407h	2	U32	1000:1	RW	no	Gain for the velocity feedforward	VL.KVFF
3407h	3	U32		RW	no	Gain for the acceleration feedforward	VL.KBUSFF
3407h	4	U32	1:1	RW	no	Sets the velocity error	VL.ERR
3412h	0	INT8		RW	no	Type of regen resistor	REGEN.TYPE
3414h	0	U8		RW		Returns and sets the regen resistor fault level temperature.	REGEN.WATTEXT
3415h	0	U32	1000:1	RO	no	Thermal regen resistor time constant	REGEN.TEXT
3416h	0	U32		RO	no	Gets regen resistor's calculated power	REGEN.POWER
3420h	0	U16	1000:1	RW	no	Sets the foldback fault level.	IL.FOLDFTHRESH
3421h	0	U32	1000:1	RW	no	Sets the user value for the foldback fault level.	IL.FOLDFTHRESHU
3422h	0	U32	1000:1		no	Sets friction compensation value.	IL.FRICTION
3423h	0	INT32	1000:1		no	A constant current command added to compensate for gravity.	IL.OFFSET
3424h	0	U16			no	Enables/disables the integrator part of the PI loop.	IL.INTEN (Password Protected)
3425h	0	U32	1000:1	RO	no	Reads the overall foldback current limit	IL.IFOLD
3426h	0	U32	1000:1	RW	no	Sets current loop acceleration feedforward gain value	IL.KACCFF
3427h		RECORD				Motor protection parameters	—
3427h	0	U8		RO	no	Number of entries	—
3427h	1	U8		RW	no		IL.MIMODE
3427h	2	U8		RW	no		IL.MI2TWTHRESH
3427h	3	U32		RW	yes		IL.MI2T
3430h	0	U8		RW	no	Sets the direction for absolute motion tasks.	PL.MODPDIR
3431h	0	U16		RW	no	Sets the motion task in the drive	MT.SET
3440h		ARRAY				Controlled stop parameters	—

Index	Sub-index	Data Type	Float Scaling	Access	PDO mapp.	Description	ASCII object
3440h	0	U8		RO	no	Number of entries	—
3440h	1	U32	1:1	RW	no	Sets the deceleration value for a controlled stop.	CS.DEC
3440h	2	U32	1:1	RW	no	Sets the velocity threshold for a controlled stop.	CS.VTHRESH
3440h	3	U32		RW	no	Sets the time value for the drive velocity to be within CS.VTHRESH.	CS.TO
3441h	0	U8		RO	no	Controlled stop state	CS.STATE
3443h	0	U16		RO	no	Returns the possible reason for a drive disable	DRV.DIS
3444h	0	U16	1000:1	RO	no	Maximum current for dynamic braking	DRV.DBILIMIT
3445h	0	U32		RO	no	Emergency timeout for braking	DRV.DISTO
3450h	0	U8		WO	no	Release or enable brake	MOTOR.BRAKERLS
3451h	0	U8		RW	no	Determines which drive parameters are calculated automatically.	MOTOR.AUTOSET
3452h	0	U16		RW	no	Sets the motor maximum voltage	MOTOR.VOLTMAX
3453h	0	U32		RW	no	Sets the motor temperature warning level	MOTOR.TEMPWARN
3454h	0	U32	1000:1	RW	no	Sets the thermal constant of the motor coil	MOTOR.CTF0
3455h	0	U32	1000:1	RW	no	Sets the line-to-line motor Lq	MOTOR.LQLL
3456h	0	U32	1000:1	RW	no	Sets the stator winding resistance phase-phase in ohms	MOTOR.R
3457h		RECORD				Induction Motor parameter	—
3457h	0	U8		RO	no	Number of entries	—
3457h	1	INT32	1000:1	RW	no	Configuration of induction motor's rated velocity.	MOTOR.VRATED
3457h	2	U16		RW	no	Configuration of induction motor's rated voltage.	MOTOR.VOLTRATED
3457h	3	U16		RW	no	Sets the minimum voltage for V/f Control.	MOTOR.VOLTMIN
3460h		RECORD				Capture engines parameters	—
3460h	0	U8		RO	no	Number of entries	—
3460h	1	U8		RW	no	Specifies the trigger source for the position capture.	CAP0.TRIGGER

Index	Sub-index	Data Type	Float Scaling	Access	PDO mapp.	Description	ASCII object
3460h	2	U8		RW	no	Specifies the trigger source for the position capture.	CAP1.TRIGGER
3460h	3	U8		RW	no	Selects the captured value.	CAP0.MODE
3460h	4	U8		RW	no	Selects the captured value.	CAP1.MODE
3460h	5	U8		RW	no	Controls the pre-condition logic.	CAP0.EVENT
3460h	6	U8		RW	no	Controls the pre-condition logic.	CAP1.EVENT
3460h	7	U8		RW	no	Selects the capture pre-condition edge.	CAP0.PREEDGE
3460h	8	U8		RW	no	Selects the capture pre-condition edge.	CAP1.PREEDGE
3460h	9	U8		RW	no	Sets the precondition trigger.	CAP0.PRESELECT
3460h	A	U8		RW	no	Sets the precondition trigger.	CAP1.PRESELECT
3470h		RECORD					—
3470h	0	U8		RO	no	Number of entries	—
3470h	1	INT8		RW	no	Sets the analog output mode.	AOUT.MODE
3470h	2	INT16	1000:1	RW	yes	Reads the analog output value.	AOUT.VALUE
3470h	3	INT16	1000:1	RW	yes	Reads and writes the analog output value.	AOUT.VALUEU
3470h	4	INT16	1000:1	RO	yes	Reads the value of the analog input signal.	AIN.VALUE
3470h	5	U32	1000:1	RW	no	Sets velocity scale factor for analog output	AOUT.VSCALE
3471h	0	U32	1:1	RW	no	Sets the analog position scale factor	AOUT.PSCALE
3472h	0	U32	1:1	RW	no	Sets analog pscale factor	AIN.PSCALE
3474h		ARRAY				DINx.PARAM	—
3474h	0	U8		RO	no	Number of entries	—
3474h	1	U32		RW	no	Lower 32-bit part of input parameter 1	DIN1.PARAM
3474h	2	U32		RW	no	Lower 32-bit part of input parameter 2	DIN2.PARAM
3474h	3	U32		RW	no	Lower 32-bit part of input parameter 3	DIN3.PARAM
3474h	4	U32		RW	no	Lower 32-bit part of input parameter 4	DIN4.PARAM
3474h	5	U32		RW	no	Lower 32-bit part of input parameter 5	DIN5.PARAM

Index	Sub-index	Data Type	Float Scaling	Access	PDO mapp.	Description	ASCII object
3474h	6	U32		RW	no	Lower 32-bit part of input parameter 6	DIN6.PARAM
3474h	7	U32		RW	no	Lower 32-bit part of input parameter 7	DIN7.PARAM
3474h	8	U32		RW	no	Higher 32-bit part of input parameter 1	DIN1.PARAM
3474h	9	U32		RW	no	Higher 32-bit part of input parameter 2	DIN2.PARAM
3474h	A	U32		RW	no	Higher 32-bit part of input parameter 3	DIN3.PARAM
3474h	B	U32		RW	no	Higher 32-bit part of input parameter 4	DIN4.PARAM
3474h	C	U32		RW	no	Higher 32-bit part of input parameter 5	DIN5.PARAM
3474h	D	U32		RW	no	Higher 32-bit part of input parameter 6	DIN6.PARAM
3474h	E	U32		RW	no	Higher 32-bit part of input parameter 7	DIN7.PARAM
3475h		ARRAY				DOUTx.PARAM	—
3475h	0	U8		RO	no	Number of entries	—
3475h	1	U32		RW	no	Lower 32-bit part of output parameter 1	DOUT1.PARAM
3475h	2	U32		RW	no	Lower 32-bit part of output parameter 2	DOUT2.PARAM
3475h	3	U32		RW	no	Higher 32-bit part of output parameter 1	DOUT1.PARAM
3475h	4	U32		RW	no	Hogher 32-bit part of output parameter 2	DOUT2.PARAM
3480h	0	U32	1000:1	RW	no	Integral gain of position regulator PID loop	PL.KI
3481h		ARRAY				PL.INTMAX	—
3481h	0	U8		RO	no	Number of entries	—
3481h	1	U32	1:1	RW	no	Input saturation	PL.INTINMAX
3481h	2	U32	1:1	RW	no	Output saturation	PL.INTOUTMAX
3482h	0	INT32	1:1	RO	no	Maximum value of following error in homing	HOME.PERRTHRESH
3483h	0	INT32	1:1	RW	no	Sets the position error warning level	PL.ERRWTHRESH
3484h	0	INT32	1:1	RW	no	Specification of an additional movement after homing is completed.	HOME.DIST
3490h	0	INT32	1:1	RO	no	Position feedback offset	FB1.OFFSET
3491h	0	U32		RO	no	Location of index pulse on EEO	DRV.EMUEMTURN
3492h	0	U32		RO	no	Motion status of the drive	DRV.MOTIONSTAT

Index	Sub-index	Data Type	Float Scaling	Access	PDO mapp.	Description	ASCII object
3493h	0	U8		RO	no	Direction of EEO (emulated encoder output)	DRV.EMUEDIR
3494h		RECORD				WS parameters	—
3494h	0	U8		RO	no	Number of entries	—
3494h	1	INT16	1000:1	RW	no	Sets maximum current used for wake and shake	WS.IMAX
3494h	2	INT32	1:1	RW	no	Sets the maximum movement required for wake and shake	WS.DISTMAX
3494h	3	U16		RW	no	Sets the delay for wake and shake between loops in mode 0	WS.TDELAY3
3494h	4	INT32	1:1	RW	no	Defines the maximum allowed velocity for Wake & Shake	WS.VTHRESH
3494h	5	U8		RO	no	Reads wake and shake status	WS.STATE
3494h	6	U8		RW	no	Arm Wake and Shake to start	WS.ARM
3495h	0	U16	1000:1	RW	no	Voltage level for undervoltage warning.	VBUS.UVWTHRESH
3496h		ARRAY				FBUS synchronization parameters	—
3496h	0	U8		RO	no	Number of entries	—
3496h	1	U32		RW	no	expected time distance between clearing the PLL counter and calling the PLL function	FBUS.SYNCDIST
3496h	2	U32		RW	no	actual time distance between clearing the PLL counter and calling the PLL function	FBUS.SYNCACT
3496h	3	U32		RW	no	Time window, which is used in order to consider the drive as being synchronized	FBUS.SYNCWND
3496h	4	U32		RW	no	Time, which is used for extending or lowering the sample rate of the internal 16[kHz] IRQ	—
34A0h		ARRAY				PLS Position	
34A0h	0	U8		RO	no	Number of entries	—
34A0h	1	INT32	1:1	RW	no	Limit switch 1 compare value	PLS.P1
34A0h	2	INT32	1:1	RW	no	Limit switch 2 compare value	PLS.P2
34A0h	3	INT32	1:1	RW	no	Limit switch 3 compare value	PLS.P3

Index	Sub-index	Data Type	Float Scaling	Access	PDO mapp.	Description	ASCII object
34A0h	4	INT32	1:1	RW	no	Limit switch 4 compare value	PLS.P4
34A0h	5	INT32	1:1	RW	no	Limit switch 5 compare value	PLS.P5
34A0h	6	INT32	1:1	RW	no	Limit switch 6 compare value	PLS.P6
34A0h	7	INT32	1:1	RW	no	Limit switch 7 compare value	PLS.P7
34A0h	8	INT32	1:1	RW	no	Limit switch 8 compare value	PLS.P8
34A1h		ARRAY				PLS Width	—
34A1h	0	U8		RO	no	Number of entries	—
34A1h	1	INT32	1:1	RW	no	Sets Limit Switch1 Width	PLS.WIDTH1
34A1h	2	INT32	1:1	RW	no	Sets Limit Switch 2 Width	PLS.WIDTH2
34A1h	3	INT32	1:1	RW	no	Sets Limit Switch 3 Width	PLS.WIDTH3
34A1h	4	INT32	1:1	RW	no	Sets Limit Switch 4 Width	PLS.WIDTH4
34A1h	5	INT32	1:1	RW	no	Sets Limit Switch 5 Width	PLS.WIDTH5
34A1h	6	INT32	1:1	RW	no	Sets Limit Switch 6 Width	PLS.WIDTH6
34A1h	7	INT32	1:1	RW	no	Sets Limit Switch 7 Width	PLS.WIDTH7
34A1h	8	INT32	1:1	RW	no	Sets Limit Switch 8 Width	PLS.WIDTH8
34A2h		ARRAY				PLS Time	—
34A2h	0	U8		RO	no	Number of entries	—
34A2h	1	U16		RW	no	Sets limit switch 1 time	PLS.T1
34A2h	2	U16		RW	no	Sets limit switch 2 time	PLS.T2
34A2h	3	U16		RW	no	Sets limit switch 3 time	PLS.T3
34A2h	4	U16		RW	no	Sets limit switch 4 time	PLS.T4
34A2h	5	U16		RW	no	Sets limit switch 5 time	PLS.T5
34A2h	6	U16		RW	no	Sets limit switch 6 time	PLS.T6
34A2h	7	U16		RW	no	Sets limit switch 7 time	PLS.T7
34A2h	8	U16		RW	no	Sets limit switch 8 time	PLS.T8
34A3h		ARRAY				PLS Configuration	—
34A3h	0	U8		RO	no	Number of entries	—
34A3h	1	U16		RW	no	Enables the limit switches	PLS.EN
34A3h	2	U16		RW	no	Resets limit switches	PLS.RESET
34A3h	3	U16		RW	no	Selects limit switch mode	PLS.MODE
34A3h	4	U16		RW	no	Reads the limit switch state	PLS.STATE

Index	Sub-index	Data Type	Float Scaling	Access	PDO mapp.	Description	ASCII object
34A4h	0	U8		RW	no	Sets limit switch units	PLS.UNITS
3501h	0	INT32	1:1	RW	no	Acceleration ramp	DRV.ACC
3502h	0	INT32	1:1	RW	no	Acceleration ramp for homing/jog modes	HOME.ACC
3506h	0	INTEGER			no	Action that hardware enable digital input will perform.	DRV.HWENMODE
3509h	0	INT32	1000:1	RO	no	Analog input voltage	AIN.VALUE
3522h	0	INT32	1:1	RW	no	Deceleration rate	DRV.DEC
3524h	0	INT32	1:1	RW	no	Deceleration ramp for homing/jog modes	HOME.DEC
352Ah	0	INT32		RW	no	Direction of movements	DRV.DIR
3533h	0	U32		RO	no	Resolution of motor encoder	FB1.ENCRES
3534h	0	U32		RO	no	Mode of EEO connector	DRV.EMUEMODE
3535h	0	U32		RO	no	Resolution of EEO	DRV.EMUERES
3537h	0	U32		RO	no	Location of EEO index pulse	DRV.EMUEZOFFSET
353Bh	0	INT32		RO	no	Selection of the feedback type	FB1.SELECT
3542h	0	U32	1000:1	RW	no	Position Control Loop: Proportional Gain	PL.KP
3548h	0	U32	1000:1	RW	no	Velocity Control Loop: Proportional Gain	VL.KP
354Bh	0	INT32	1000:1	RW	no	Sets the velocity loop velocity feedforward gain value	VL.KVFF
354Dh	0	INT32	1000:1	RW	no	Velocity Control Loop: I-Integration Time	VL.KI
3558h	0	INT32	1000:1	RO	no	Current Monitor	IL.FB
3559h	0	INT32	1000:1	RO	no	Drive Ifold	IL.DIFOLD
355Ah	0	INT32	1000:1	RW	no	I2T Warning	IL.FOLDWTHRESH
3562h	0	INT32		RW	no	Function of Digital Input 1	DIN1.MODE
3565h	0	INT32		RW	no	Function of Digital Input 2	DIN2.MODE
3568h	0	INT32		RW	no	Function of Digital Input 3	DIN3.MODE
356Bh	0	INT32		RW	no	Function of Digital Input 4	DIN4.MODE
356Eh	0	INT32	1000:1	RW	no	Application Peak Current, positive direction	IL.LIMITP
356Fh	0	INT32	1000:1	RW	no	Application Peak Current, negative direction	IL.LIMITN
3586h	0	U32		RW	no	Sets the motor temperature fault level	MOTOR.TEMPFAULT

Index	Sub-index	Data Type	Float Scaling	Access	PDO mapp.	Description	ASCII object
3587h	0	INT32		RW	no	Select Motor Holding Brake	MOTOR.BRAKE
358Eh	0	U32	1000:1	RW	no	Motor Continuous Current Rating	MOTOR.ICONT
358Fh	0	U32	1000:1	RW	no	Motor Peak Current Rating	MOTOR.IPEAK
3593h	0	U32	1000:1	RW	no	Sets the torque constant of the motor	MOTOR.KT
3596h	0	U32	1000:1	RO	no	Sets the proportional gain of the d-component current PI-regulator as a percentage of IL.KP	IL.KPDRATIO
3598h	0	INT32	1000:1	RW	no	Absolute Gain of Current Control loop	IL.KP
359Ch	0	U32		RW	no	Sets the motor phase.	MOTOR.PHASE
359Dh	0	U32		RW	no	Sets the number of motor poles	MOTOR.POLES
35A3h	0	U32		RW	no	Sets the maximum motor speed	MOTOR.VMAX
35A4h	0	INT32	1000:1	RW	no	Maximum motor current	IL.MIFOLD
35ABh	0	U32	1000:1	RW	no	Sets the motor inertia	MOTOR.INERTIA
35AFh	0	U32		RW	no	Sets the digital output 1 mode	MT.CNTL
35B2h	0	U32		RW	no	Sets the digital output 2 mode	MT.MTNEXT
35B4h	0	INT32		RW	no	Operating Mode	DRV.OPMODE
35BCh	0	INT32		RW	no	Next Task Number for Motion Task 0	MT.MTNEXT
35C2h	0	INT32		RW	no	Select regen resistor	REGEN.REXT
35C5h	0	INT32	1:1	RO	no	Actual Following Error	PL.ERR
35C6h	0	INT32	1:1	RW	no	In-Position Window	MT.TPOSWND
35C7h	0	INT32	1:1	RW	no	Max. Following Error	PL.ERRFTHRESH
35CAh	0	INT32		RW	no	Position Resolution (Numerator)	UNIT.PIN
35CBh	0	INT32		RW	no	Position Resolution (Denominator)	UNIT.POUT
35E2h	0	U32	1:1	RW	no	Sets the current limit during homing procedure to a mechanical stop	HOME.IPEAK
35EBh	0	INT32		WO	no	Save Data in EEPROM	DRV.NVSAVE
35F0h	0	INT32		WO	no	Set Reference Point	HOME.SET
35FEh	0	INT32		WO	no	Stop Motion Task	DRV.STOP
35FFh	0	U32		RW	no	Selects between disable immediately or stop and then disable	DRV.DISMODE
3610h	0	INT32		RO	no	Ambient Temperature	DRV.TEMPERATURES
3611h	0	INT32		RO	no	Heat Sink Temperature	DRV.TEMPERATURES

Index	Sub-index	Data Type	Float Scaling	Access	PDO mapp.	Description	ASCII object
3612h	0	INT32		RO	no	Motor Temperature	MOTOR.TEMP
3617h	0	U32	1:1	RW	no	Undervoltage mode	VBUS.UVMODE
3618h	0	INT32	1:1	RO	no	Actual Velocity	VL.FB
361Ah	0	INT32		RO	no	DC-bus voltage	VBUS.VALUE
361Dh	0	U32	1000:1	RW	no	Voltage level for under-voltage fault	VBUS.UVFTHRESH
3622h	0	INT32	1:1	RW	no	Max. Velocity	VL.LIMITP
3623h	0	INT32	1:1	RW	no	Max. Negative Velocity	VL.LIMITN
3627h	0	INT32	1:1	RW	no	Overspeed	VL.THRESH
3629h	0	INT32	1000:1	RW	no	SW1 Velocity Scaling Factor	AIN.VSCALE
3656h	0	U64	1:1	RW	no	Initial feedback position	FB1.ORIGIN
3659h	0	INT32		RW	no	Type of acceleration set-point for the system	UNIT.ACCROTARY
365Bh	0	INT32		RW	no	Presetting for motion task that is processed later	MT.NUM
365Fh	0	INT32		RW	no	Systemwide Definition of Velocity/Speed	UNIT.VROTARY
3660h	0	INT32		RW	no	Set Resolution of the Position	UNIT.PROTARY
366Eh	0	INT32		RW	no	Disable Delaytime with Holding Brake	MOTOR.TBRAKEAPP
366Fh	0	INT32		RW	no	Enable Delaytime with Holding Brake	MOTOR.TBRAKERLS
3683h	0	U16		RW	no	Delay for wake and shake timing	WS.TDELAY1
3685h	0	U16		RW	no	Sets delay for wake and shake timing	WS.TDELAY2
36D0h	0	U16		RW	no	Sets wake and shake current-vector appliance time	WS.T
36D1h	0	U32	1:1	RW	no	Sets the minimum move-ment required for wake and shake	WS.DISTMIN
36D7h	0	U32	1000:1	RW	no	Sets homing auto move flag	HOME.AUTOMOVE
36E2h	0	U8		RW	no	Sets the number of repe-titions for wake and shake	WS.NUMLOOPS
36E5h	0	U32		RW	no	CAN baud rate selection	FBUS.PARAM01
36E6h	0	U32		RW	no	pll synchronization	FBUS.PARAM02
36E7h	0	U32		RW	no	-	FBUS.PARAM03
36E8h	0	U32		RW	no	SYNC surveillance	FBUS.PARAM04
36E9h	0	U32		RW	no	-	FBUS.PARAM05
36EAh	0	U32		RW	no	-	FBUS.PARAM06
36EBh	0	U32		RW	no	-	FBUS.PARAM07
36ECh	0	U32		RW	no	-	FBUS.PARAM08

Index	Sub-index	Data Type	Float Scaling	Access	PDO mapp.	Description	ASCII object
36EDh	0	U32		RW	no	-	FBUS.PARAM09
36EEh	0	U32		RW	no	-	FBUS.PARAM10
36F6h	0	INT32		RW	no	Function of Digital Input 5	DIN5.MODE
36F9h	0	INT32		RW	no	Function of Digital Input 6	DIN6.MODE
36FCh	0	U32		RW	no	Function of Digital Input 7	DIN7.MODE
3856h	0	INT32	1:1	RW	no	velocity window for profile position mode	MT.TVELWND
5000h	0	UINT32		RW	no	Analog input low-pass filter cutoff frequency.	AIN.CUTOFF
5001h	0	UINT32		RW	no	Analog input signal deadband.	AIN.DEADBAND
5002h	0	UINT32		RW	no	Analog current scale factor.	AIN.ISCALE
5003h	0	UINT32		RW	no	Analog input offset.	AIN.OFFSET
5009h	0	UINT32		RW	no	Analog current scale factor.	AOUT.ISCALE
500Bh	0	UINT32		RW	no	Analog output offset.	AOUT.OFFSET
5013h	0	UINT32		RW	no	Controls how often the excitation is updated.	BODE.EXCITEGAP
5015h	0	UINT32		RW	no	Current command value used during the Bode procedure.	BODE.IAMP
5016h	0	UINT32		RW	no	Sets whether the excitation uses current or velocity excitation type.	BODE.INJECTPOINT
5019h	0	UINT32		RW	no	Length of the PRB signal before it repeats.	BODE.PRBDEPTH
5060h	0	UINT32		RW	no	Sets the fault relay mode.	DOUT.RELAYMODE
5080h	0	UINT32		RW	no	Default state of the software enable.	DRV.ENDEFAULT
5083h	0	UINT32		RW	no	Continuous rated current value.	DRV.ICONT
5084h	0	UINT32		RW	no	Peak rated current value.	DRV.IPEAK
5085h	0	UINT32		RW	no	Current that will be used during the DRV.ZERO procedure.	DRV.IZERO
508Ch	0	UINT32		RW	no	Number of Biss Sensor (Position) Bits for the BiSS Mode C encoder in use.	FB1.BISSBITS
508Fh	0	UINT32		RW	no	Initial feedback value as signed or unsigned.	FB1.INITSIGNED

Index	Sub-index	Data Type	Float Scaling	Access	PDO mapp.	Description	ASCII object
5096h	0	UINT32		RW	no	Current value used during the phase finding procedure (PFB.PFIND=1)	FB1.PFINDCMDU
5097h	0	UINT32		RW	no	Number of feedback poles.	FB1.POLES
5099h	0	UINT32		RW	no	Resolver nominal transformation ratio.	FB1.RESKTR
509Ah	0	UINT32		RW	no	Electrical degrees of phase lag in the resolver.	FB1.RESREFPHASE
509Ch	0	UINT32		RW	no	Controls tracking calibration algorithm.	FB1.TRACKINGCAL
50B1h	0	UINT32		RW	no	Number of successful synchronized cycles needed to lock the PLL.	FBUS.PLLTHRESH
50B7h	0	UINT32		RW	no	Maximum allowed acceleration value; active in opmode 2 (position) only.	GEAR.ACCMAX
50B9h	0	UINT32		RW	no	Maximum allowed deceleration value; active in opmode 2 (position) only.	GEAR.DECMAX
50BBh	0	UINT32		RW	no	Denominator of the electronic gearing ratio; active in opmode 2 (position) only.	GEAR.IN
50BCh	0	UINT32		RW	no	Electronic gearing mode; active in opmode 2 (position) only.	GEAR.MODE
50BEh	0	UINT32		RW	no	Numerator of the electronic gearing ratio; active in opmode 2 (position) only.	GEAR.OUT
50E2h	0	UINT32		RW	no	Current loops fieldbus injected feed-forward gain	IL.KBUSFF
50FBh	0	UINT32		RW	no	Motor pitch.	MOTOR.PITCH
50FEh	0	UINT32		RW	no	Type of thermal resistor inside the motor.	MOTOR.RTYPE
5104h	0	UINT32		RW	no	Motor type.	MOTOR.TYPE
510Eh	0	UINT32		RW	no	Motion task to be triggered after an emergency stop procedure; active in opmode 2 (position) only.	MT.EMERGMT
5121h	0	UINT32		RW	no	Type of following error warning and fault usage.	PL.ERRMODE
5128h	0	UINT32		RW	no	Feedback source for the position loop.	PL.FBSOURCE

Index	Sub-index	Data Type	Float Scaling	Access	PDO mapp.	Description	ASCII object
5175h	0	UINT32		RW	no	Service motion current 1; active in opmode 0 (torque) only.	SM.I1
5176h	0	UINT32		RW	no	Service motion current 2; active in opmode 0 (torque) only.	SM.I2
5177h	0	UINT32		RW	no	Service motion mode.	SM.MODE
5179h	0	UINT32		RW	no	Service motion time 1.	SM.T1
517Ah	0	UINT32		RW	no	Service motion time 2.	SM.T2
517Eh	0	UINT32		RW	no	Enables and disables software travel limit switches.	SWLS.EN
5184h	0	UINT32		RW	no	Linear accel-eration/deceleration units.	UNIT.ACCLINEAR
5187h	0	UINT32		RW	no	Linear position units.	UNIT.PLINEAR
518Ah	0	UINT32		RW	no	Linear velocity units.	UNIT.VLINEAR
518Eh	0	UINT32		RW	no	Voltage level for over volt-age warning.	VBUS.OVWTHRESH
51AEh	0	UINT32		RW	no	Feedback source for the velocity loop; active in opmodes 1 (velocity) and 2 (position) only.	VL.FBSOURCE
51B0h	0	UINT32		RW	no	Mode of velocity gen-eration (Observer, d/dt); active in opmodes 1 (velocity) and 2 (position) only.	VL.GENMODE
51B3h	0	UINT32		RW	no	Scales the observer velocity signal; active in opmodes 1 (velocity) and 2 (position) only.	VL.KO
51B8h	0	UINT32		RW	no	Ratio of the estimated load moment of inertia rel-ative to the motor moment of inertia; active in opmodes 1 and 2 only.	VL.LMJR
51BAh	0	UINT32		RW	no	Bandwidth of the observer in Hz.	VL.OBSBW
51BBh	0	UINT32		RW	no	Observer operating mode.	VL.OBSMODE
51CBh	0	UINT32		RW	no	Filter mode for Digital In 1.	DIN1.FILTER
51CCh	0	UINT32		RW	no	Filter mode for Digital In 2.	DIN2.FILTER
51CDh	0	UINT32		RW	no	Filter mode for Digital In 3.	DIN3.FILTER
51CEh	0	UINT32		RW	no	Filter mode for Digital In 4.	DIN4.FILTER

Index	Sub-index	Data Type	Float Scaling	Access	PDO mapp.	Description	ASCII object
51CFh	0	UINT32		RW	no	Filter mode for Digital In 5.	DIN5.FILTER
51D0h	0	UINT32		RW	no	Filter mode for Digital In 6.	DIN6.FILTER
51D1h	0	UINT32		RW	no	Filter mode for Digital In 7.	DIN7.FILTER
51E7h	0	UINT32		RW	no	Modbus User Units Input parameter	MODBUS.PIN
51E8h	0	UINT32		RW	no	Modbus User Units Output parameter.	MODBUS.POUT
51E9h	0	UINT32		RW	no	Feedback Resolution (per rev) over Modbus.	MODBUS.PSCALE
51ECh	0	UINT32		RW	no	Secondary feedback (FB2) resolution.	FB2.ENCRES
51EDh	0	UINT32		RW	no	Mode for the second feedback inputs and high speed digital inputs.	FB2.MODE
51EEh	0	UINT32		RW	no	Source for the second feedback input.	FB2.SOURCE
51EFh	0	UINT32		RW	no	Brake apply timeout for vertical axis.	MOTOR.TBRAKETO
51F0h	0	UINT32		RW	no	i.p.	MODBUS.MSGLOG
520Ch	0	UINT32		RW	no	Scaling mode for Modbus values.	MODBUS.SCALING
520Dh	0	UINT32		RW	no	Encoder output pulse width for modes 6 to 7.	DRV.EMUEPULSEWIDTH
520Eh	0	UINT32		RW	no	Enable/disable motor velocity vs. maximum emulated encoder velocity monitoring function.	DRV.EMUECHECKSPEED
5251h	0	UINT32		RW	no	Analog input deadband mode.	AIN.DEADBANDMODE
5252h	0	UINT32		RW	no	Analog input mode	AIN.MODE
5253h	0	UINT32		RW	no	Direction of IOs from X9.	DIO10.DIR
5254h	0	UINT32		RW	no	Inverting the output voltage of the IO, when in the output direction.	DIO10.INV
5255h	0	UINT32		RW	no	Direction of IOs from X9.	DIO11.DIR
5256h	0	UINT32		RW	no	Inverting the output voltage of the IO, when in the output direction.	DIO11.INV
5257h	0	UINT32		RW	no	Direction of IOs from X9.	DIO9.DIR
5258h	0	UINT32		RW	no	Inverting the output voltage of the IO, when in the output direction.	DIO9.INV
5259h	0	UINT32		RW	no	Fault Action for Fault 130.	FAULT130.ACTION

Index	Sub-index	Data Type	Float Scaling	Access	PDO mapp.	Description	ASCII object
525Ah	0	UINT32		RW	no	Fault Action for Fault 131.	FAULT131.ACTION
525Bh	0	UINT32		RW	no	Fault Action for Fault 132.	FAULT132.ACTION
525Ch	0	UINT32		RW	no	Fault Action for Fault 133.	FAULT134.ACTION
525Dh	0	UINT32		RW	no	Fault Action for Fault 702.	FAULT702.ACTION
525Eh	0	UINT32		RW	no	Method of acquiring IP Address.	IP.MODE
525Fh	0	UINT32		RW	no	Load inertia.	LOAD.INERTIA
5260h	0	UINT32		RW	no	Motor back EMF constant.	MOTOR.KE
5261h	0	UINT32		RW	no	Changing voltage thresholds for HV and MV Drives	VBUS.HALFVOLT
5262h	0	UINT32		RW	no	Direction for the second feedback input (X9 and X7).	FB2.DIR
5263h	0	UINT32		RW	no	Feedback for handwheel operation.	DRV.HANDWHEELSRC
5264h	0	UINT32		RW	no	Delay time between inactive Hardware Enable input and drive disable.	DRV.HWENDELAY
5265h	0	UINT32		RW	no	Index into the Current Loop Gain Scheduling Table.	IL.KPLOOKUPINDEX
5266h	0	UINT32		RW	no	Value of the current loop gain scheduling index.	IL.KPLOOKUPVALUE
5267h	0	UINT32		RW	no	Fault Action for Fault 451.	FAULT451.ACTION
5268h	0	UINT32		RW	no	Brake Immediately in the case of a drive disable.	MOTOR.BRAKEIMM

7.1.4 Profile specific SDOs

Index	Sub-index	DataType	Float Scaling	Access	PDO mapp.	Description	ASCII object
6040h	0	U16		WO	yes	Control word	—
6041h	0	U16		RO	yes	Status word	—
6060h	0	INT8		RW	yes	Modes of Operation	—
6061h	0	INT8		RO	yes	Modes of Operation Display	—
6063h	0	INT32		RO	yes	Position actual value (increments)	—
6064h	0	INT32	1:1	RO	yes	Position actual value (position units)	PL.FB
6065h	0	U32	1:1	RW	no	Following error window	PL.ERRFTHRESH
606Bh	0	INT32	1:1	RO	no	Velocity demand value	VL.CMD
606Ch	0	INT32	1000:1	RO	yes	Velocity actual value (PDO in RPM)	VL.FB
606Dh	0	U16		RW	yes	Velocity window	
606Eh	0	U16		RW	yes	Velocity window time	
6071h	0	INT16		RW	yes	Target torque	—
6072h	0	U16		RW	yes	Max torque	—
6073h	0	U16		RW	no	Max current	
6077h	0	INT16		RO	yes	Torque actual value	—
607Ah	0	INT32	1:1	RW	yes	Target position	MT.P
607Ch	0	INT32	1:1	RW	no	Reference offset	HOME.P
607Dh		ARRAY				Software position limit	
607Dh	0	U8		RO	no	Number of entries	
607Dh	1	INT32	1:1	RW	no	Software position limit 1	SWLS.LIMIT0
607Dh	2	INT32	1:1	RW	no	Software position limit 2	SWLS.LIMIT1
6081h	0	U32	1:1	RW	yes	Profile Velocity	MT.V
6083h	0	U32	1:1	RW	yes	Profile Acceleration	MT.ACC
6084h	0	U32	1:1	RW	yes	Profile Deceleration	MT.DEC
608Fh		ARRAY				Position encoder resolution	—
608Fh	0	U8		RO	no	Number of entries	—
608Fh	1	U32		RW	no	Encoder increments	—
608Fh	2	U32		RW	no	Motor revolutions	
6091h		ARRAY				Gear ratio	—
6091h	0	U8		RO	no	Number of entries	—
6091h	1	U32		RW	no	Motor revolution	
6091h	2	U32		RW	no	Shaft revolutions	
6092h		ARRAY				Feed constant	—
6092h	0	U8		RO	no	Number of entries	—
6092h	1	U32		RW	no	Feed	UNIT.PIN
6092h	2	U32		RW	no	Shaft revolutions	UNIT.POUT
6098h	0	INT8		RW	no	Homing type	HOME.MODE HOME.DIR
6099h		ARRAY				Homing velocity	—
6099h	0	U8		RO	no	Number of entries	—
6099h	1	U32	1:1	RW	no	Speed while searching for limit switch	HOME.V

Index	Sub-index	DataType	Float Scaling	Access	PDO mapp.	Description	ASCII object
6099h	2	U32		RW	no	Speed while searching for zero mark	HOME.FEEDRATE
609Ah	0	U32	1:1	RW	no	Homing acceleration	HOME.ACC HOME.DEC
60B1h	0	INT32	1:1	RW	yes	Velocity offset	VL.BUSFF
60B2h	0	INT16		RW	yes	Torque offset (PDO only)	
60B8h	0	U16		RW	yes	Touch probe function	—
60B9h	0	U16		RW	yes	Touch probe status	—
60BAh	0	INT32		RW	yes	Touch probe 1 positive edge	—
60BBh	0	INT32		RW	yes	Touch probe 1 negative edge	—
60BCh	0	INT32		RW	yes	Touch probe 2 positive edge	—
60BDh	0	INT32		RW	yes	Touch probe 2 negative edge	—
60C0h	0	INT16		RW	no	Interpolation submode select	—
60C1h		ARRAY				Interpolation data record	—
60C1h	0	U8		RO	no	Number of entries	—
60C1h	1	INT32		RW	yes	Interpolation target position	—
60C1h	2	U32		RW	yes	Interpolation time	—
60C1h	3	INT32		RW	yes	Interpolation target velocity	—
60C2h		RECORD				Interpolation time period	—
60C2h	0	U8		RO	no	Number of entries	FBUS.SAMPLEPERIOD
60C2h	1	U8		RW	no	Interpolation time units	—
60C2h	2	INT16		RW	no	Interpolation time index	—
60C4h		RECORD				Interpolation data configuration	—
60C4h	0	U8		RO	no	Number of entries	—
60C4h	1	U32		RO	no	Maximum buffer size	—
60C4h	2	U32		RO	yes	Actual buffer size	—
60C4h	3	U8		RW	no	Buffer organization	—
60C4h	4	U16		RW	no	Buffer position	—
60C4h	5	U8		WO	no	Siza of data record	—
60C4h	6	U8		WO	no	Buffer clear	—
60D0h		ARRAY				Touch probe source	—
60D0h	0	U8		RO	no	Highest sub-index supported	-
60D0h	1	INT16		RW	no	Touch probe 1 source	—
60D0h	2	INT16		RW	no	Touch probe 2 source	—
60F4h	0	INT32		RO	yes	Following error actual value	PL.ERR
60FDh	0	U32		RO	yes	Digital inputs	DIN1.MODE TO DIN6.MODE
60FEh		ARRAY				Digital outputs	
60FEh	0	U8		RO	no	Number of entries	
60FEh	1	U32		RW	yes	Physical outputs	
60FEh	2	U32		RW	no	Bit mask	
60FFh	0	INT32		RW	yes	Target velocity	VL.CMDU
6502h	0	U32		RO	no	Supported drive modes	—

7.2 Examples

7.2.1 Examples, setup

All examples are valid for the AKD. All values are hexadecimal.

7.2.1.1 Basic testing of the connection to the AKD controls

When the AKD is switched on, a boot-up message is transmitted over the bus. The telegram continues to be transmitted, as long as it has not yet found a suitable receiver in the bus system.
If a CAN master is unable to recognize this message, then the following measures can be taken to test communication:

- Check the bus cable: correct characteristic impedance, correct termination resistors at both ends?
- With a multimeter: check the quiescent level of the bus cables CAN-H and CAN-L against CAN-GND (approx. 2.5 V).
- With an oscilloscope: check the output signals on CAN-H and CAN-L at the AKD. Are signals being transmitted on the bus? The voltage difference between CAN-H and CAN-L for a logical "0" is approx. 2-3 V.
- Does signal transmission stop if the master is connected?
- Check the master hardware.
- Check the master software!

7.2.1.2 Example: Operating the Status Machine

NOTE **The status machine must be used sequentially during boot-up period. Leaving out a state (except for state "switched on") is not possible.**

When the AKD is switched on and the boot-up message has been detected, communication via SDOs can be initiated. For example: all the parameters can be read out or written to, or the status machine for the drive can be controlled.

The state of the status machine can be obtained through the query of Object 6041h Sub 0.

Directly after switch-on, a value will be returned, such as 0240h. This corresponds to the status "Switch on disabled".

The following data would then be visible on the CAN bus:

COB-ID	Control byte	Index		Sub-index	Data	Comment
		Low byte	High byte			
603	40	41	60	00h	40 00 00 00	
583	4B	41	60	00h	40 02 00 00	response telegram
	2 bytes of data				status	

If the supply power is present and the hardware enable is at the High level (24 V to DGND) then you can try to switch the drive to the state "Switched on" by writing 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 = 60h).

Switch on

The messages then appear as follows:

COB-ID	Control byte	Index		Sub-index	Data	Comment
		Low byte	High byte			
603	2B	40	60	00h	06 00 00 00	Shut down
583	60	40	60	00h	00 00 00 00	response telegram
603	2B	40	60	00h	07 00 00 00	Switch on
583	60	40	60	00h	00 00 00 00	response telegram

control word = 0x0007 meaning:
Bit 0, Bit 1, Bit 2 set => Switch On,
Disable Voltage off, Quick Stop off

Status query 2

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	00h	—	query status
583	4B	41	60	00h	33 02 00 00	response telegram

Status = 0x0233 meaning:
Bit 0, Bit 1, Bit 5 set => ready to Switch On,
Bit 9 set => remote, operation possible via RS232

7.2.1.3 Example: Jog Mode via SDO

The motor shall work with constant velocity.

COB-ID	Control byte	Index		Sub-index	Data	Comment
		Low byte	High byte			
603	2F	60	60	00h	03 00 00 00	Mode of operation “Profile Velocity”
583	60	60	60	00h	00 00 00 00	response telegram
603	23	FF	60	00h	00 00 00 00	setpoint=0
583	60	FF	60	00h	00 00 00 00	response telegram
603	2B	40	60	00h	06 00 00 00	shutdown
583	60	40	60	00h	00 00 00 00	response telegram
603	2B	40	60	00h	07 00 00 00	switch on
583	60	40	60	00h	00 00 00 00	response telegram
603	2B	40	60	00h	0F 00 00 00	enable operation
583	60	40	60	00h	00 00 00 00	response telegram
603	23	FF	60	00h	00 41 00 00	velocity setpoint
583	60	FF	60	00h	00 00 00 00	response telegram
603	2B	40	60	00h	0F 01 00 00	Intermediate Stop
583	60	40	60	00h	00 00 00 00	response telegram

7.2.1.4 Example: Torque Mode via SDO

The motor shall work with constant torque. CAN data:

COB-ID	Control byte	Index		Sub-index	Data	Comment
		Low byte	High byte			
603	2F	60	60	00h	04 00 00 00	Mode of operation “Torque”
583	60	60	60	00h	00 00 00 00	response telegram
603	2B	71	60	00h	00 00 00 00	setpoint=0
583	60	71	60	00h	00 00 00 00	response telegram
603	2B	40	60	00h	06 00 00 00	shutdown
583	60	40	60	00h	00 00 00 00	response telegram
603	2B	40	60	00h	07 00 00 00	switch on
583	60	40	60	00h	00 00 00 00	response telegram
603	2B	40	60	00h	0F 00 00 00	enable operation
583	60	40	60	00h	00 00 00 00	response telegram
603	2B	71	60	00h	90 01 00 00	setpoint 400 mA
583	60	71	60	00h	00 00 00 00	response telegram
603	2B	40	60	00h	0F 01 00 00	intermediate Stop
583	60	40	60	00h	00 00 00 00	response telegram

7.2.1.5 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	00h	03 00 00 00	mode of operation “Profile Velocity”
583	60	60	60	00h	00 00 00 00	response telegram
603	2F	00	16	00h	00 00 00 00	delete entries for the first RXPDO
583	60	00	16	00h	00 00 00 00	response telegram
603	23	00	16	01h	20 00 FF 60	mapping RXPDO1, Object 60FF, Subindex 0 speed setpoint, data length 32bit
583	60	00	16	01h	00 00 00 00	response telegram
603	2F	00	16	00h	01 00 00 00	confirm number of mapped objects
583	60	00	16	00h	00 00 00 00	response telegram
603	2F	00	1A	00h	00 00 00 00	delete entries for the first TXPDO
583	60	00	1A	00h	00 00 00 00	response telegram
603	23	00	1A	01h	20 00 64 60	mapping TXPDO1/1, Object6064, Subindex 0 current position value in SI units, data length 32bit
583	60	00	1A	01h	00 00 00 00	response telegram
603	23	00	1A	02h	20 00 6C 60	mapping TXPDO1/2, Object606C, Subindex 0 current speed value, data length 32bit
583	60	00	1A	02h	00 00 00 00	response telegram
603	2F	00	1A	00h	02 00 00 00	check number of mapped objects
583	60	00	1A	00h	00 00 00 00	response telegram
603	2F	00	18	02h	01 00 00 00	set TXPDO1 to synchronous, transmission with every SYNC
583	60	00	18	02h	00 00 00 00	response telegram
603	23	01	18	01h	83 02 00 80	disable TPDO2, set bit 31 (80h)
583	60	01	18	01h	00 00 00 00	response telegram
603	23	02	18	01h	83 03 00 80	disable TPDO3
583	60	02	18	01h	00 00 00 00	response telegram
603	23	03	18	01h	83 04 00 80	disabled TPDO4
583	60	03	18	01h	00 00 00 00	response telegram
603	23	01	14	01h	03 03 00 80	disabled RPDO2
583	60	01	14	01h	00 00 00 00	response telegram
603	23	02	14	01h	03 04 00 80	disabled RPDO3
583	60	02	14	01h	00 00 00 00	response telegram
603	23	03	14	01h	03 05 00 80	disabled RPDO4
583	60	03	14	01h	00 00 00 00	response telegram
000					01 03	enable NMT
603	2B	40	60	00h	06 00 00 00	shutdown
583	60	40	60	00h	00 00 00 00	response telegram
603	2B	40	60	00h	07 00 00 00	switch on
583	60	40	60	00h	00 00 00 00	response telegram
603	2B	40	60	00h	0F 00 00 00	enable operation
583	60	40	60	00h	00 00 00 00	response telegram

COB-ID	Control byte	Index		Sub-index	Data	Comment
		Low byte	High byte			
203					00 40	velocity setpoint
080						send SYNC
183					FE 45 01 00 A6 AB 1A 00	response
603	2B	40	60	00h	0F 01 00 00	intermediate stop
583	60	40	60	00h	00 00 00 00	response telegram

7.2.1.6 Example: Torque Mode via PDO

It is useful to disable unused PDOs. The first TX_PDO shall transmit the actual current value with every SYNC.

COB-ID	Control byte	Index		Sub-index	Data	Comment
		Low byte	High byte			
603	2F	60	60	00h	04 00 00 00	Mode of operation “Torque”
583	60	60	60	00h	00 00 00 00	response telegram
603	2F	00	16	00h	00 00 00 00	delete entry for the first RXPDO
583	60	00	16	00h	00 00 00 00	response telegram
603	23	00	16	01h	10 00 71 60	mapping RXPDO1, Object6071,Subindex 0 current setpoint, data length 16bit
583	60	00	16	01h	00 00 00 00	response telegram
603	2F	00	16	00h	01 00 00 00	check number of mapped objects
583	60	00	16	00h	00 00 00 00	response telegram
603	2F	00	1A	00h	00 00 00 00	delete entry for TXPDO1
583	60	00	1A	00h	00 00 00 00	response telegram
603	23	00	1A	01h	10 00 77 60	mapping TXPDO1, Object6077,Subindex 0 actual current value, Data length16bit
583	60	00	1A	01h	00 00 00 00	response telegram
603	2F	00	1A	00h	01 00 00 00	number of mapped objects
583	60	00	1A	00h	00 00 00 00	response telegram
603	2F	00	18	02h	01 00 00 00	set TXPDO1 to synchronous, transmission with every SYNC
583	60	00	18	02h	00 00 00 00	response telegram
603	23	01	18	01h	83 02 00 80	disable TPDO2, set bit 31 (80h)
583	60	01	18	01h	00 00 00 00	response telegram
603	23	02	18	01h	83 03 00 80	disable TPDO3
583	60	02	18	01h	00 00 00 00	response telegram
603	23	03	18	01h	83 04 00 80	disabled TPDO4
583	60	03	18	01h	00 00 00 00	response telegram
603	23	01	14	01h	03 03 00 80	disabled RPDO2
583	60	01	14	01h	00 00 00 00	response telegram
603	23	02	14	01h	03 04 00 80	disabled RPDO3
583	60	02	14	01h	00 00 00 00	response telegram
603	23	03	14	01h	03 05 00 80	disabled RPDO4
583	60	03	14	01h	00 00 00 00	response telegram
000					01 03	enable NMT
603	2B	40	60	00h	06 00 00 00	shutdown
583	60	40	60	00h	00 00 00 00	response telegram
603	2B	40	60	00h	07 00 00 00	switch on
583	60	40	60	00h	00 00 00 00	response telegram
603	2B	40	60	00h	0F 00 00 00	enable operation
583	60	40	60	00h	00 00 00 00	response telegram
203					12 02	setpoint 530 mA
080						send SYNC
183					19 02	actual value 537 mA
603	2B	40	60	00h	0F 01 00 00	intermediate stop
583	60	40	60	00h	00 00 00 00	response telegram

7.2.1.7 Example: Homing via SDO

When the AKD is operated as a linear axis, a reference/homing point must be defined before positioning tasks can be executed. This must be done by executing a homing run in the Homing mode (0x6).

This example shows the procedure in the Homing mode.

Now some of the parameters that affect the homing movement are set via the bus. If you can be absolutely certain that no-one has altered the parameters in the servoamplifier, then this part can be omitted, since the servoamplifier save the data in non-volatile memory. The inputs must be configured as limit switches.

Because the dimension parameters are not finally defined in DS402, you must select these units:

UNIT.PROTARY = 3
UNIT.VROTARY = 3
UNIT.ACCROTARY = 3

The basic setup of the servoamplifier must be done with the help of the setup software before starting the homing run. The resolution has been set to 10000 µm/turn in this example.

COB-ID	Control byte	Index		Sub-index	Data	Comment
		Low byte	High byte			
703	00					boot-up message
603	40	41	60	00h	00 00 00 00	read profile status
583	4B	41	60	00h	40 02 00 00	response : 0240h
603	23	99	60	01h	10 27 00 00	v_{ref}=10000 counts/s until limit switch is reached
583	60	99	60	01h	00 00 00 00	response telegram
603	23	99	60	02h	88 13 00 00	v_{ref}=5000 counts/s from limit switch to zero mark
583	60	99	60	02h	00 00 00 00	response telegram
603	23	9A	60	00h	10 27 00 00	Decel. and Accel. ramp 1000counts/s²
583	60	9A	60	00h	00 00 00 00	response telegram
603	23	7C	60	00h	A8 61 00 00	Reference offset 25000counts
583	60	7C	60	00h	00 00 00 00	response telegram

Homing type (6098h)

COB-ID	Control byte	Index		Sub-index	Data	Comment
		Low byte	High byte			
603	2F	60	60	00h	06 00 00 00	mode of operation = homing
583	60	60	60	00h	00 00 00 00	response telegram
603	40	41	60	00h	00 00 00 00	read profile status, response: 0250h Voltage Enabled
583	4B	41	60	00h	40 02 00 00	response : 0240h
603	2B	40	60	00h	06 00 00 00	Control word Transition_2,"ready to switch on". Shutdown
583	60	40	60	00h	00 00 00 00	response telegram
603	2B	40	60	00h	07 00 00 00	Transition_3, "switch on". switch on
583	60	40	60	00h	00 00 00 00	response telegram
603	2B	40	60	00h	0F 00 00 00	Transition_4,"operation enable"
583	60	40	60	00h	00 00 00 00	response telegram
603	40	41	60	00h	00 00 00 00	read profile status
583	4B	41	60	00h	37 02 00 00	response telegram
603	2B	40	60	00h	1F 00 00 00	Homing_operation_start
583	60	40	60	00h	00 00 00 00	response telegram
603	40	41	60	00h	00 00 00 00	read profile status, response: homing not finished
583	4B	41	60	00h	37 02 00 00	response: homing not finished
603	40	41	60	00h	00 00 00 00	read profile status, response: homing finished
583	4B	41	60	00h	37 16 00 00	response:homing finished

Bit 12 in SDO 6041 indicates, whether homing is finished. Reading of the profile status is not necessary.

7.2.1.8 Example: Using the Profile Position Mode

This example shows the operation of the Profile position mode. For this, the PDOs are set as follows:

First RPDO. No special mapping necessary, because the default mapping enters the controlword RXPDO1.

Second RPDO

COB-ID	Control	Index		Sub-	Data	Comment
	byte	Low byte	High byte	index		
603	2F	01	16	00h	00 00 00 01	RPDO2: delete mapping
583	60	01	16	00h	00 00 00 00	response telegram
603	23	01	16	01h	20 00 7A 60	RPDO2, entry 1: target_position
583	60	01	16	01h	00 00 00 00	response telegram
603	23	01	16	02h	20 00 81 60	RPDO2, entry 2: profile_velocity
583	60	01	16	02h	00 00 00 00	response telegram
603	2F	01	16	00h	02 00 00 00	enter number of mapped objects
583	60	01	16	00h	00 00 00 00	response telegram

First TPDO

COB-ID	Control	Index		Sub-	Data	Comment
	byte	Low byte	High byte	index		
603	2F	00	1A	00h	00 00 00 01	TPDO1: delete mapping
583	60	00	1A	00h	00 00 00 00	response telegram
603	23	00	1A	01h	10 00 41 60	TPDO1, entry 1: profile statusword
583	60	00	1A	01h	00 00 00 00	response telegram
603	2F	00	1A	00h	01 00 00 00	enter number of mapped objects
583	60	00	1A	00h	00 00 00 00	response telegram

Second TPDO

COB-ID	Control	Index		Sub-	Data	Comment
	byte	Low byte	High byte	index		
603	2F	01	1A	00h	00 00 00 01	TPDO2: delete mapping
583	60	01	1A	00h	00 00 00 00	response telegram
603	23	01	1A	01h	20 00 64 60	TPDO2, entry 1: position_actual_value
583	60	01	1A	01h	00 00 00 00	response telegram
603	23	01	1A	02h	20 00 6C 60	TPDO2, entry 2: velocity_actual_value
583	60	01	1A	02h	00 00 00 00	response telegram
603	2F	01	1A	00h	02 00 00 00	enter number of mapped objects
583	60	01	1A	00h	00 00 00 00	response telegram

The second TPDO should be sent with every SYNC by the servoamplifier.

COB-ID	Control	Index		Sub-	Data	Comment
	byte	Low byte	High byte	index		
603	2F	01	18	02h	01 00 00 00	TPDO2 with every SYNC
583	60	01	18	02h	00 00 00 00	response telegram

Disable unused TPDOs

COB-ID	Control	Index		Sub-	Data	Comment
	byte	Low byte	High byte	index		
603	23	02	18	01h	83 03 00 80	disable TPDO3
583	60	02	18	01h	00 00 00 00	response telegram
603	23	03	18	01h	83 04 00 80	disable TPDO4
583	60	03	18	01h	00 00 00 00	response telegram

Disable unused RPDOs

COB-ID	Control	Index		Sub-	Data	Comment
	byte	Low byte	High byte	index		
603	23	02	14	01h	03 04 00 80	disable RPDO3
583	60	02	14	01h	00 00 00 00	response telegram
603	23	03	14	01h	03 05 00 80	disable RPDO4
583	60	03	14	01h	00 00 00 00	response telegram

Define mechanical resolution via Object 6092h, Subindex 01h and 02h. Default values are the motion specific factors PGEARI and PGEARO:

COB-ID	Control	Index		Sub-	Data	Comment
	byte	Low byte	High byte	index		
603	23	93	60	01h	00 00 10 00	2E20 increments
583	60	93	60	01h	00 00 00 00	response telegram
603	23	93	60	02h	A0 8C 00 00	3600 user units
583	60	93	60	02h	00 00 00 00	response telegram

After defining the PDOs they can be released with the NMT:

COB-ID		Data	Comment
000		01 03	enable NMT
183		40 02	profile status

Now the homing can be set and started.

COB-ID	Control	Index		Sub-	Data	Comment
	byte	Low byte	High byte	index		
603	2F	60	60	00h	06 00 00 00	Operation mode = homing
583	60	60	60	00h	00 00 00 00	response telegram
603	2F	98	60	00h	0C 00 00 00	homing type 12, negative direction (DS402)
583	60	98	60	00h	00 00 00 00	response telegram
603	23	99	60	01h	40 19 01 00	homing speed 72000 units/s=2s-1
583	80	99	60	01h	31 00 09 06	response telegram
603	2B	40	60	00h	06 00 00 00	Transition_2,"ready to switch on".Shutdown
583	60	40	60	00h	00 00 00 00	response telegram
183					21 02	response telegram
603	2B	40	60	00h	07 00 00 00	Transition_3,"switch on".Switch on
583	60	40	60	00h	00 00 00 00	response telegram
183					33 02	response telegram
603	2B	40	60	00h	0F 00 00 00	Control word: Operation Enable
583	60	40	60	00h	00 00 00 00	response telegram
183					37 02	response telegram
603	2B	40	60	00h	1F 00 00 00	start homing response telegram response: target reached response: homing attained
583	60	40	60	00h	00 00 00 00	response telegram
183					37 06	
183					37 16	

Finish homing with Control word 1_RPDO

COB-ID		Data	Comment
203		0F 00	

Switch to Profile Position Mode and set ramps for positioning

COB-ID	Control	Index		Sub-	Data	Comment
	byte	Low byte	High byte	index		
603	2F	60	60	00h	01 00 00 00	Profile Positioning Mode
583	60	60	60	00h	00 00 00 00	response telegram
603	23	83	60	00h	32 00 00 00	50ms acceleration time
583	60	83	60	00h	00 00 00 00	response telegram
603	23	84	60	00h	32 00 00 00	50ms deceleration time
583	60	84	60	00h	00 00 00 00	response telegram

Setpoint

COB-ID		Data	Comment
303		20 4E 00 00	Pos 8CA0 =36000µm ; V= 20000 µm/s
080			send a SYNC
283		BB F8 FF FF	response telegram

Set controlword with „new setpoint“ by bit (bit 4)

COB-ID		Data	Comment
203		1F 00	

Wait

COB-ID		Data	Comment
183		37 12	setpoint acknowledge

Reset controlword with „new setpoint“ by bit (bit 4) reset

COB-ID		Data	Comment
203		0F 00	
183		37 02	reset Setpoint acknowledge

Wait

COB-ID		Data	Comment
183		37 06	response: target reached
080			SYNC
283		92 FC FF FF	response: 92 FC position , FF FF speed

7.2.1.9 Example: ASCII Communication

The following example reads the active faults from the drive (ASCII command DRV.FAULTS).

COB-ID	Control	Index		Sub-	Data	Comment
	byte	Low byte	High byte	index		
601	23	26	20	01h	44 52 56 2E	send ASCII code "DRV."
581	60	26	20	01h	00 00 00 00	response telegram
601	23	26	20	01h	46 41 55 4C	send ASCII code "FAUL"
581	60	26	20	01h	00 00 00 00	response telegram
601	23	26	20	01h	54 53 0D 0A	send ASCII code "TS\r\n"
581	60	26	20	01h	00 00 00 00	response telegram
601	40	26	20	02h	00 00 00 00	read response
581	43	26	20	02h	3E 4E 6F 20	read ASCII code ">No"
601	40	26	20	02h	00 00 00 00	read response
581	43	26	20	02h	66 61 75 6C	read ASCII code "FAUL"
601	40	26	20	02h	00 00 00 00	read response
581	43	26	20	02h	74 73 20 61	read ASCII code "ts a"
601	40	26	20	02h	00 00 00 00	read response
581	43	26	20	02h	63 64 69 76	read ASCII code "ctiv"
601	40	26	20	02h	00 00 00 00	read response
581	43	26	20	02h	66 0A 0D 0A	read ASCII code "e\n\r\n"

7.2.1.10 Test for SYNC telegrams

Configuration

- Assign Target Position and Profile Velocity to a PDO (2nd receive-PDO)
- Assign Actual Position to a PDO (1st transmit-PDO), generated with every 2nd SYNC.
- Assign Status word and Manufacturer Status to a PDO (2nd transmit-PDO), generated with every 3rd SYNC.

Telegrams with the corresponding responses:

COB-ID	Control byte	Index		Sub-index	Data	Comment
		Low byte	High byte			
603	2F	01	16	00h	00 00 00 00	RPDO2: delete mapping
583	60	01	16	00h	00 00 00 00	
603	23	01	16	01h	20 00 7A 60	RPDO2, entry 1: target position
583	60	01	16	01h	00 00 00 00	
603	23	01	16	02h	20 00 81 60	RPDO2, entry 2: profile velocity
583	60	01	16	02h	00 00 00 00	
603	2F	01	16	00h	02 00 00 00	RPDO2: enter number of mapped objects
583	60	01	16	00h	00 00 00 00	
603	2F	00	1A	00h	00 00 00 00	TPDO1: delete mapping
583	60	00	1A	00h	00 00 00 00	
603	23	00	1A	01h	20 00 64 60	TPDO1: entry 1: Actual Position
583	60	00	1A	01h	00 00 00 00	
603	2F	00	1A	00h	01 00 00 00	TPDO1: enter number of mapped objects
583	60	00	1A	00h	00 00 00 00	
603	2F	00	18	02h	02 00 00 00	TPDO1: send with every 2nd SYNC
583	60	00	18	02h	00 00 00 00	
603	2F	01	1A	00h	00 00 00 00	TPDO2: delete mapping
583	60	01	1A	00h	00 00 00 00	
603	23	01	1A	01h	10 00 41 60	TPDO2: entry 1: Status word
583	60	01	1A	01h	00 00 00 00	
603	23	01	1A	02h	20 00 02 10	TPDO2: entry 2: Manufacturer Status
583	60	01	1A	02h	00 00 00 00	
603	2F	01	16	00h	02 00 00 00	TPDO2: enter number of mapped objects
583	60	01	16	00h	00 00 00 00	
603	2F	01	18	02h	03 00 00 00	TPDO2: send with every 3rd SYNC
583	60	01	18	02h	00 00 00 00	

SYNC-Object

COB-ID	Comment
080	Object 181 (TPDO 1) appears at every 2nd SYNC Object 281 (TPDO 2) appears at every 3rd SYNC

Emergency-Object

If, for instance, the resolver connector is disconnected, a serious error will be caused in the controller. This results in an Emergency telegram.

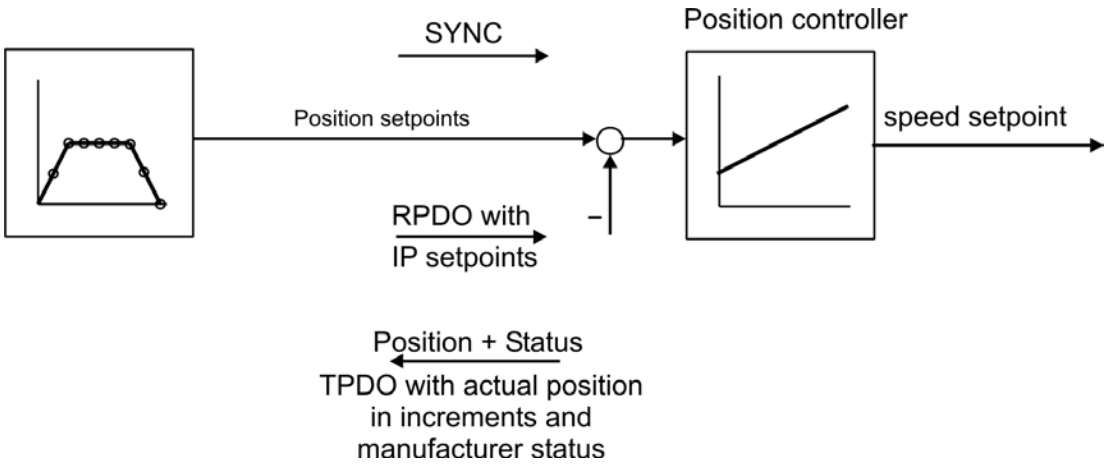
COB-ID	Emergency error		Error register		
	Low	High			
081	10	43	08	00 00 00 00	motor temperature, temperature, manufacturer specific
081	00	00	88	00 00 00 00	

7.2.2 Examples, special applications

7.2.2.1 Example: External Trajectory with Interpolated Position Mode

This example shows the possible application for giving two axes position setpoints within one PDO.

Controller structure for the position controller within the drive:



All data are hexadecimal. In the example, the two axes in the system have the station addresses 1 and 2.

Before you begin this procedure, the axes should be homed (just for this example).

The common PDO contains 2 IP (interpolated position) – setpoints and can be transmitted simultaneously to two stations, whereby each station can extract the relevant data. The other data can be made ignored by using dummy entries (Object 2100 sub 0). For this purpose both axes must react on the same RPDO-COB-ID.

Action

Do the RPDO2-mapping for both axis:

Axis 1:

COB-ID	Control	Index		Sub-	Data	Comment
	byte	Low byte	High byte	index		
601	2F	01	16	00h	00 00 00 00	RPDO2: delete mapping
581	60	01	16	00h	00 00 00 00	
601	23	01	16	01h	20 01 C1 60	RPDO2, entry 1: IP setpoint axis 1
581	60	01	16	01h	00 00 00 00	
601	23	01	16	02h	20 00 00 21	RPDO2, entry 2: Dummy entry 4 bytes
581	60	01	16	02h	00 00 00 00	
601	2F	01	16	00h	02 00 00 00	RPDO2, enter number of mapped objects
581	60	01	16	00h	00 00 00 00	

Axis 2:

COB-ID	Control	Index		Sub-	Data	Comment
	byte	Low byte	High byte	index		
602	2F	01	16	00h	00 00 00 00	RPDO2: delete mapping
582	60	01	16	00h	00 00 00 00	
602	23	01	16	02h	20 00 00 21	RPDO2, entry 1: Dummy entry 4 bytes
582	60	01	16	02h	00 00 00 00	
602	23	01	16	01h	20 01 C1 60	RPDO2, entry 2: IP setpoint axis 2
582	60	01	16	01h	00 00 00 00	
602	2F	01	16	00h	02 00 00 00	RPDO2, enter number of mapped objects
582	60	01	16	00h	00 00 00 00	
602	23	01	16	01h	01 03 00 00	RPDO2: Set COB-ID identical to axis 1
582	60	01	16	01h	00 00 00 00	

Now both axis react to the same COB-identifier 0x301, axis 1 takes byte 0 to 3 as IP setpoint, axis 2 takes byte 4 to 7. The second TPDOs shall contain the actual position in increments and the manufacturer status.

Mapping configuration for axis 1:

COB-ID	Control	Index		Sub-	Data	Comment
	byte	Low byte	High byte	index		
601	2F	01	1A	00h	00 00 00 00	TPDO2: delete mapping
581	60	01	1A	00h	00 00 00 00	
601	23	01	1A	01h	20 00 63 60	TPDO2, entry 1: actual position in increments
581	60	01	1A	01h	00 00 00 00	
601	23	01	1A	02h	20 00 02 10	TPDO2, entry 2: Dummy entry 4 bytes
581	60	01	1A	02h	00 00 00 00	
601	2F	01	1A	00h	02 00 00 00	TPDO2, enter number of mapped objects
581	60	01	1A	00h	00 00 00 00	

The same must be done for axis 2.

Here it is assumed that both drives accept new trajectory values with every SYNC command, and must return their incremen tal position and manufacturer status values. The communication parameters must be set accordingly.

Axis 1:

COB-ID	Control	Index		Sub-	Data	Comment
	byte	Low byte	High byte	index		
601	2F	01	14	02h	01 00 00 00	RPDO2 axis 1, reaction on every sync
581	60	01	14	02h	00 00 00 00	
602	2F	01	14	02h	01 00 00 00	RPDO2 axis 2, reaction on every sync
582	60	01	14	02h	00 00 00 00	
601	2F	01	18	02h	01 00 00 00	TPDO2 axis 1, reaction on every sync
581	60	01	18	02h	00 00 00 00	
602	2F	01	18	02h	01 00 00 00	TPDO2 axis 2, reaction on every sync
582	60	01	18	02h	00 00 00 00	

The other Tx-PDOs 3 and 4 should be switched off to minimize bus-load:

COB-ID	Control	Index		Sub-	Data	Comment
	byte	Low byte	High byte	index		
601	23	02	18	01h	81 03 00 80	Switch off TPDO3
581	60	02	18	01h	00 00 00 00	
601	23	03	18	01h	81 04 00 80	Switch off TPDO4
581	60	03	18	01h	00 00 00 00	

The same must be done for axis 2.

In order to be able to make trajectory movements, both drives must be operating in the appropriate mode. This is set through Index 6060h:

COB-ID	Control	Index		Sub-	Data	Comment
	byte	Low byte	High byte	index		
601	2F	60	60	00h	07 00 00 00	Set IP mode for axis 1
581	60	60	60	00h	00 00 00 00	
602	2F	60	60	00h	07 00 00 00	Set IP mode for axis 2
582	60	60	60	00h	00 00 00 00	

The cycle interval for the IP-mode shall be 1 ms. This must be defined with Object 60C1 sub 1 and 2:

COB-ID	Control	Index		Sub-	Data	Comment
	byte	Low byte	High byte	index		
601	2F	C2	60	01h	01 00 00 00	Interpolation time unit 1
581	60	C2	60	01h	00 00 00 00	
601	2F	C2	60	02h	FD 00 00 00	Interpolation time index -3 -> Cycle time = 1 * 10^{-3} s
581	60	C2	60	02h	00 00 00 00	

The same must be done for axis 2.

To start up the axes, the drives must be put into the operational status (operation enable) and the network management functions must be started.

The network management functions enable the application of the Process Data Objects (PDOs) and are initia lized by the following telegram for both axes:

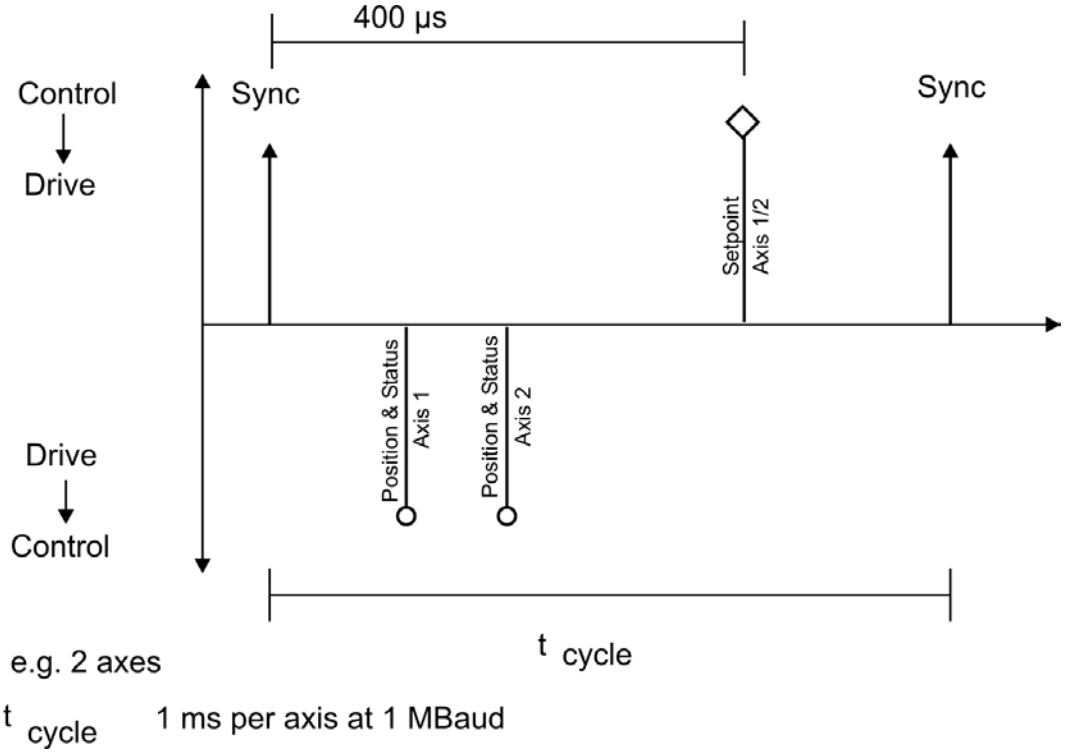
Switch the NMT (Network Management) status machine to operation enable:

COB-ID	Command specifier (CS)	Node-ID	Comment
0	1	1	NMT enable for all axes

Next, power is applied to each drive, and they are put into the operation enable condition. This should be done in steps with waiting for the appropriate reaction of the drive (e.g. axis 1):

COB-ID	Data	Comment
201	06 00	Shutdown command
181	31 02	State Ready_to_switch_on
201	07 00	Switch_on command
181	33 02	State Switched_on
201	0F 00	Enable_operation command
181	37 02	State Operation_enabled
201	1F 00	Enable IP-mode
181	37 12	IP-mode enable

The configuration above now enables a cyclical sequence, as shown in the diagram:



RPDO 2 can now be used to supply trajectory data for both axes, e.g.:

COB-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
301	F4	01	00	00	E8	03	00	00

In this example, the first axis receives a trajectory value of 500 increments (Bytes 0 to 3) and the second axis receives a trajectory value of 1000 increments.

The axes accept these values, and the positioning is made when the next SYNC telegram is received.

SYNC telegram

COB-ID
080

Afterwards, both axes send back their incremental positions and the contents of their status registers when the SYNC Object with the COB-ID for the 2nd TPDO is received.

COB-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Comment
181	23	01	00	00	00	00	03	44	position + manufacturer status register for axis1
182	A5	02	00	00	00	00	03	44	position + manufacturer status register for axis2

If an error occurs during operation, the axis transmits an Emergency message, which could appear like this:

Emergency Object

COB-ID	Emergency error		Error register	Category		
	Low	High				
081	10	43	08	01	00 00 00 00	motor temperature, temperature, manufacturer-specific
081	00	00	08	00	00 00 00 00	

7.2.2.2 Example: PVT Interpolation

This example shows the possible application for PVT. The PVT feature is cubic polynomial interpolation using target position, velocity and time from CAN bus. All data in this example is hexadecimal. Make sure that the firmware in your drive supports the PVT feature.

Step 1: Do the PDO mapping

COB-ID	Control byte	Index		Sub-index	Data	Comment
		Low byte	High byte			
601	23	01	14	01	01 03 00 80	Stop possible transmit of the RPDO2
601	2F	01	16	00	00 00 00 00	Delete actual mapping of the RPDO2
601	23	01	16	01	20 01 C1 60	Build the mapping for 60C1 sub1
601	23	01	16	02	20 02 C1 60	Build the mapping for 60C1 sub2
601	2F	01	16	00	02 00 00 00	Enter number of mapped object
601	23	01	14	01	01 03 00 00	Enable transmit of RPDO2
601	23	02	14	01	01 04 00 80	Stop possible transmit of the RPDO3
601	2F	02	16	00	00 00 00 00	Delete actual mapping of the RPDO3
601	23	02	16	01	20 03 C1 60	Build the mapping for 60C1 sub2
601	2F	02	16	00	01 00 00 00	Enter number of mapped object
601	23	02	14	01	01 04 00 00	Enable transmit of RPDO3

Step 2: Set PDO transmission type

Assuming the drive accepts new trajectory values with every SYNC command, the communication parameters must be set accordingly.

COB-ID	Control byte	Index		Sub-index	Data	Comment
		Low byte	High byte			
601	23	01	14	02	01 00 00 00	RPDO2 transmit with every sync
601	23	02	14	02	01 00 00 00	RPDO3 transmit with every sync

Step 3: Switch off unused TxPDO

TxPDO which is not used, should be switch off to minimize bus-load. In this example, all TxPDO are not used, and all are switched off.

COB-ID	Control byte	Index		Sub-index	Data	Comment
		Low byte	High byte			
601	23	00	18	01	81 02 00 80	Switch off TPDO1
601	23	01	18	01	81 02 00 80	Switch off TPDO2
601	23	02	18	01	81 03 00 80	Switch off TPDO3
601	23	03	18	01	81 04 00 80	Switch off TPDO4

Step 4: Set interpolation mode

COB-ID	Control byte	Index		Sub-index	Data	Comment
		Low byte	High byte			
601	2B	C0	60	00	00 00 FE FF	Set PVT interpolation mode
601	2F	60	60	00	07 00 00 00	Set IP mode

Step 5: Configure parameters for PVT segments

This step is optional depending on the application. It serves to modify the resolution or range of the PVT segments. AKD have default values for these two objects. The following example sets the default values.

COB-ID	Control byte	Index		Sub-index	Data	Comment
		Low byte	High byte			
601	2B	8F	60	01	00 00 10 00	Default value 2^{20}
601	2F	4C	20	01	01 00 00 00	Default value 1
601	2F	4C	20	02	01 00 00 00	Default value 1

Step 6: Switch NMT

The network management functions enable the application of the Process Data Objects (PDOs).

COB_ID	Command specifier (CS)	Node-ID	Comment
0	1	1	NMT enable

Step 7: Enable the drive

This assumes that default TPDO1 map is not changed. Execute the following steps, waiting for the appropriate reaction of the drive.

COB_ID	Data	Comment
201	06 00	Shutdown command
201	07 00	Switch on command
201	0F 00	Enable command
201	1F 00	Enable IP-mode

Step 8: Check the actual buffer size

Always check the actual buffer size before transmitting a PVT segment and make sure the usable buffer size is not 0..

COB-ID	Control byte	Index		Sub-index	Data	Comment
		Low byte	High byte			
601	40	C4	60	02	00 00 00 00	

Step 9: Transmit PVT segment

RPDO2 and RPDO3 can now be used to supply trajectory data, set the PVT segment value as required and transmit it.

The target position is absolute. Make sure that the first segment is based on the current position. The trajectory is a move from current position to the first set point.

COB_ID	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Comment
301	00	00	00	00	00	00	00	00	Position and velocity command
401	00	00	00	00					Time
80									sync

8 Index

This page intentionally left blank.

About KOLLMORGEN

Kollmorgen is a leading provider of motion systems and components for machine builders. Through world-class knowledge in motion, industry-leading quality and deep expertise in linking and integrating standard and custom products, Kollmorgen delivers breakthrough solutions that are unmatched in performance, reliability and ease-of-use, giving machine builders an irrefutable marketplace advantage.

For assistance with your application needs, visit www.kollmorgen.com or contact us at:

North America
KOLLMORGEN
203A West Rock Road
Radford, VA 24141 USA

Web: www.kollmorgen.com
Mail: support@kollmorgen.com
Tel.: +1 - 540 - 633 - 3545
Fax: +1 - 540 - 639 - 4162

Europe
KOLLMORGEN Europe GmbH
Pempelfurtstraße 1
40880 Ratingen, Germany

Web: www.kollmorgen.com
Mail: technik@kollmorgen.com
Tel.: +49 - 2102 - 9394 - 0
Fax: +49 - 2102 - 9394 - 3155

Asia
KOLLMORGEN
Rm 2205, Scitech Tower, China
22 Jianguomen Wai Street

Web: www.kollmorgen.com
Mail: sales.asia@kollmorgen.com
Tel.: +86 - 400 666 1802
Fax: +86 - 10 6515 0263