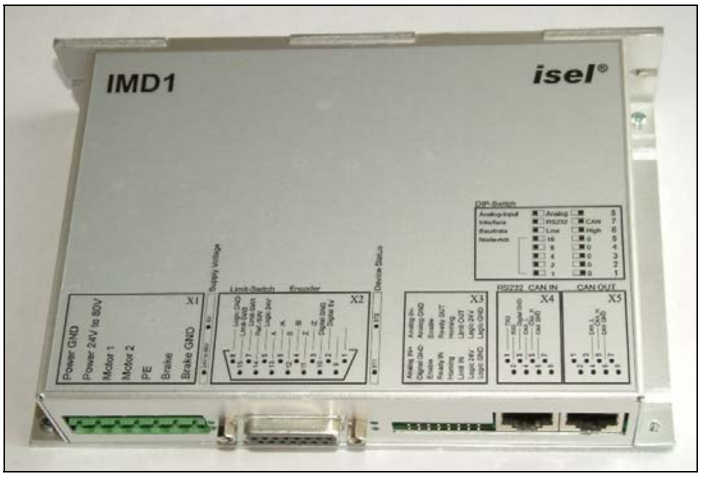
From Components to Systems

DC servo positioning module with CanOpen interface

UVE8112

IMD10



www.isel.com

Remarks concerning this manual:

Despite the utmost care, print and other errors cannot be excluded.
If you have any suggestions for improvements and hints as regards errors, don't hesitate to contact us.

Manufacturer: ***isel*** Germany AG
Bürgermeister-Ebert-Straße 40
36124 Eichenzell
Germany

Telephone: +49-(0)-6659-981-0
Fax: +49-(0)-6659-981-776
Email: automation@isel.com
http://www.isel.com

Last Updated: 05/18/2010

Table of contents

Overview

The present manual contains all descriptions and documentations required for the wiring, commissioning and control of the drive module.

It is intended for qualified personnel with basic knowledge of the control and automation technology as well as of the CAN field bus.

The basics of the CanOpen protocol are described in a separate chapter to the extent that they are important to the drive process. This chapter also presents an overview of all CanOpen objects of this module as well as a description of the drive functions.

The 3rd chapter gives a description of the DcSetup programme and serves as a guide to the commissioning of the drive.

The content of this manual is also provided as online help together with the setting programme.

Features of the drive module

- Supply voltage: 40-95 V (DC)
- Motor current up to 25 A (continuous current: 12 A)
- CAN bus interface according to CanOpen DS301 V4.0 and DS402 V1.0
- RS232 interface
- Analogue input (±10 V) with 11-bit resolution
- Inputs for limit and reference switches
- Digital current, velocity and position control with high cycle times of 100, 244 and 488 µs
  (in the mode of operation “CAN bus current control” with a cycle time of up to 70 µs possible)
  see Current Controller dialogue box on page 49 and Position Controller dialogue box on page 59
- Brake control
- Gantry operation or synchronous control of 2 modules
- Monitoring of the motor current and encoder signals
- Monitoring of the software through the internal watchdog timer
- Galvanic isolation of processor, power section and I/Os
- Easy firmware update via RS232

Scope of application

The drive module is suited ideally for the control of small- and medium-power brush-type DC motors. What has to be pointed out is the high torque even at small velocities and the outraging synchronous characteristics even at low speeds. Thanks to six modes of operation and a high diversity of adjustable parameters, a broad spectrum of the most diverse applications in the automation technology and the mechanical engineering can be covered.

The CanOpen interface is an open interface which allows to build up a flexible, extensible plant structure or to integrate the modules into existing plants.

Operational environment

To get the full functionality of the drive module the following environmental conditions should be ensured

- Ambient temperature during operation: from +5^{o}C to 40^{o}C
- Temperature during transport and storage: from -25^{o}C to 55^{o}C and at 70^{o}C for maximum 24 hours
- Maximum height of installation: about 1000m above sea level
- Maximum humidity 50% at 40^{o}C, 90% at 20^{o}C
- Vertical mounting position with appropriate distance each other to allow adequate air circulation
- Shielded motor cable with maximum lenght of 25m and minimal thickness of 1,5mm^2
- Protection in power supply

Description of the drive module

Settings and status display

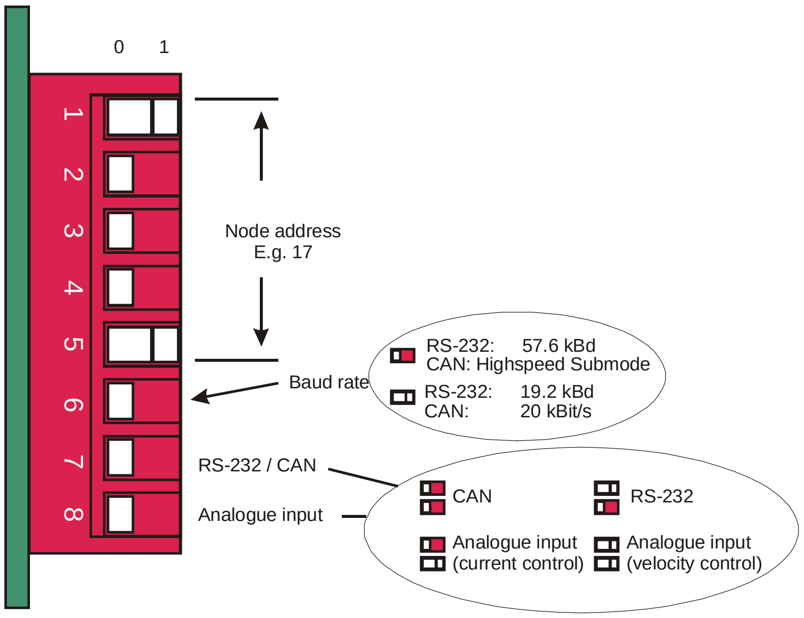
On the front (UVE8112) or on the side (IMD10) of the module, a DIP switch is provided for settings concerning the node address, the baud rate and the mode of communication / operation. Furthermore, LEDs show the current mode of operation.

DIP switch

The DIP switch is only queried, when the module is switched on or after it was reset. During the operation, a commutation of the switch does not have any effects whatsoever.

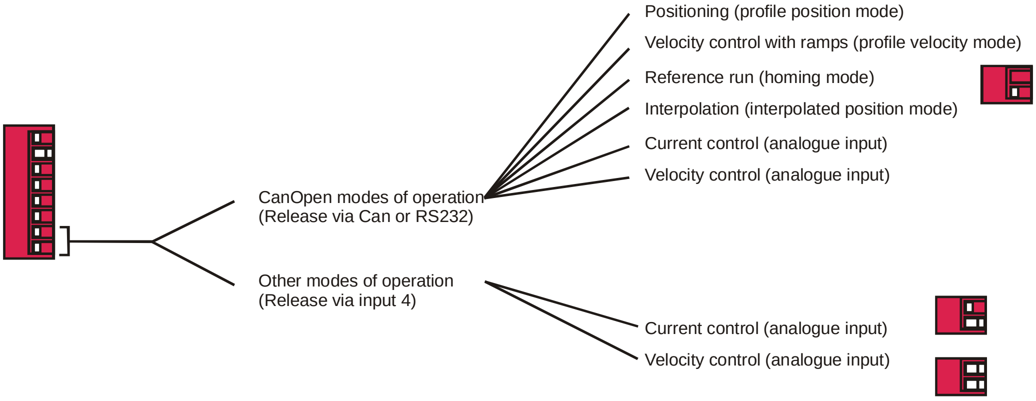
- The switches S1 to S5 are used to set up the CanOpen node address. Possible node addresses are: **1** to **30**.
  - ! If the **Node number 0** is set, Address 1 is used automatically, since 0 is reserved for the CanOpen master.
  - ! The setting of **Node number 31** is provided for the operation of a single-axis control with a special SPS. The node address is set automatically to Address 6.
- The switch S6 defines the baud rate for the CAN connection **and** the RS232 interface. In High Speed Submode the CAN baud rate of the module is determined by the object Can Baud Rate (2001) (see object directory, manufacturer specific objects, object Can Baud Rate). Under „Setting->Object directory->Manufacturer specific objects->2001 Can Baud Rate->03 New High Speed Submode“ of DCSetup you can change the CAN baud rate of the moduel (double click on the object). After switch off and switch on of the modul the new baud rate is used. When you change the baud rate of the module do not forget to change also the baud rate in DCSetup and CANSet. For the interpolation with gantry axis at least a baud rate of 250 kBit/s is needed, for normal interpolation you need at least 125 kBit/s. Default is 1 Mbit/s.
- The two switches S7 and S8 determine which communication channel has priority (CAN, RS232 or analogue input). In case of an analogue input, one of two set modes of operation is chosen (current or velocity control).

  See "Data transfer" on Page 24.



Modes of operation

Here, two groups of modes of operation are available differing from each other mainly in the type of the controller release. In case of the CanOpen modes of operation, the internal state (state machine) is controlled via the Can bus or the serial interface, while there are only two states for the other modes of operation (Enabled, Disabled) which are linked directly with the enabling signal (Input 4).



The active mode of operation of the drive module is defined through the settings of the DIP switch on the front side and through the “Modes of operation” parameter ("Modes of Operation", 6060_h).

See “Mode of operation” on page 89.

When the module is switched on, the DIP switch is queried at first. The switch S8 defines whether the “Mode of operation” parameter is evaluated or whether a set mode of operation is used.

S8			**Mode of operation**
0	The mode of operation is defined by the “Modes of operation” parameter.	**Modes of operation**	
		1	Profile position mode (profile position mode)
		3	Profile velocity mode (profile velocity mode)
		6	Reference run (homing mode)
		7	Interpolation (interpolated position mode)
		-1	Current control, analogue input
		-2	Velocity control, analogue input
		-3	Moving the axis away from a limit switch
1	The mode of operation is set firmly (depending on S7).	**S7**	
		1	Current control, analogue input
		0	Velocity control, analogue input

LEDs UVE8112

Via the LEDs on the front, it is possible to indicate the existence of the two main supply voltages. Furthermore, the current mode of operation of the module can be viewed here (STS1 and STS2).

For this, see: "Status diagram" on page 75 and "Device control" on page 87.

In the Operation Enable state, the two related LEDs (STS1 and STS2) are used to show the current motor current in the negative and positive direction. Concerning this, the maximum brightness corresponds to the peak current.

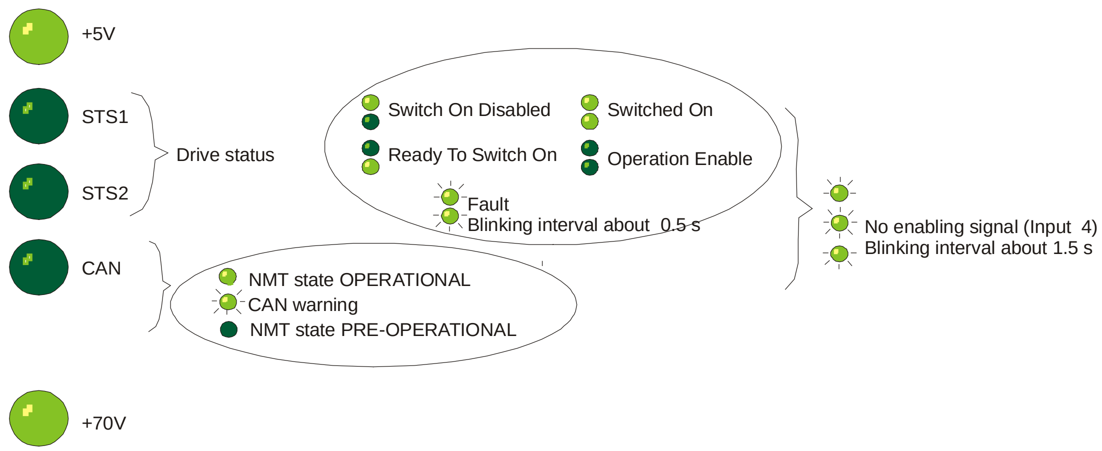
In case of a fault (Fault state), the two-digit error code of the current fault is given via a blinking code.

1st digit: number of blinking impulses of both status LEDs.

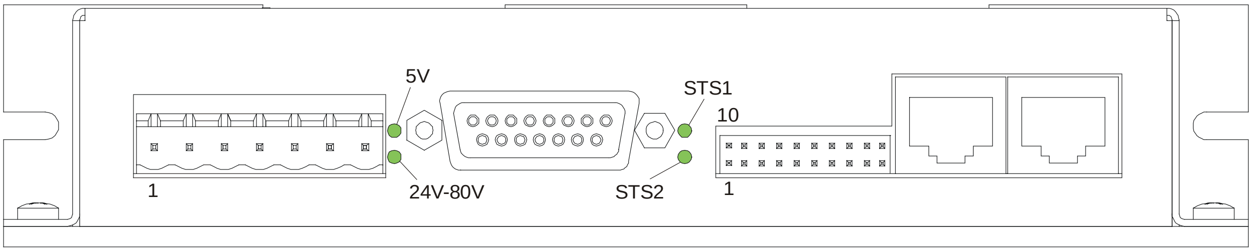
2nd digit: number of blink impulses of LED STS2.

See "EMCY" on page 74 for the description of the error code.

In case of a missing enabling signal, all three LEDs (STS1, STS2 and CAN) blink about once every 1.5 seconds.



LEDs IMD10



With the help of the LEDs on the front side, it is possible to indicate the existence of the two main supply voltages. Furthermore, the current mode of operation of the module can be viewed here (STS1 and STS2).

For this, see "Status diagram" on page 75 and Device control" on page 87.

In the Operation Enable state, the two related LEDs (STS1 and STS2) are used to show the current motor current in the negative and positive direction. Concerning this, the maximum brightness corresponds to the peak current.

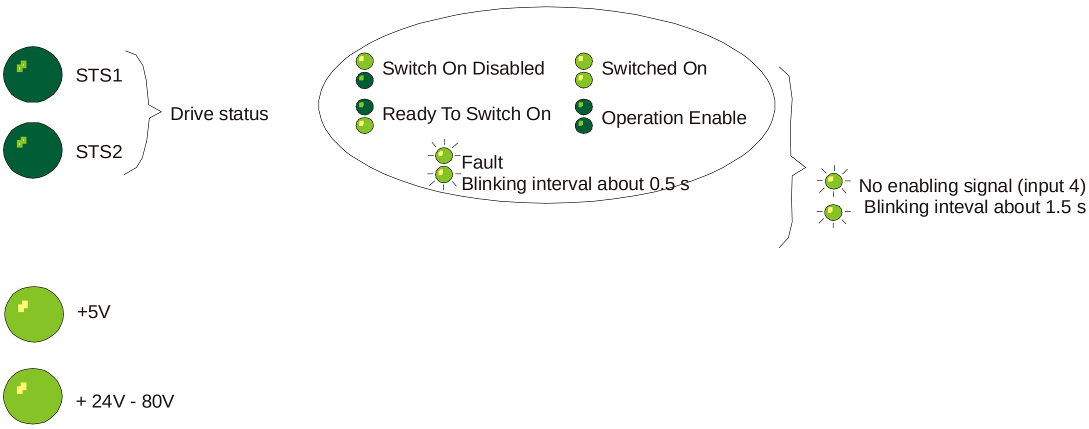
In case of a fault (Fault state), the two-digit error code of the current fault is given via a blink code.

1st digit: number of blink impulses of both state LEDs.

2nd digit: number of blink impulses of LED STS2.

See "EMCY" on page 74 for the description of the error code.

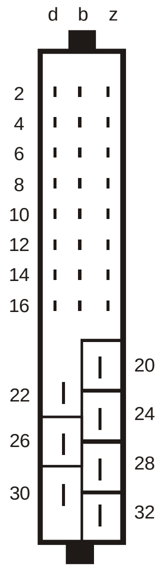
In case of a missing release signal, all three LEDs (STS1, STS2 and CAN) blink about once every 1.5 seconds.



Hardware description UVE8112

Connector UVE8112

Frame connector on the rear side of the module



	D	b	Z
2	← Output 2	← Output 1	→ +24 V (for outputs)
4	→ Input 2	→ Input 1	24 V GND
6	→ Input 4	→ Input 3	Output 3
8	↔ CAN Low	↔ CAN High	CAN GND
10	→ Analogue GND	→ Analogue input +	→ Analogue input -
12	← Encoder voltage 5 V	-	Encoder GND
14	→ Encoder /Z	→ Encoder /B	→ Encoder /A
16	→ Encoder Z	→ Encoder B	→ Encoder A
20			→ VSS_W (Processor section: 18-30 V)
22	→ VSS_M (Power section: 40-95 V)		
24			← Motor output A
26	← Motor output B		
28			-
30	GND_W (Processor section)		
32			GND_M (Power section)

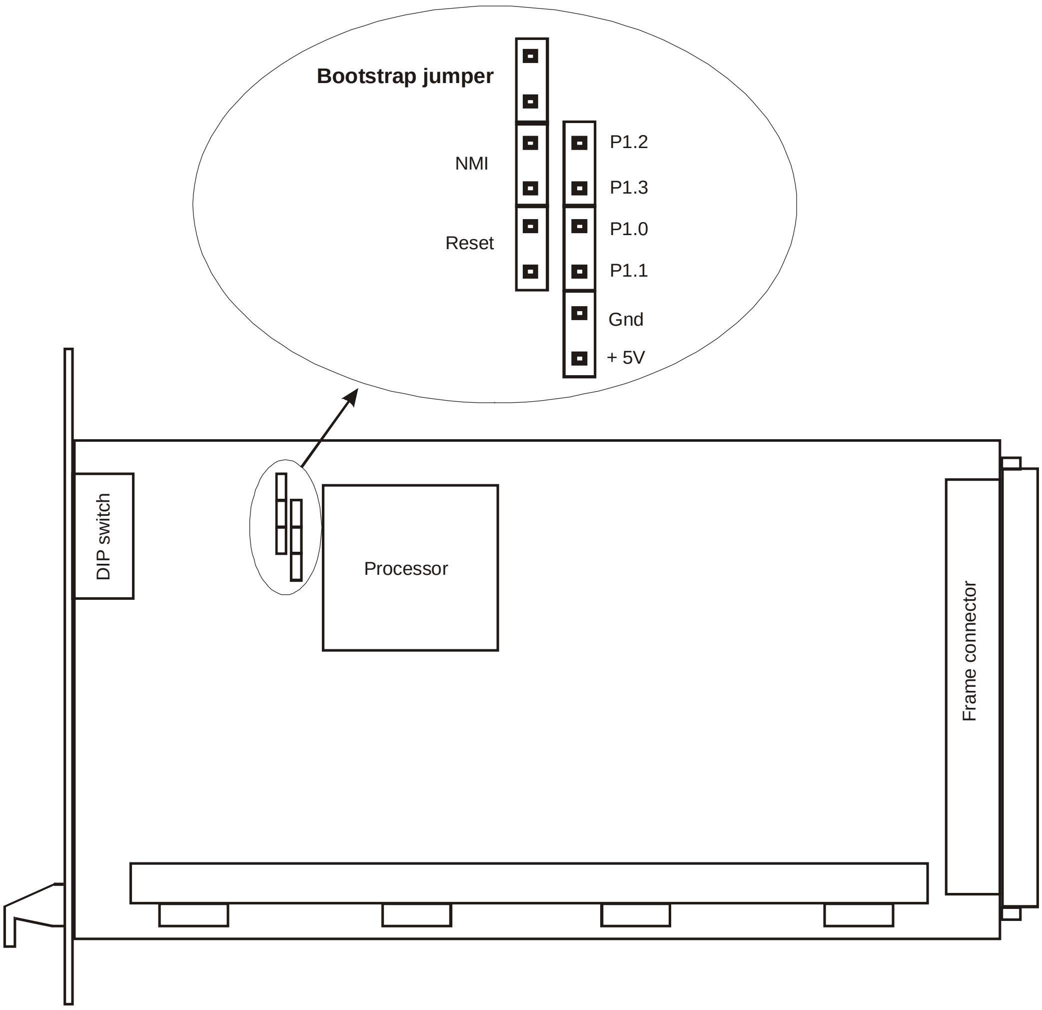
Sub-D9 plug for RS-232

Pin	1	2	3	4	5	6	7	8	9
Signal	-	RxD	TxD	-	GND	-	-	-	-

Jumper settings

For the normal operation, no settings via jumpers are required. Only for the loading process of a new software version by means of the bootstrap loader, the corresponding jumper has to be connected.

See also "The Firmware Update / via Bootstrap Loader command" on page 34.

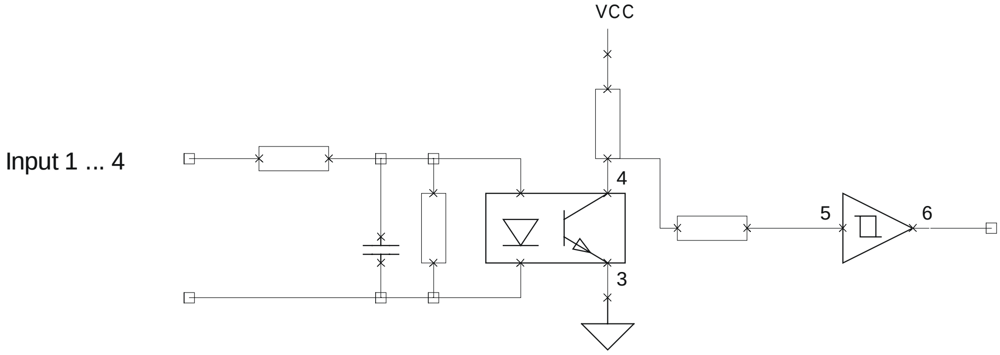


Wiring of the inputs and outputs

Digital inputs 1 to 4

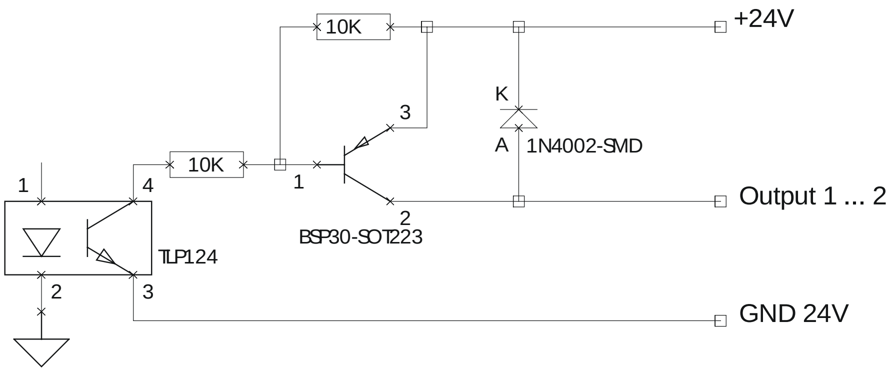
The evaluation of the inputs can be set by means of software.

See “Inputs dialogue box” on page 63.



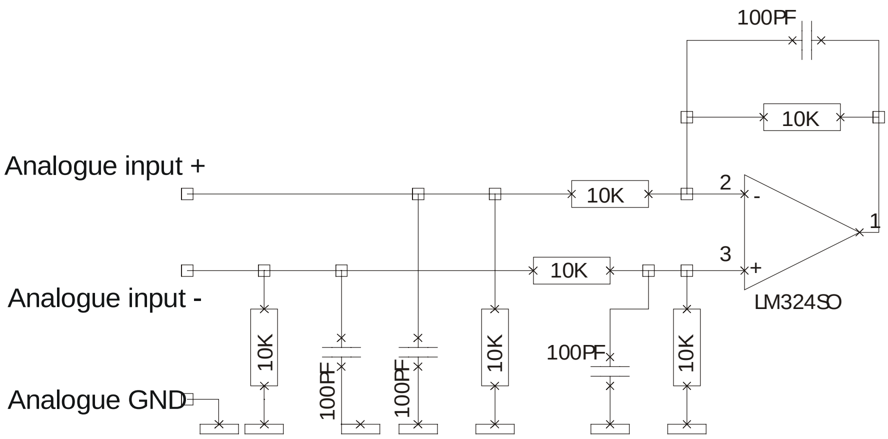
Digital outputs 1 to 3

- Output 1 indicates the operational readiness of the module and is set only, if the module software works correctly (monitoring through internal watchdog timer).
- Output 2 can be used to shunt the limit switch in the safety circuit. It is set during the reference run (on the limit switch).
- Output 3 is used for the brake control. The way in which the brake is controlled can be defined by means of the software.



Analogue input

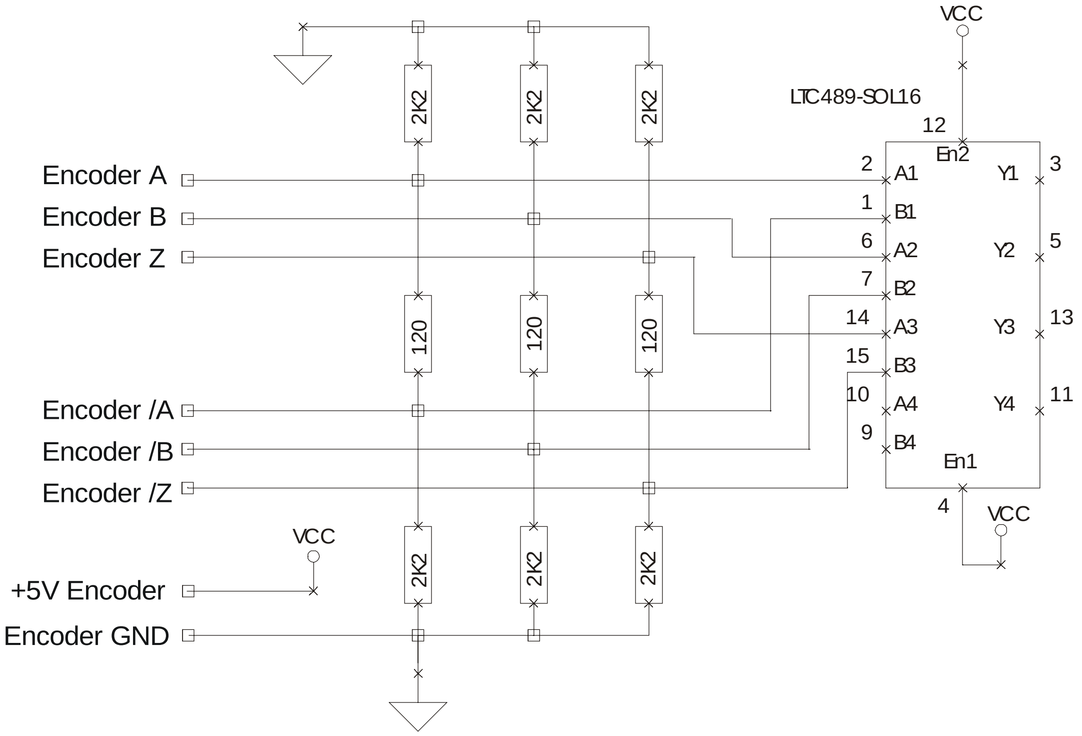
The analogue input is preset for a voltage level in the area of -10 V … +10 V. The signal can either be connected to the positive input (+) or to the inverting input (-). The reference potential is always Analogue GND.



Encoder connections

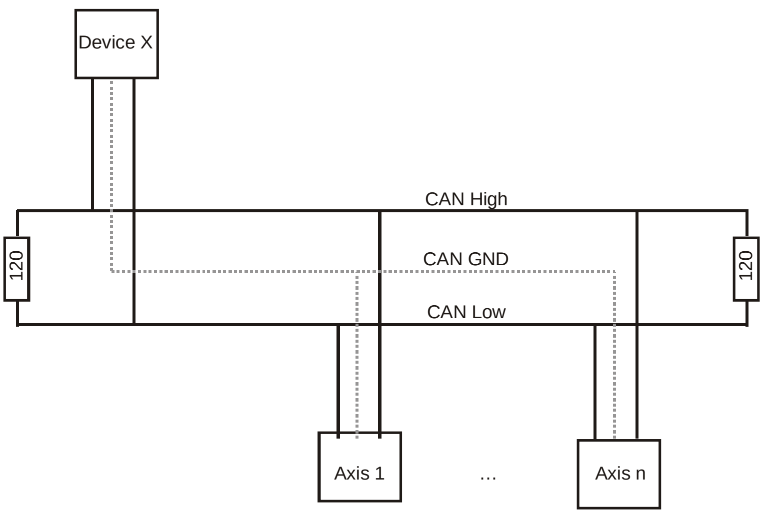
The drive module is preset for the connection of a quadrature incremental encoder with index signal. The transfer of the signals is carried out according to the RS422 specification. It is recommended to use shielded cables twisted in pairs for the encoder wiring.

The voltage supply of the encoder (5 volt) is provided by the drive module. The maximum supply current amounts to 100 mA.



CAN

The structure of the Can network is to be realised so as to guarantee that a 120 ohm terminating resistor is provided on both sides. The drive module itself does not possess a terminating resistor. The stub lines leading from the bus to the individual modules should not be longer than 50 cm in case of a baud rate of 1 Mbit/s.

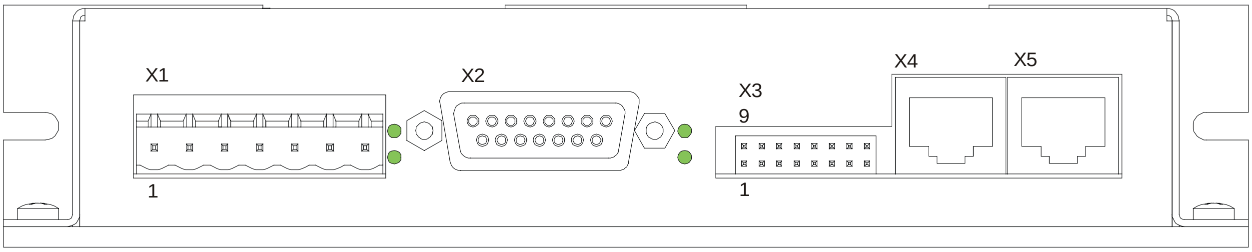


For the CAN wiring, it is recommended to use twisted shielded cables with a characteristic impedance of 108 to 132 ohms. In case of very small network extensions, it may be possible to dispense with the connection of the reference potential (CAN ground) (not recommended).

Hardware description IMD10

Connectors

All connectors are provided on the front of the drive module.



Frame connector X1 – Connection of the operating voltage 24 V-80 V, motor and brake

Pin	Signal
1	GND_M (power section)
2	→ VSS_M (power section: 40–95 V)
3	← motor output A
4	← motor output B
5	ground
6	← brake
7	GND_24V

Frame connector X2 – 15-pin Sub-D – Connection of the encoder, limit switch and brake

Pin	Signal
1	not connected
2	← encoder voltage: 5 V
3	→ encoder /Z
4	→ encoder /B
5	→ encoder /A
6	+24 V
7	→ limit switch 1
8	GND_24V
9	not connected
10	encoder GND
11	→ encoder Z
12	→ encoder B
13	→ encoder A
14	→ reference switch
15	→ limit switch 2

Frame connector X3 – 20-pin system connection

Attention technical modification: This frame connector is expanded from 16 to 20 pins. The plug connection is not valid for the 16 pin connector. The standstill monitoring is added (see object index 0x6510 subindex B)

Pin	Signal
1	Stop 1 IN: standstill monitoring channel 1 input
2	Stop 2 IN: standstill monitoring channel 2 input
3	→ analogue input +
4	digital GND
5	→ Enable input
6	→ Ready In input
7	← Homing output
8	→ limit switch chain input In
9	+24 V
10	GND_24 V
11	Stop 1 OUT: standstill monitoring channel 1 output
12	Stop 2 OUT: standstill monitoring channel 2 output
13	→ analogue input -
14	→ analogue GND
15	→ Enable input
16	← Ready Out output
17	← Homing output
18	← limit switch chain output Out
19	+24 V
20	GND_24 V

Frame connector X4 – RJ45 - RS-232 and CAN IN

Pin	1	2	3	4	5	6	7	8
Signal	RS232 TxD	RS232 RxD	RS232 GND	↔ CAN Low	↔ CAN High	CAN GND	-	-

Frame connector X5 – RJ45 – CAN OUT

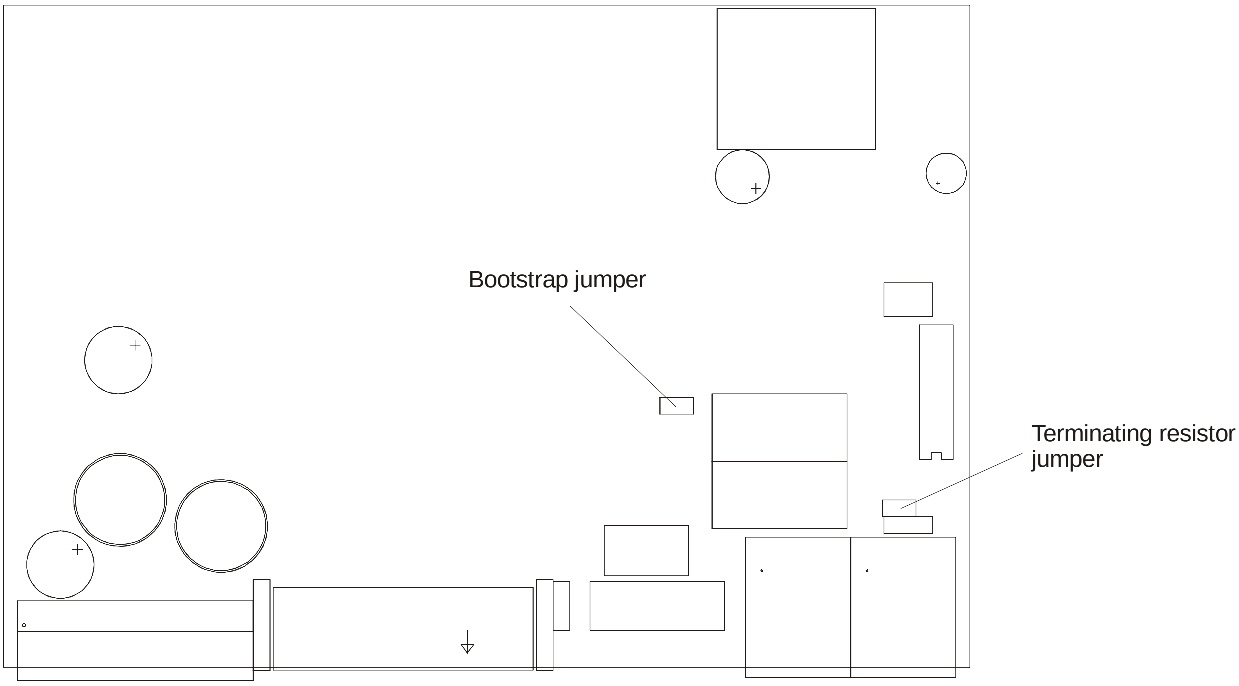
Pin	1	2	3	4	5	6	7	8
Signal	-	-	-	↔ CAN Low	↔ CAN High	CAN GND	-	-

Jumper settings

For the normal operation, no settings via jumpers are required. Only for the loading process of a new software version by means of the bootstrap loader, the bootstrap jumper has to be connected.

See also "The Firmware Update / via Bootstrap Loader command" on page 34.

Every drive module comprises a 120 ohm terminating resistor, which is activated through the connection of the terminating resistor jumper. See also “Wiring of the inputs and outputs – CAN“ on page.

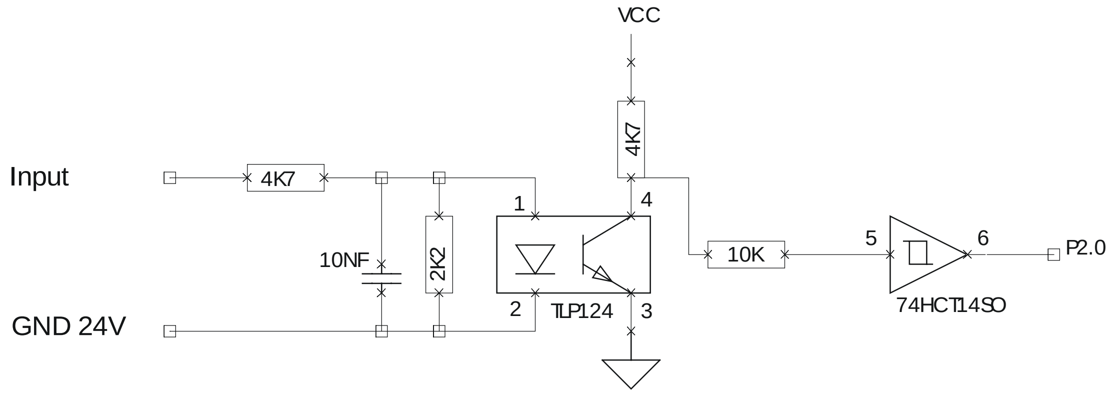


Wiring of the inputs and outputs

Digital inputs, limit switches, reference switches, Enable

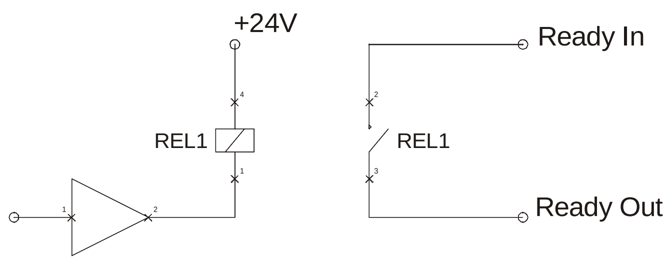
The evaluation of the inputs can be set by means of software.

See “Inputs dialogue box” on page 63.



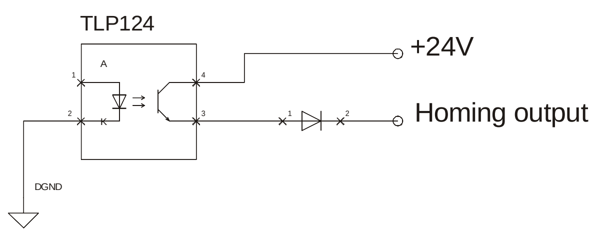
Digital Ready output

The Ready output is designed as a potential-free contact. It indicates the operational readiness of the module and is only set, if the module software works correctly (monitoring through internal watchdog timer). The design as a potential-free contact makes it possible to easily link several Ready outputs. This sum signal can be processed e.g. in the safety circuit.



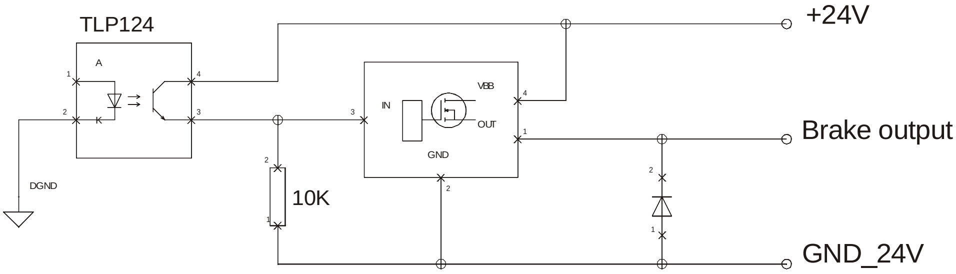
Digital Homing output

The Homing output can be used to shunt the limit switch in the safety circuit. It is set during the reference run (on limit switch).



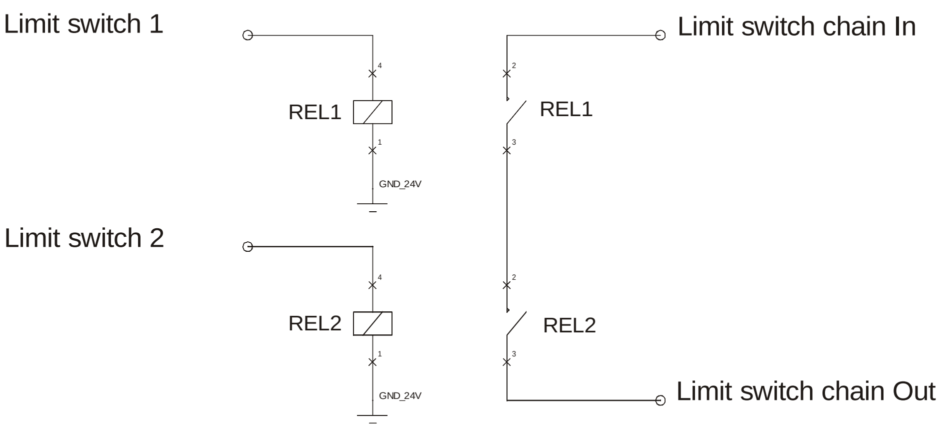
Digital Brake output

The Brake output is used for the brake control. The way in which the brake is controlled can be defined by means of the software.



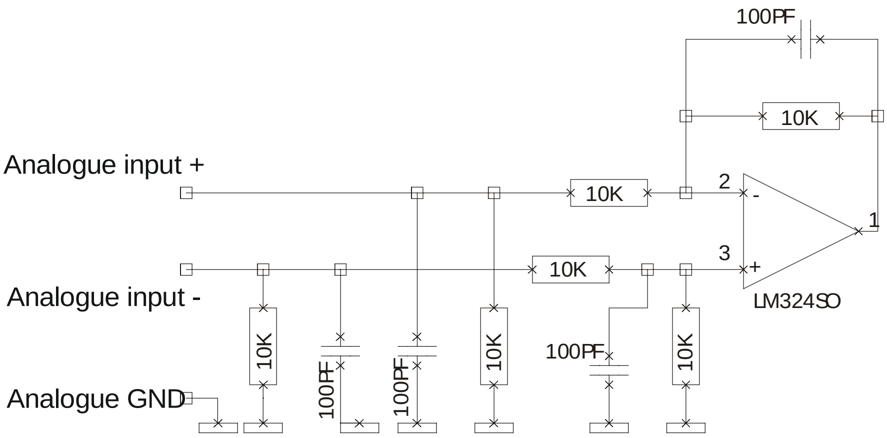
Digital Limit Switch Chain output

The Limit Switch Chain output is designed as a potential-free contact. It indicates, if both limit switches are not activated (closed potential-free contact). If one or both limit switch(es) is (are) activated, the contact is open. In the safety circuit, this signal can be used to monitor the limit switch.



Analogue input

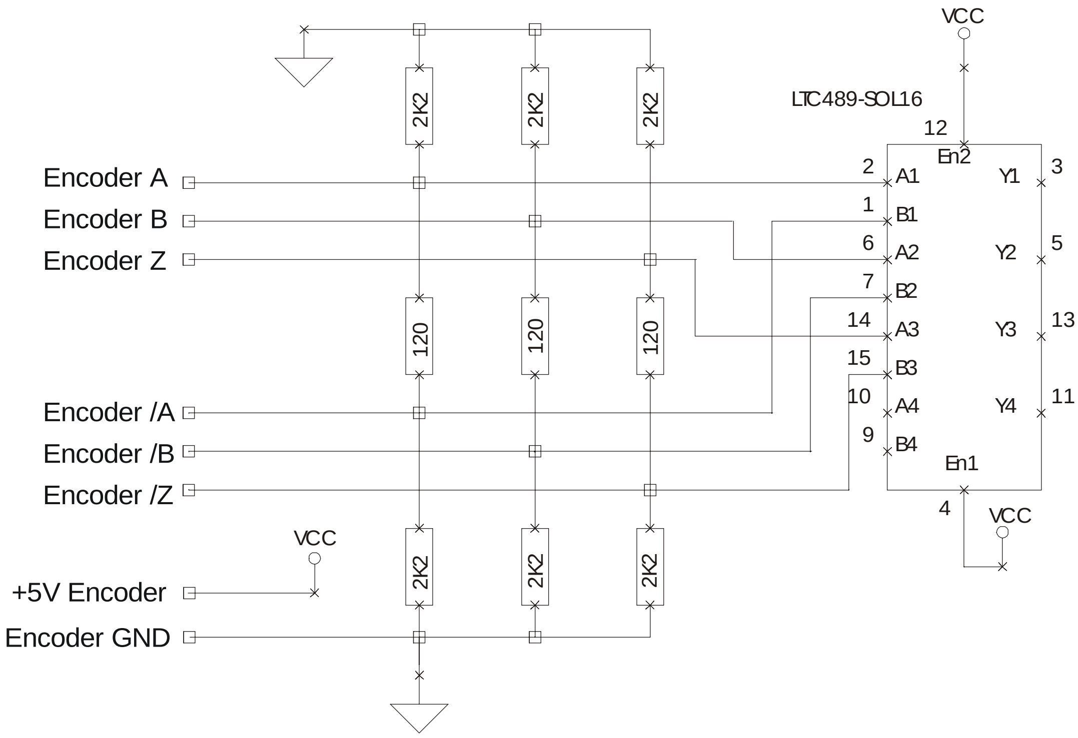
The analogue input is preset for a voltage level in the area of -10 V … +10 V. The signal can either be connected to the positive input (+) or to the inverting signal (-). The reference potential is always Analogue GND.



Encoder connections

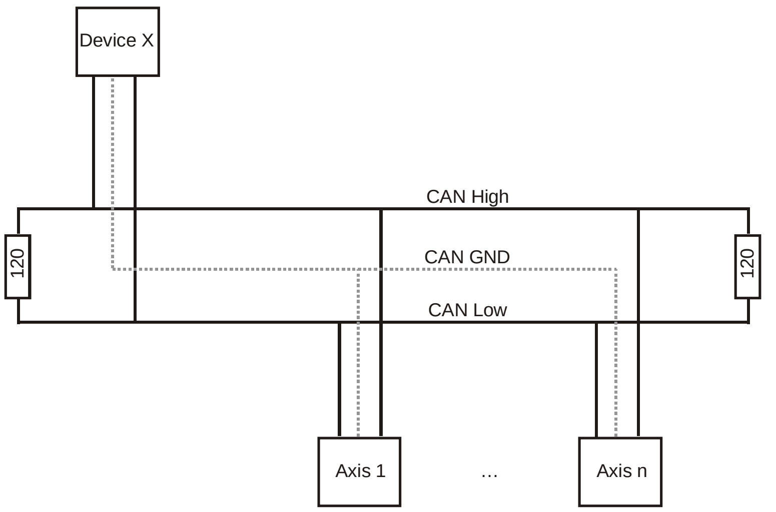
The drive module is preset for the connection of a quadrature incremental encoder with index signal. The transfer of the signals is carried out according to the RS422 specification. It is recommended to use shielded cables twisted in pairs for the encoder wiring.

The voltage supply of the encoder (5 volt) is provided by the drive module. The maximum supply current amounts to 100 mA.



CAN

The structure of the Can network is to be realised so as to guarantee that a 120 ohm terminating resistor is provided on both sides. The drive module itself does not possess a terminating resistor. The stub lines leading from the bus to the individual modules should not be longer than 50 cm in case of a baud rate of 1 Mbit/s.



For the CAN wiring, it is recommended to use twisted shielded cables with a characteristic impedance of 108 to 132 ohms. In case of very small network extensions, it may be possible to dispense with the connection of the reference potential (CAN ground) (not recommended).

Data transfer

The data exchange with the drive module can be realised via the two following interfaces:

- serial interface RS232
- Can bus

The RS232 interface can be found on the front side of the module; the Can lines are led via the rear connector.

Both interfaces can be used to access all objects (parameters) of the module. The access rights depend on the communication channel which was set by means of the DIP switch on the front side.

Communication channel	Switch position		Access right	
CAN	S7	0	Can:	Read and Write
	S8	0	RS232:	Read only
RS232	S7	1	Can:	Read only
	S8	0	RS232:	Read and Write
Analogue input	S7	X	Can, RS232:	Read and Write ("Control word" and "Modes of operation" cannot be written)
	S8	1		

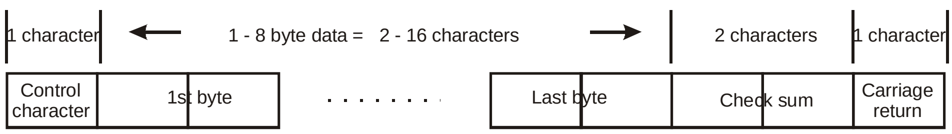
In case of a communication via Can, the CanOpen protocol is used (see last chapter).

For the serial communication (asynchronous, 1 start bit, 8 data bits, 1 stop bit and no parity bit), parts of this protocol are used, i.e. the SDO transfer, the emergency message and the boot-up message.

In case of the SDO, telegrams of a length of up to 8 bytes are exchanged between two participants (see SDO description on page 70). These 8 bytes are transferred in hexadecimal format as ASCII characters, framed by an identifier and a checksum. Each message is terminated with a carriage return (CR, 0x0D). Contrary to SDO, not the entire 8 bytes have to be transferred, if the data contained are shorter than 4 bytes.

Likewise, the boot-up message and the emergency messages in case of a fault is sent from the module via the serial interface.

See "Monitor window for serial interface" on page 37.



Control character	Meaning	
C	SDO query (is received by the module)	
B	SDO response (is sent by the module)	
E	Boot-up message	
1	Emergency telegram	
F	Error during transfer	F1: received too few characters
	F2: checksum error	F3: received too many characters
	F4: error during the conversion of the characters	

The checksum is formed on the basis of the 1 ... n data bytes (not the ASCII characters). It has to be taken into account that the control character is of no importance for the forming of the checksum. This only concerns the “pure” data bytes that are to be transferred. The checksum also has to be converted into ASCII characters again, before it is sent. This transfer protocol is applicable for both directions of transfer, i.e. both for the reading and the writing process.

Conversion of the data bytes into ASCII signs: e.g. 0x3F = '3' and 'F' (0x33 and 0x46).

Checksum = - ((Byte1 + Byte2 + ... + Byte_n) mod 256) or

= - (Division remainder of ((Byte1 + Byte2 + ... + Byte_n) : 256))

In the following, example 2 is used to show why a checksum of 0x10 (hexadecimal number) is created.

C *2F 60 60 00 01* ***10*** **CR**

The control character ‚C’ and Carriage-Return ‚CR’ are not taken into account for the calculation.

Checksum	=	- (0x2F + 0x60 + 0x60 + 0x00 + 0x01)	(hexadecimal number)
	=	- (47 + 96 + 96 + 0 + 1)	(decimal number)
	=	- 240	(decimal number)
	=	- Division remainder of (240 : 256)	(decimal number)
	=	-240	(decimal number)
	=	0xFF10	(hexadecimal number)
	=	0x10	(hexadecimal number)

The two characters 1 and 0 for the last byte 0x10 of 0xFF10 are transferred as the checksum before the carriage return.

Commissioning

Overview

An essential condition for the successful commissioning is the correct pin assignment of all required signals and connections.

The DIP switch has to be configured according to the desired mode of operation and communication interface, before the module is switched on.

An auxiliary means for the commissioning is the supplied DCSetup programme. It contains a special menu point allowing a step-by-step commissioning of the module.

See "Step-by-step commissioning" on page 41.

After the completion of the step-by-step commissioning, all drive- and motor-specific parameters should be set. If additional settings have to be carried out, e.g. the configuration of different communication parameters or device parameters which are not changed through the step-by-step commissioning, this can also be done by means of the setting programme.

See "Object Directory dialogue box" on page 65.

Particularities as regards the commissioning of a gantry axis

In the gantry operation, two axes are synchronously controlled without a toothed belt or a bevel gear system. One axis works as the master and the other works as the slave. In order to commission a gantry axis, each axis (master or slave) has to be commissioned separately, before the machine is assembled. These parameters are stored as usual in the UVE8112 module. On the basis of the control and motion parameters of the master and slave axis, the ISEL CNC control determines the optimum motion behaviour for the gantry axis during the initialisation phase.

By means of the CANSET programme, it can be stipulated, if an axis is in the gantry operation or not and which axis is the master and which the slave. In case of a slave axis, the user also has to define the maximum admissible deviation of position and the coupling factor between the master and the slave during the gantry operation in the “Can interpolation” dialogue box. The reference point is the start of the synchronous control. Every axis (master or slave) has its own reference switch or a limit switch serving as reference switch. In order to compensate for the tolerance during the installation of the switch, users may define different reference distances for the master and the slave (see “Reference run” dialogue box).

The following settings have to be identical for the master and the slave:

- axis directions (see “Direction” dialogue box)
- axis type: Linear and Rotary Axis (see “Can interpolation” dialogue box)
- motion parameter unit µm or “...” (see “Transmission” dialogue box)

All other parameters may differ for the master and slave.

Examples for the motion control

Example 1:

Communication:	Can
Baud rate:	1 Mbit/s (RS232: 57600 baud)
Node address:	2
Mode of operation:	Velocity control with ramp profile (profile velocity mode)

Set the "Modes of operation" (6060_h) parameter via SDO to the value 3.

Send		**Receive**	
ID	Data	ID	Data
602	2F 60 60 00 03 00 00 00	582	60 60 60 00 00 00 00 00

Set the "Control word" (6040_h) parameter via SDO successively to the values 0x06, 0x07, 0x0F. (Activate the “Operation Enable” status.)

Send		**Receive**	
ID	Data	ID	Data
602	2B 40 60 00 06 00 00 00	582	60 40 60 00 00 00 00 00
602	2B 40 60 00 07 00 00 00	582	60 40 60 00 00 00 00 00
602	2B 40 60 00 0F 00 00 00	582	60 40 60 00 00 00 00 00

The "Profile acceleration" (6083_h) parameter is used to set the desired acceleration, e.g. 100000 (0x000186A0).

Send		**Receive**	
ID	Data	ID	Data
602	23 83 60 00 A0 86 01 00	582	60 83 60 00 00 00 00 00

Set the "Target velocity" ($60FF_h$) parameter, e.g. 50000 (0x0000C350). The motion is launched.

Send		**Receive**	
ID	Data	ID	Data
602	23 FF 60 00 50 C3 00 00	582	60 FF 60 00 00 00 00 00

Example 2:

Communication:	RS-232
Baud rate:	19200 baud (Can: 20 kBit/s)
Node address:	7 (is only of importance for optional Can accesses)
Mode of operation:	Positioning (profile position mode)

Set the "Modes of operation" (6060_h) via RS232 to the value 1.

Send	**Data**
Data (ASCII characters)	Data
C 2F 60 60 00 01 10 $_{CR}$	B 60 60 60 00 E0 $_{CR}$

Set the "Control word" (6040_h) parameter successively to the values 0x06, 0x07, 0x0F. (Activate the "Operation Enable" status.)

Send	**Receive**
Data (ASCII characters)	Data
C 2B 40 60 00 06 00 2F $_{CR}$	B 60 40 60 00 00 $_{CR}$
C 2B 40 60 00 07 00 2E $_{CR}$	B 60 40 60 00 00 $_{CR}$
C 2B 40 60 00 0F 00 26 $_{CR}$	B 60 40 60 00 00 $_{CR}$

The "Profile acceleration" (6083_h) parameter is used to set the desired acceleration, e.g. 100000 (0x000186A0).

Send	**Receive**
Data (ASCII characters)	Data
C 23 83 60 00 A0 86 01 00 D3 $_{CR}$	B 60 83 60 00 BD $_{CR}$

Set the "Profile velocity" (6081_h) speed, e.g. 300000 (0x000493E0).

Send	**Receive**
Data (ASCII characters)	Data
C 23 81 60 00 80 38 01 00 43 $_{CR}$	B 60 81 60 00 BF $_{CR}$

Set the “Target position” ($607A_h$), e.g. 300000 (0x000493E0).

Send	**Receive**
Data (ASCII characters)	Data
C 23 7A 60 00 E0 93 04 00 8C $_{CR}$	B 60 7A 60 00 C6 $_{CR}$

Launch the absolute motion by setting Bit 4 in Control Word (0x001F).

Send	**Receive**
Data (ASCII characters)	Data
C 2B 40 60 00 1F 00 16 CR	B 60 40 60 00 00 CR

Query of the status word (6041_h)

Send	**Receive**
Data (ASCII characters)	Data
C 40 41 60 00 1F CR	B 4B 41 60 00 27 12 DB CR

Example 3:

Communication: Analogue input

Baud rate: (Can: 1 Mbit/s, RS232: 57600 baud)

Node address: 3

Mode of operation: Velocity control with analogue input (±10 V)

Set the control range by configuring the maximum speed with the help of the DcSetup programme.

Activate the “Operation Enable” status via the enabling signal (dig. input 4).
Velocity target via analogue signal (±10 V corresponds to ± set maximum speed).

Fault states

If an error occurs, the module switches to the fault state, in which the final stage is switched off. The LEDs on the front side indicate the current error by a blinking sequence.

See EMCY on page 74.

Overview of possible fault states:

Internal error code	EEC	MEC	LED display(1)	Description
1	0x6100	0x00xx	12	Internal software error
2	0x2320		13	Short circuit
3	0x4210		14	Overtemperature
6	0x7305		17	Fault Encoder Track A
7	0x7306		18	Fault Encoder Track B
8	0x7307		19	Fault Encoder Track Z
9	0x8100	0x0002	21	CAN error
10	0x7308		22	Deviation of position between the master and the slave is greater than the maximum admissible value.
11	0x8120		23	Node guarding failed
12	0x5441		24	Negative limit switch active
13	0x5442		25	Positive limit switch active
14	0x5444		26	No enabling signal
16	0x7309		28	Fault in the slave axis
17	0x730A		29	Fault in the master axis
18	0x6010		31	Watchdog reset
19	0x3100		32	No main supply voltage
20	0x5114		33	No programming voltage
21	0x5530		34	Error during the deletion / writing of the Flash memory
23	0xFF10		36	Synchronisation error
26	0x2330		39	I^2t current limitation active

(1) 1^{st} digit = number of blinking impulses of both LEDs, 2^{nd} digit = number of blinking impulses of one LED.

DcSetup programme

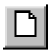
Menus

Commands of the File menu

The New command (File menu)

Use this command to create a new DCF file with default parameters.

Shortcuts

Tool bar: 

Keyboard: CTRL+N

The Open command (File menu)

Use this command to open an existing DCF file. If the online mode is enabled, the parameters of the module have to be read in once again.

Shortcuts

Tool bar: 

Keyboard: CTRL+O

The Save command (File menu)

Use this command in order save the active set of parameters in the DCF format under the current name and in the current directory. If the document is saved for the first time, the DcSetup programme indicates the Save As dialogue box in which you can name your set of parameters. If you would like to rename an existing document or move it to a different directory, you can also choose the Save As command.

Shortcuts

Tool bar: 

Keyboard: CTRL+S

The Save As command (File menu)

Use this command in order to save and name the active set of parameters. The programme opens the Save As dialogue box in order for you to enter the name for your set of parameters.

The 1, 2, 3, 4 commands (File menu)

Use the numbers and file names given at the end of the File menu in order to open the last four files you closed. For this open the number of the file to be opened.

The Exit command (File menu)

Use this command in order to exit your DcSetup session. Alternatively, you can choose the Close command from the application’s system menu. The programme asks, if you would like to save the current set of parameters with the unsaved changes.

Shortcuts

Mouse: System menu

Keyboard: ALT+F4

Commands of the Connection menu

The Online Mode On/Off command

Use this command in order to enable or disable the online mode (see “Online operation” on page 39). In the online mode, all parameters have the same values in the DcSetup programme and in the drive module.

When the online mode is enabled, all parameters of the module are read in and then compared to the values of the current set of parameters in the setting programme. In case of a mismatch, you can choose, whether the module values or the values from the setting programme are to be used as current parameters.

Shortcuts

Tool bar: 

The Active Connection / CAN command

By means of this command you can configure the DcSetup programme so that the CAN bus is to be used as the active connection to the drive module, i.e. all parameters are read and written via CAN.

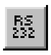
Shortcuts

Tool bar: 

The Active Connection / RS232 command

By means of this command, you can configure the DcSetup programme so that the serial interface is to be used as the active connection to the drive module, i.e. all parameters are read and written via RS232.

Shortcuts

Tool bar: 

The RS232 Settings command

This command opens the RS232 configuration. Here, you can choose a serial interface (COM1 - COM4) and set the baud rate.

The CAN Settings command

This command opens the CAN Configuration dialogue box. Here, the CAN driver and the node number of the desired drive module can be activated or set, respectively.

Commands of the Commissioning menu

The Step-by-Step Commissioning command

This command launches a wizard for the step-by-step parameterisation of the drive module. For this, the dialogue boxes for the setting of the drive and control parameters are run through in the given order. In this way, a save commissioning is guaranteed.

All setting dialogue boxes can be directly accessed with the help of the random order command.

Shortcuts

Tool bar: 

The Random Order command

With this command, all setting dialogue boxes (e.g. current controller, mode of operation ...) are displayed as tabs in the Commissioning dialogue box.

Shortcuts

Tool bar: 

The Reset command

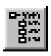
This command launches a reset of the drive module. The reset corresponds to the switch-off and subsequent switch-on of the module. All parameters are loaded with the last values saved.

Commands of the Settings menu

The Object Directory command

Use this command in order to access the Object Directory dialogue box. Via the object directory, you can access all drive module parameters. The parameters here are sorted according to an index and displayed with their CanOpen characteristics.

Shortcuts

Tool bar: 

Commands of the Tools menu

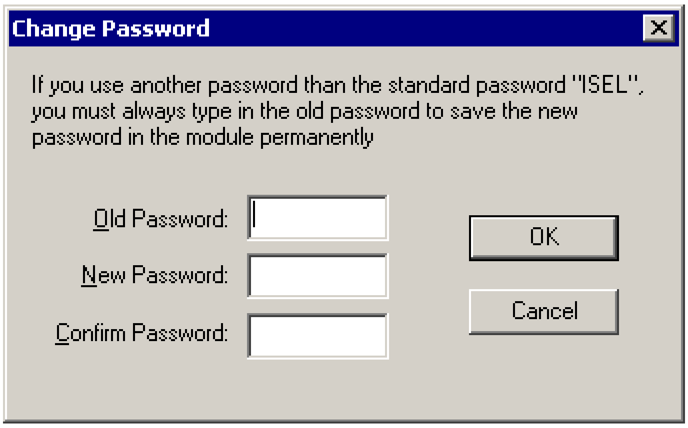
The Extended Functions command

By means of this command, you can switch the menu bar to a simplified view giving the most important functions or to an extended view with the possibility to access all existing commands.

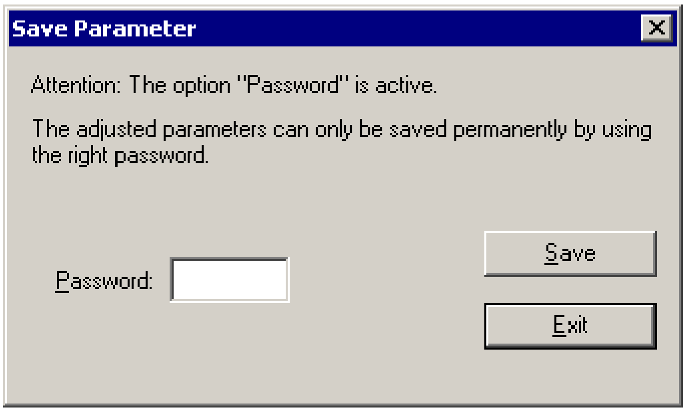
During the first start of the programme, only the simplified menu bar is displayed in order to facilitate the commissioning process.

Password

With the allocation of a password, the DCSetup programme allows users to protect the parameters in the drive module from unauthorised access. After a firmware update via the bootstrap loader has been realised, the module has the default password “ISEL”. Via the menu Tools\Password, it is possible to define a new password at any given point in time.



As long as the default password is valid, the users do not notice during the application of the DcSetup programme that a Password is required. If the password differs from the default password, however, users are required to enter the right password in order to permanently store the parameters in the module. It has to be taken into account that the module can be commissioned as usual. It is just not possible to permanently store the parameters, i.e. the set values are lost, when the module is switched off.



The Firmware Update / Normal command

Use this command in order to load a new firmware version into the module. The Hex file (e.g. DCCON.HEX) with the new programme version must be in the same directory as the DcSetup programme.

The update can only be realised via the serial interface.

All current settings of the drive module parameters are kept during an update. The update process must not be interrupted (e.g. by switching off the module, interrupting the connection, etc.).

If the update process could not be completed successfully, the drive module is not operational. In this case, a new update has to be realised with the help of the Firmware Update / via Bootstrap Loader command.

See “Firmware-Update” on page 67.

The Firmware Update / via Bootstrap Loader command

Use this command in order to load a new software application into the drive module, if there is no executable programme.

For this, it is necessary to install the bootstrap jumper on the module (see “Jumper setting” on page 20). After the module is switched on again, the update process can be launched. After a successful update, this jumper has to be removed again.

Existing parameter values of the drive module are overwritten with default values.

See “Firmware-Update” on page 67.

Commands of the View menu

The Tool Bar command

By means of this command, you can show or hide the tool bar. The tool bar contains shortcuts for certain menu commands.

The Status Bar command

Use this command in order to show or hide the status bar. The status bar gives information on the programme or the selected menu.

The Serial Interface command

This command switches on or off the monitor window for the serial interface RS232. In this window, all characters of the active serial connection that were sent or received are displayed.

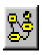
The CAN Monitor command

This command switches on or off the monitor window for the CAN transfer. In this window, all signals (sent and received) are displayed that are transferred between the DcSetup programme and the active drive module.

The Drive Status command

Use this command in order to display the status window for the current operating state of the drive module according to CANOpen / DS402.

Shortcuts

Tool bar: 

The Language command

With this command you can select the language of DCSetup. At the moment English and German are supported.

Commands of the ? menu

The Help Topics command

With this command, you receive an overview of the existing topics of the online help and can consult the desired topic.

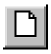
The Info command

This command shows a dialogue box with the current programme version.

Programme user interface

Tool bar

The tool bar offers shortcuts for frequently used menu commands:

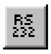
 Opens a new parameter file

 Opens an existing parameter file

 Saves the open parameter file

 Makes a connection to the drive module

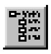
 Chooses CAN as active connection to the module

 Chooses the serial interface as active connection

 Indicates the operating state of the module

 Launches the commissioning wizard

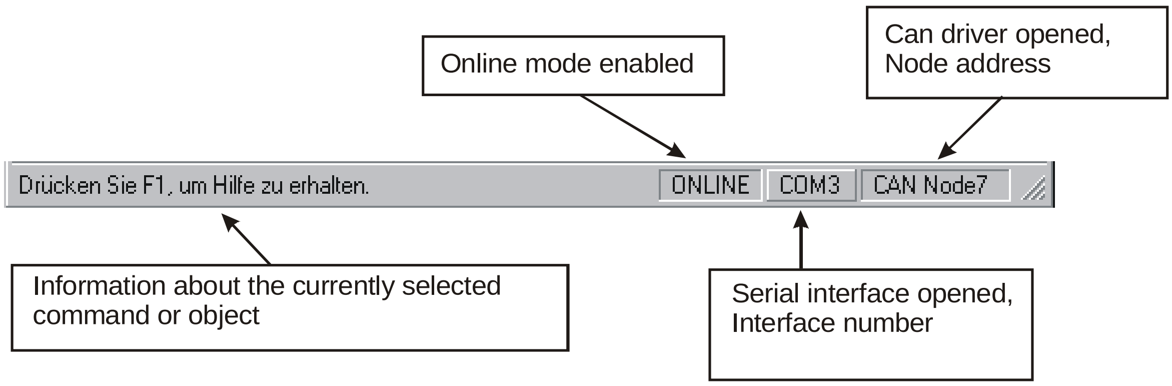
 Shows all setting dialogue boxes as tabs

 Displays a tree diagram with all module parameters (objects)

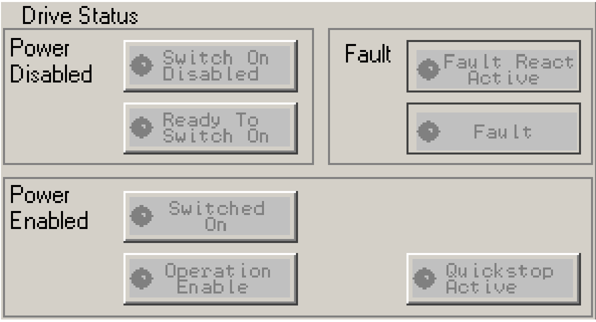
 Launches the direct help

Status bar

The status bar gives information on the programme and on the opened interfaces and shows the current state of the connection to the drive module.



Drive status



This display can be used to consult and change the operating state of the drive module in case of an active online mode. The status is queried about 5 times every second via the active connection.

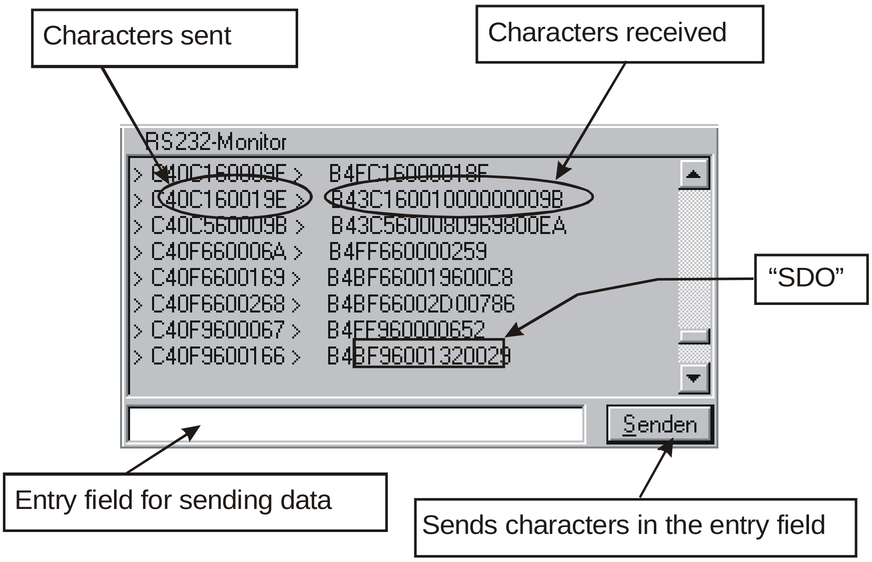
The individual states and transfers are defined in the CANOpen DS402 specification for drives. See “Device control” on page 87. The states are controlled via the “Control word” parameter (CANOpen object 6040h).

A light green LED indicates the current status. Dark green LEDs show the possible transitions to other operating states. A switch-over to the states illustrated in grey is not possible from the current state.

If, on the DIP switch, the “Analogue input” mode of operation is set (current or velocity controller), there are only the two possible states: “Ready To Switch On” and “Operation Enable” that are controlled by the enabling signal. A change via this dialogue box (or the “Control word” is thus not possible).

With the help of the View / Drive status command, the window can be faded in or out. If the window is faded out, the status query is interrupted.

Monitor window for serial interface

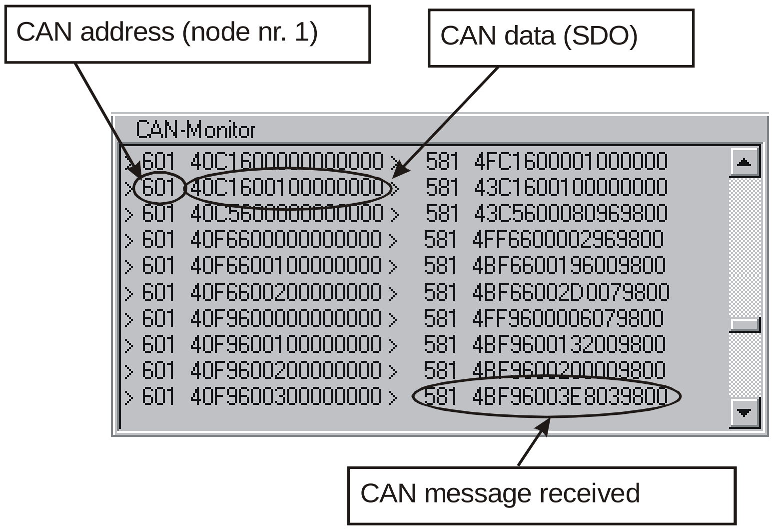


Here, all data are shown that are transferred via the serial interface (RS232). The characters sent are marked with two arrows by the DcSetup programme. Characters received are written behind this.

The transfer format of the serial interface was stipulated by analogy with the SDO transfer of CANOpen. The content of an SDO message ((max.) 8 bytes) is transferred in the Hex format as ASCII characters. An identifier is prefixed and two characters with a checksum for the data backup are suffixed. See also “Data transfer” on page 24.

For test purposes, different data can be sent to the module via the entry field and the <Send> button.

Monitor window for CAN communication



In this window, CAN data are shown which are exchanged between the DcSetup programme and the selected drive module.

For the communication, the CANOpen SDO protocol is used.

The setting of the node number for the connection with the desired module can be realised via the Connection / CAN settings menu command.

Programme functions

Create a data connection

The positioning module UVE8112 can internally store a configurable set of parameters, in which all drive and control settings are included. If no settings have been carried out yet, the parameters contain default values.

Furthermore, the DcSetup administers an active set of parameters that is read in, while the DCF file is opened. During the start of the programme, a default set of parameters is created.

Offline operation

If the online mode is not enabled, the DcSetup programme can be used to either edit a set of parameters from an existing DCF file or to create a new set of parameters. All changes in the setting dialogue boxes affect the internal values only and can then be saved in a file.

Online operation

In the online operation, the change of a parameter is saved both in the DcSetup programme as well as in the module.

For this reason, two sets of parameters have to be aligned with each other, as soon as the online mode is activated. For this, first of all, all parameters of the module are read in. Then, these values can be compared to those of the setting programme. In case of discrepancies, the system asks, in which direction the adjustment is to be carried out, i.e. if the set of parameters of the module is to be copied into the setting programme or vice versa.

In this way, it is e.g. possible to load a finished set of parameters from a file into the module. The loaded parameters can then be permanently stored in the module (e.g. via the Object Directory dialogue box).

For the activation of the online mode, use the Enable / Disable Online Mode menu command. The current state is also given in the status bar.

Before the online mode can be enabled, it might be necessary to first of all set the parameters of the interface and the active connection. See also RS232 / CAN settings.

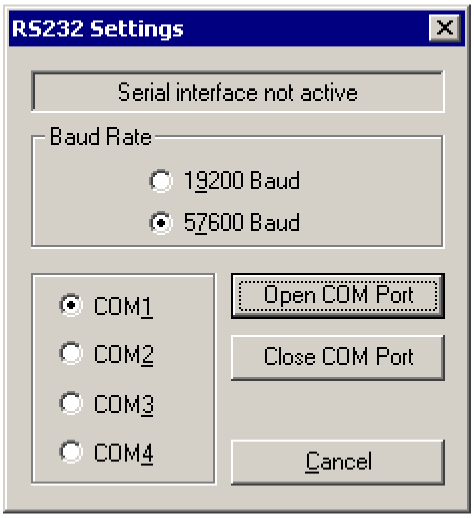
Choose Active Connection

The data communication with the positioning module can be realised either via the serial interface RS232 or via CAN. The Active Connection CAN / RS232 command is used to select a connection, before the online mode is switched on.

The connection parameters on the module and in the DcSetup programme must, of course, be identical (baud rate, COM no.) and the interface or the CAN driver must be open.

RS-232 settings

With the Connection / RS232 Settings command, you open a dialogue box in which you can define the baud rate and the interface to be used. Furthermore, it is shown here, which interface is currently used or active. The current interface is also given in the status bar.

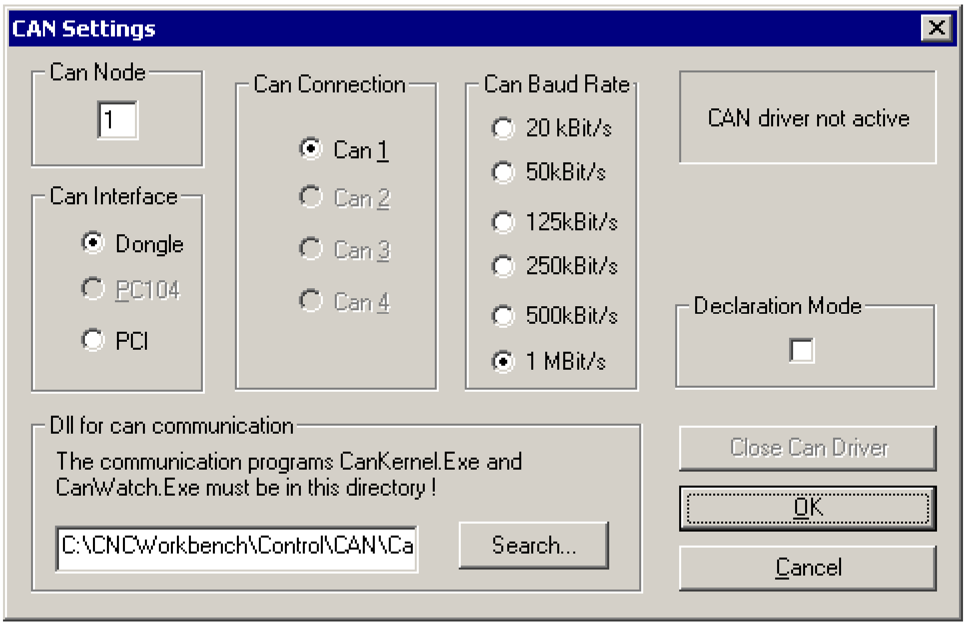


CAN settings

With the Connection / CAN Settings command, you open a dialogue box with which you can set the CAN node number, the baud rate, the Can connection and the Dll functions library for the CAN connection to the drive module. Furthermore, the current status of the CAN driver is indicated here. It has to be taken into account that the Dll functions library **CanApi.Dll** and the communications programme **CanKernel.Exe** as well as the monitoring programme **CanWatch.Exe** must be in the same directory.

If the CAN driver has already been launched from a different application, the DCSetup programme can normally no longer use the Can bus in order to communicate with the UVE8112 / IMD10 module. Only the serial interface is concerned. If you nevertheless want to use DCSetup with the CAN bus, you have to activate the “Declaration mode”. In this case, you can observe the different parameters via the “Settings\Object Directory” menu. The writing of the parameters, however, is not possible. The use of the DCSetup programme in the “Step-by-step commissioning” mode or in the “Random order” mode frequently generates error messages during the active declaration mode, which you can, however, just ignore. The reason for the generation of error messages are the unsuccessful attempts of DCSetup to overwrite different parameters for the purposes of the first initialisation. You should also take into account that the CAN driver has to be activated, before the declaration mode can be used.

The currently set node address and the status of the CAN driver are also indicated in the status bar.



Commissioning / Adjustment of parameters

Basically, all parameters of the drive module can be set via the Object Directory dialogue box. For the configuration of the CAN communications parameter, this is also both necessary and reasonable. However, for the commissioning of the actual drive functions, it would be a very unclear and lengthy procedure to find out the different parameters from the directory and to enter appropriate values.

For this reason, all relevant parameters were sorted in dialogue boxes according to functional groups. For this reason, a comfortable parameterisation is possible. In the Commissioning menu, there are two possibilities to access the setting dialogue boxes:

Step-by-step commissioning

This option should be used for the first commissioning of the drive module. The required setting dialogue boxes (differing according to the mode of operation) are run through in a predefined order. Thus, the highest level of security is ensured and uncontrolled movements of the axis e.g. due to wrongly connected encoder lines and non-adapted control parameters can be avoided to a large extent. The execution of the next step often depends on the successful completion of the current step.

During the first start of the programme, a simplified menu bar is shown which e.g. allows only the step-by-step commissioning. A permanent storing of data after the commissioning can be carried out via the “Object Directory dialogue box” of the Settings\Object Directory main menu.

If the drive module is operated together with the ISEL control card UPMV4/12, the DcSetup programme has to generate an enabling signal so as to ensure that the final stages can be switched on. For this, the path of the INI file belonging to the UPMV4/12 has to be entered in a special dialogue box.

Random order

This option allows to access all existing setting dialogue boxes. These are grouped as tabs in a superordinate dialogue box. The default buttons of this dialogue offer the following functions:

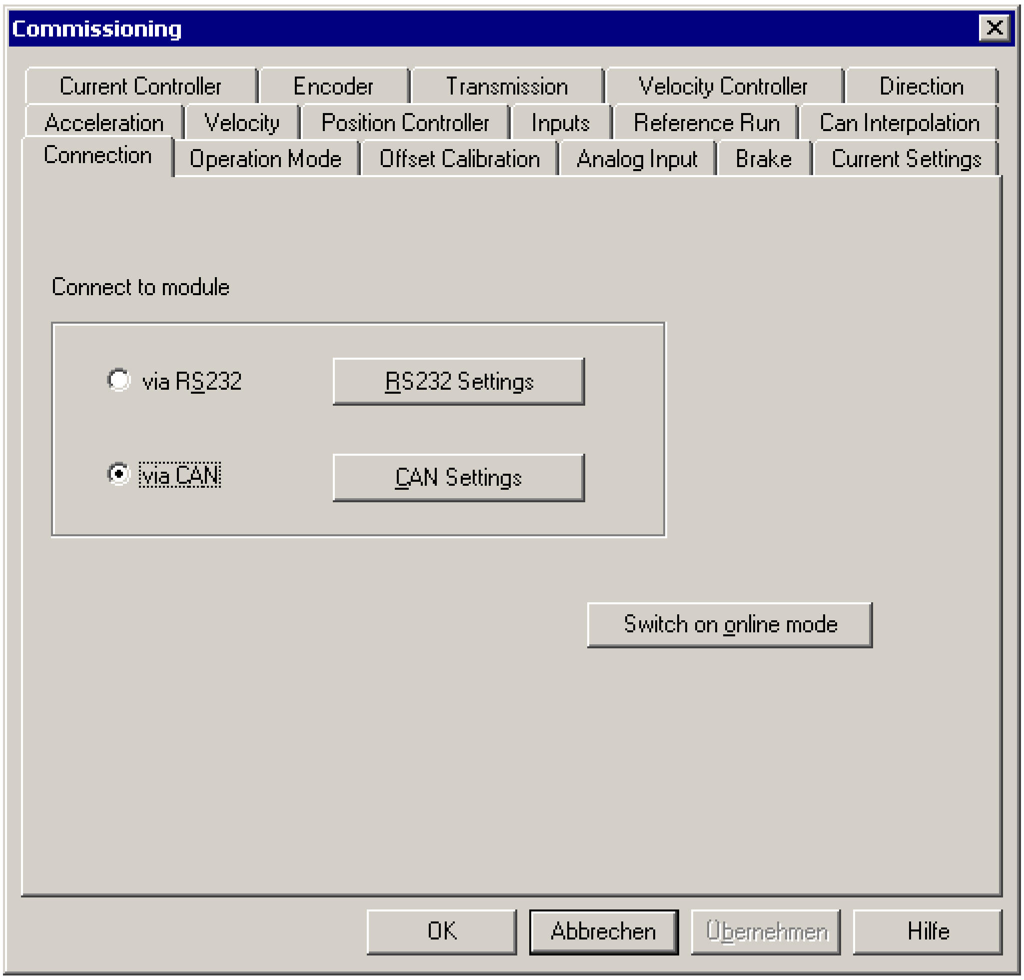
<OK> Changes are applied. The dialogue box is closed. A permanent storing does not take place. The permanent storing of data can be realised via the “Object Directory dialogue box” of the Settings\Object Directory main menu.

<CANCEL> Changes are not applied; the parameters are not saved.

<APPLY> The set of parameters is only transferred to the module. A permanent storing is only carried out automatically, if the default password “ISEL” is in use. Otherwise it is not carried out. The permanent storing of data can always be realised via the <OK> button or via the “Object Directory dialogue box” of the Settings\Object Directory main menu.

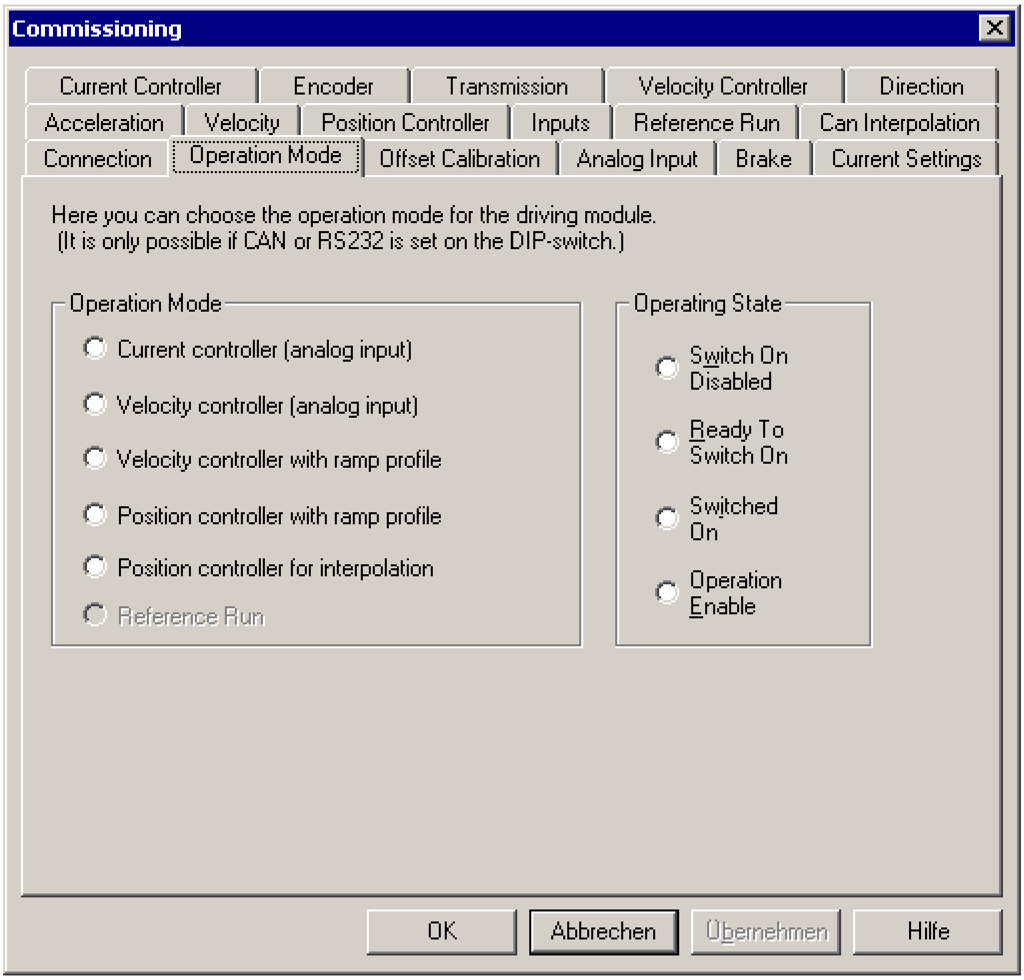
The commissioning via this option is reasonable, if one wants to make a quick change or detail improvements. In cases in which the drive module can move because of its own weight, even if the system is dead, a new commissioning shall also be carried out with the help of this option. For this, the parameters of the current, velocity or position controller have at first to be set to default values. The control is thus activated. On the basis of these default values, all other parameters can be determined. Here, in the beginning, it is hardly possible to carry out the commissioning via the “Step-by-step commissioning” option due to the inactive control.

Connection dialogue box



Here, you can select the active connection and adjust the corresponding settings. Afterwards, the online mode can be activated. During the step-by-step commissioning, the online mode is automatically launched with the <Next> button.

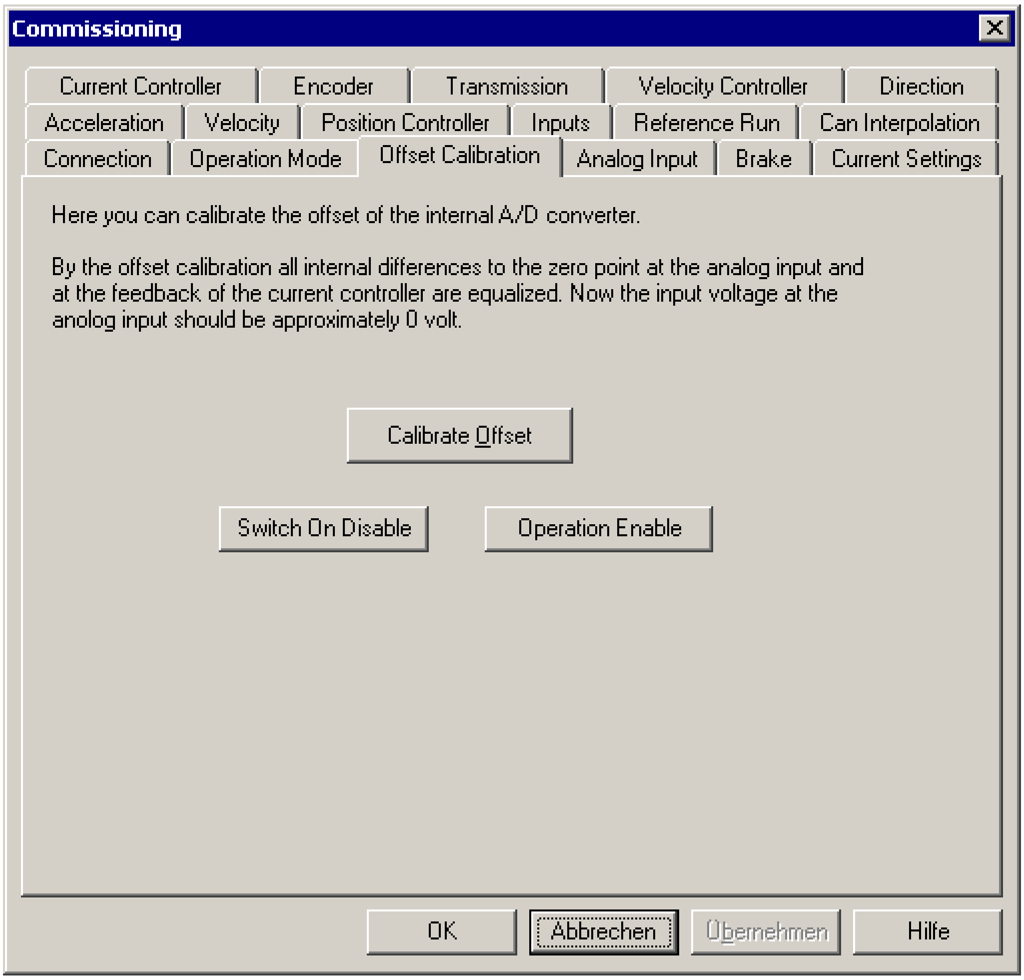
Operation Mode dialogue box



The current mode of operation is given in this dialogue box and can be changed, if the DIP switch was not used to set the analogue reference input (current or velocity control preset). In case of the step-by-step commissioning, the number of the following setting dialogue boxes depends on the mode of operation chosen here. The changing of the operational status is only possible in case of the Commissioning in a random order.

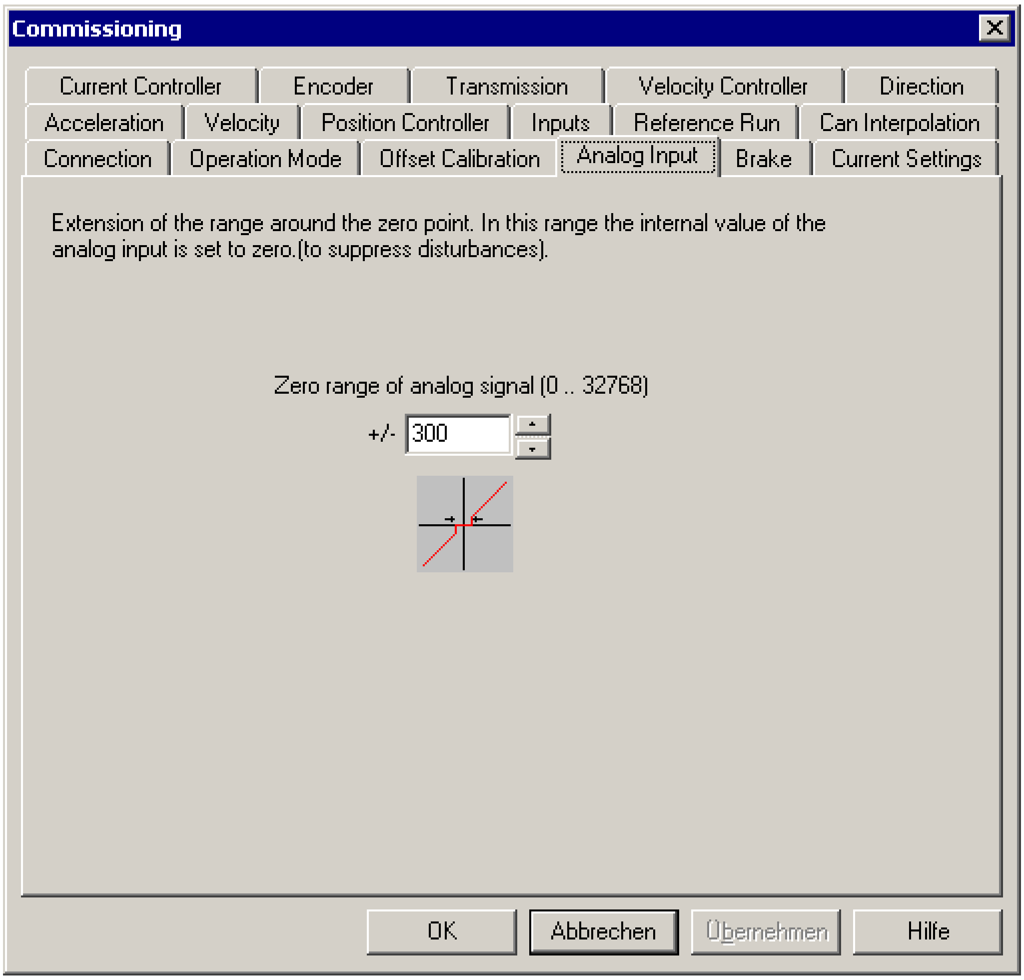
See "Modes of operation" on page 9.

Offset Calibration dialogue box



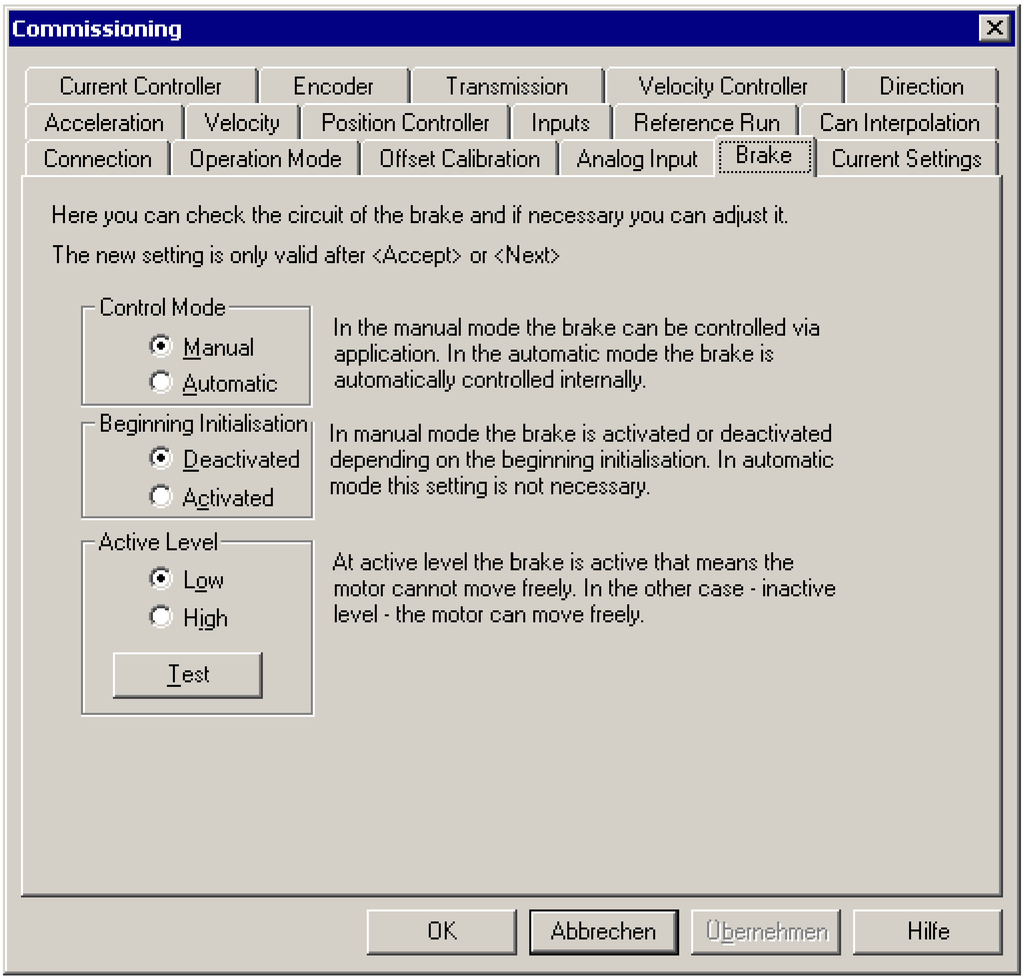
With the offset calibration of the analogue input and the internal current feedback, all static deviations from the zero point (internal or external) are adjusted. The offset calibration is realised automatically in the drive module and launched via the corresponding button. The voltage on the reference input shall correspond to the zero value and the motor must not rotate.

Analogue Input dialogue box

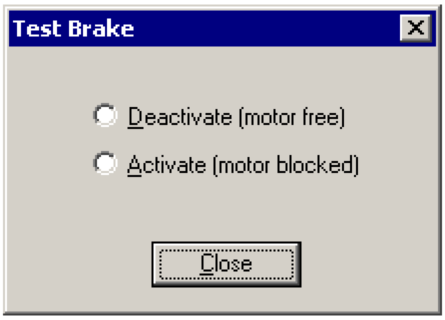


This setting possibility serves to eliminate small zero point deviations and voltage fluctuations on the analogue input. The digitalised voltage value is set to zero in the set area around the zero point. This parameter has only an effect on modes of operation with analogue input.

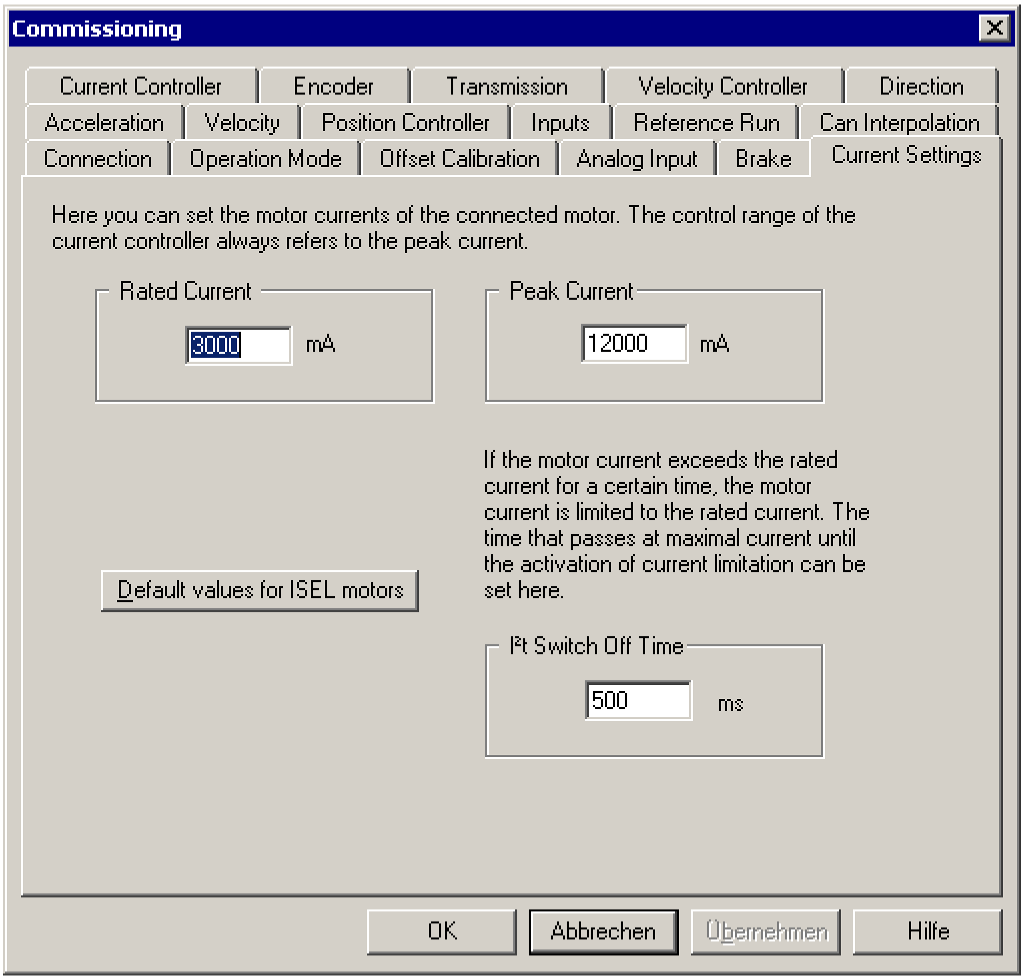
Brake dialogue box



Output 3 is specifically designed for the control of a connected brake. The use of the brake is optional. If a brake does not exist, the manual control mode has to be chosen. In this mode, the application software has to operate the output or brake itself. In the automatic mode, the brake is activated automatically, if the final stage is dead. The motor can no longer move freely. If the final stage is activated, the brake is switched off. Then, the motor can be moved. All settings must in any case be controlled with the help of the <Test> button.



Current Settings dialogue box

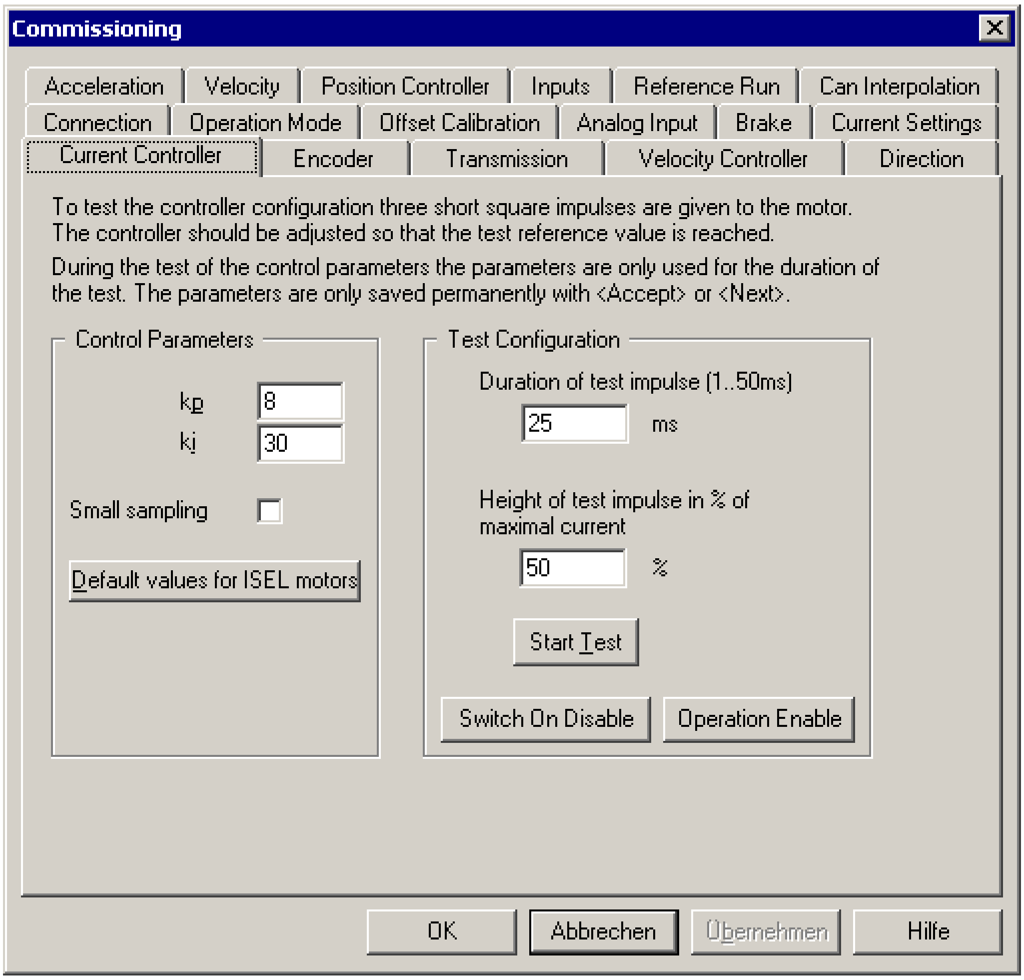


The rated current is the maximum admissible or desired continuous current in the motor. It may be briefly exceeded during normal operation. The peak current indicates a value for the maximum admissible current.

The I²t switch-off time indicates how long the peak current may flow without interruptions, until it is limited to the rated current. In case of a smaller exceeding of the rated current, the time up to the limitation increases quadratically.

For the current controller, only the peak current is of importance. I.e. 100 % target current value = peak current. If the peak current is changed, usually, the control parameters have to be adjusted, too.

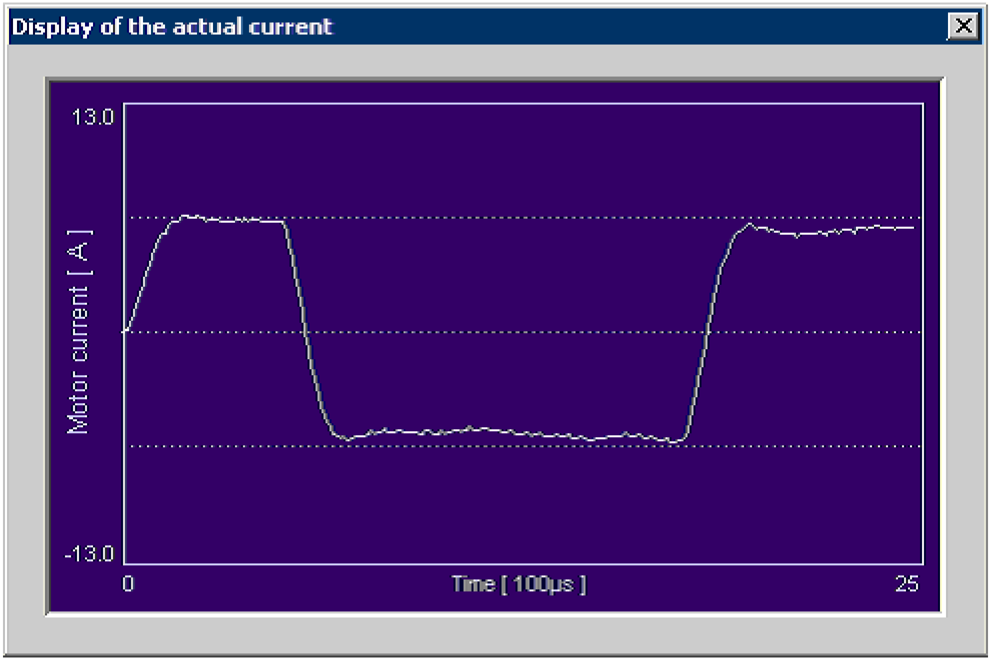
Current Controller dialogue box



The setting of the PI current controller is realised via the parameters **kp** (proportional amplification) and **ki** (integral amplification). Based upon the default values, the controller settings can be optimised. In order to facilitate the setting process, an internally generated rectangular signal (three impulses) is applied on the controller's input for the test. Here, the overall duration and the amplitude of the test signal can be set. The path of the actual current is recorded internally and then queried and displayed by the DcSetup programme (see Figure). For the setting of the controller, it has to be made sure that the target value (dotted line) is achieved and maintained.

The selection of the "Short scan time" option causes a quicker clocking of the current controller. A faster current control is very useful for the correction of errors such as e.g. in case of a toothed belt forward feed.

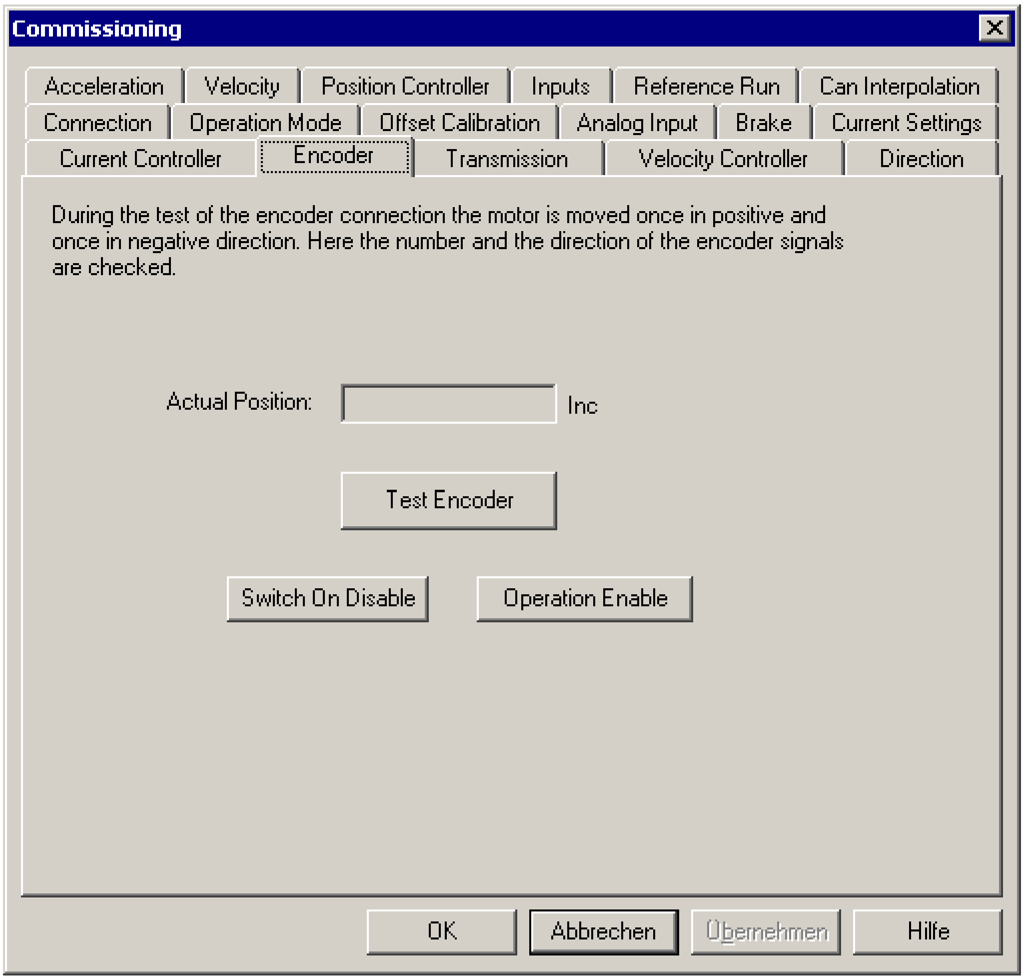
Note: If the motor does not have to drive a large mechanical load and if the duration of the test impulse is too long, it may be that the motor goes into saturation and that the current decreases.



On the basis of the actual current display, the jump response of the controller can be assessed.

Only if the target value is complied with, the set current values for the rated and peak current correspond to the actual currents.

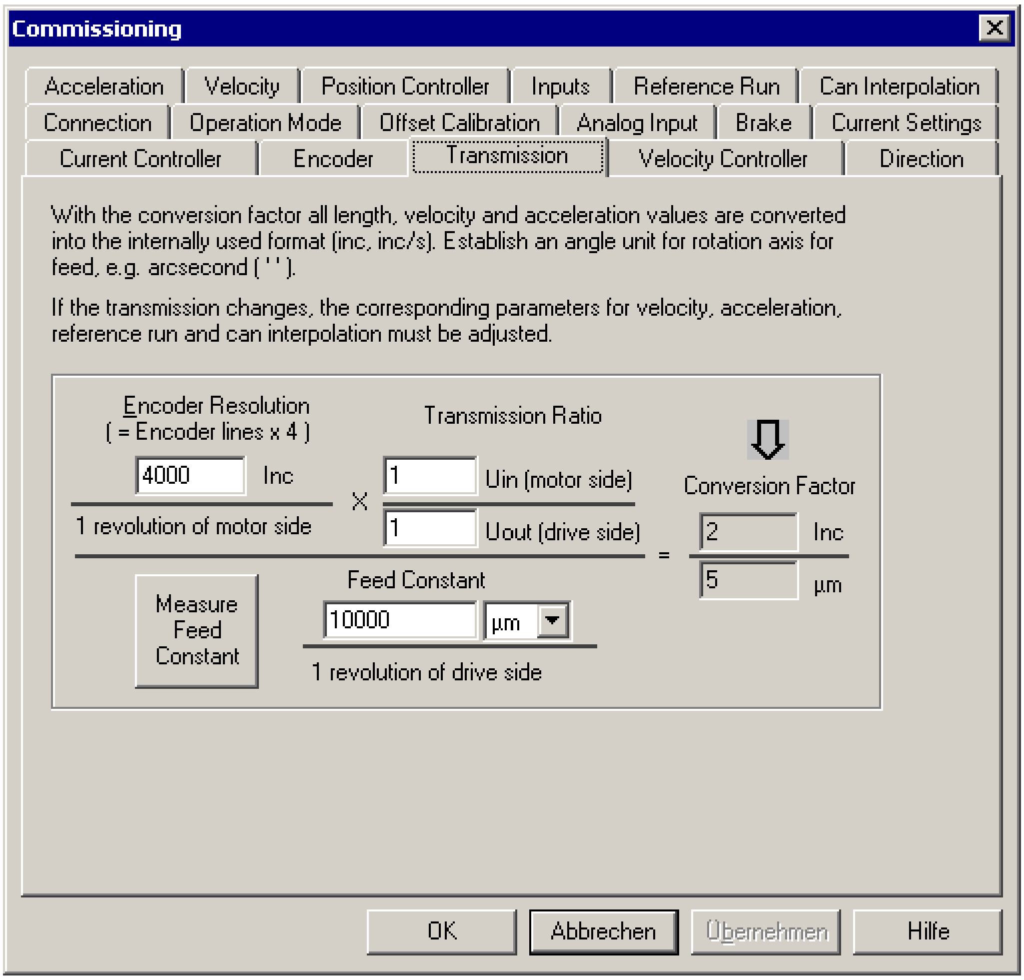
Encoder dialogue box



An operational encoder connection is an essential condition of the commissioning of the velocity and position controller. If the encoder is not connected or not connected correctly, the motor would make uncontrolled movements.

The test of the encoder connection is carried out internally in the drive module. At first, it is tested, if a certain number of impulses (e.g. 100) is exceeded at standstill (about 1 sec.). This could be caused by an open connection or faults. Afterwards, the motor is moved in one direction, until a preset number of impulses (1,000 Inc.) is achieved or until a preset time (100 ms) has been exceeded. A pause of about 2 seconds is to make sure that the motor is once again brought to a standstill. Afterwards, the same procedure is made for the other direction.

Transmission dialogue box



The conversion of units of length or angular measurement into the internally used Increments format is made in the drive module. For this, the objects of the Factor Group defined in the CANOpen specification DS402 are used.

The conversion factor (Position factor 6093_h) is the value which is used for all conversions. It is derived from the formula given, where the encoder resolution, the transmission ratio and the forward feed constant can be set. However, it is also possible to manually enter any value as conversion factor, if this is necessary. The units for the speed and acceleration are always derived from the position unit as Unit/Sec. or Unit/Sec.².

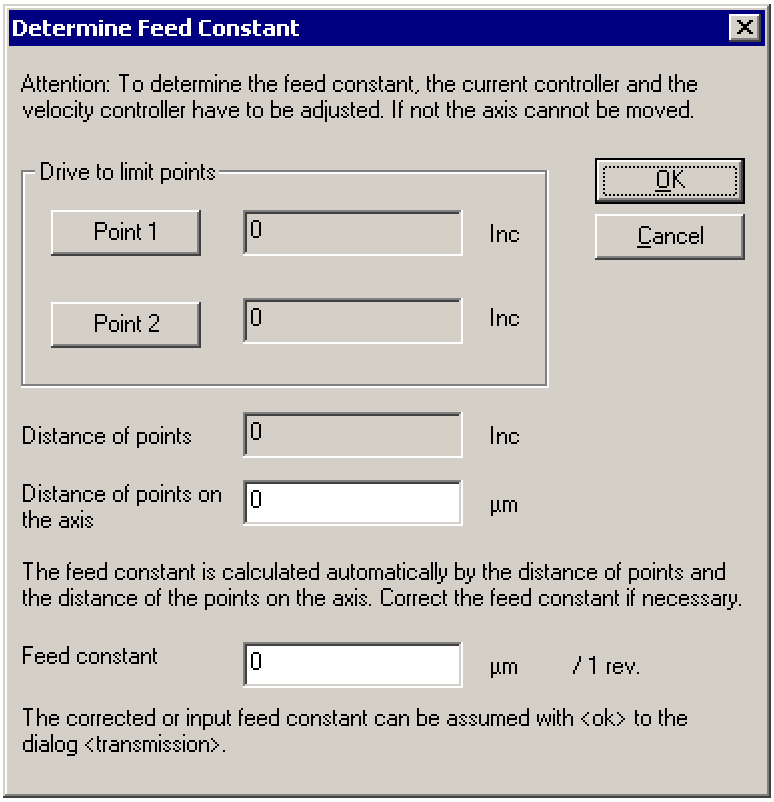
In the example above, a linear axis with a spindle pitch of 10 mm/rev without transmission is required. The position is to be given in the [µm] unit. Thus, the units for the speed and acceleration would be µm/sec. or µm/sec.², respectively.

If a rotary axis is used, it is reasonable to use a unit of angular measurement. Here, one can choose between arc second [“], arc minute [‘] and degree [°].

If a conversion is not wanted, the same value can be given for both the forward feed constant and the encoder resolution; the conversion factor would be 1.

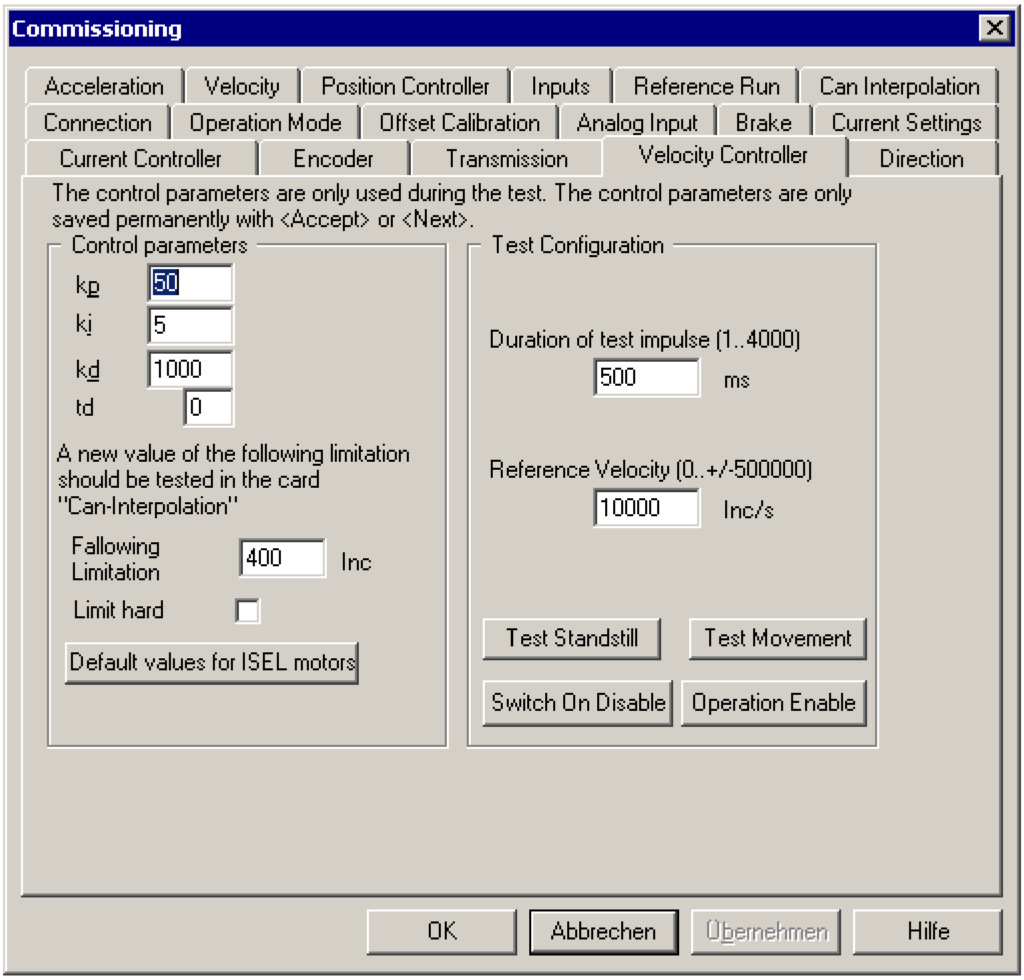
The encoder resolution must be indicated as the quadruple of the number of lines on the encoder (4-edge evaluation).

With the button „Measure Feed constant“ you can determine the feed constant out of the distance of two points on the axis. The following dialogue appears:



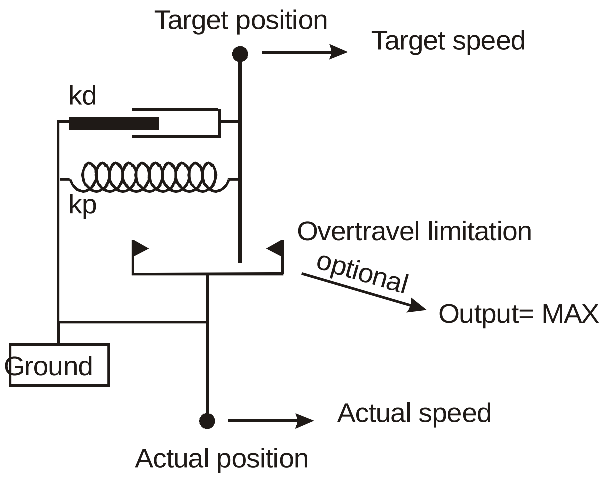
With the buttons „Point 1" and "Point 2" you can drive to two different points on the axis. Measure the distance of the two points and input the distance in µm in the edit field "Distance of points on the axis". The feed constant is calculated automatically while putting the measured distance into the edit field. Correct the calculated feed constant if necessary. With "OK" you can assume the feed constant into the dialogue box transmission.

Velocity Controller dialogue box



The setting of the PID velocity controller is carried out by analogy with the setting of the current controller (see Current Controller dialogue box). Furthermore, there are the parameters **kd** (differential amplification) and the related scan time **td**. The td factor affects the scan time of the differential portion according to the function: scan time of the D proportion = scan time of the controller x (1 + td). This value should only be set greater than 0 in case of heavy axes. An increase of the td value by one corresponds to a duplication of the kd value.

Actually, the velocity controller is a position controller with the function to count the target position with a constant rate. This rate corresponds to the actual target velocity. Due to this principle, the **Following limitation** is a further parameter to take into account. Since, if the motor cannot follow its target position, a very great deviation would be the result, which would have to be caught up with a higher speed.



The deviation between the target and the actual position cannot be greater than the following limitation, since the target position follows the actual position, if this distance is exceeded. Too great a value of an following limitation might lead to a permanent vibration of the axes. For the optimisation of the following limitation, see Can Interpolation dialogue box on page 62.

If the **Hard limit** option is enabled, the controller output is set to its maximum values, when the following limit is achieved. Otherwise, it has a value corresponding to the deviation.

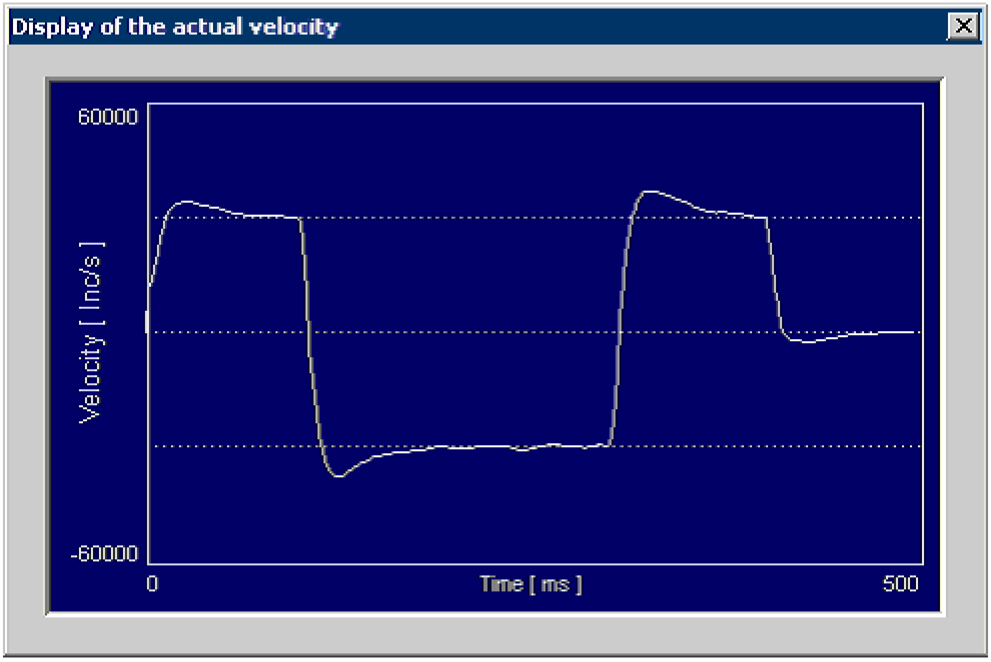
For the testing of the controller setting, a test signal can be used, too. For this, three rectangular impulses are defined as target speed (the overall duration and amplitude can be set). The actual speed can be used to verify the controller setting. New control parameters are transferred only briefly to the drive module during the test and are then replaced again by the previous parameters. The parameters can be tested both at standstill as well as in motion.

In the beginning, it is best to use the default values and to only then try different settings.

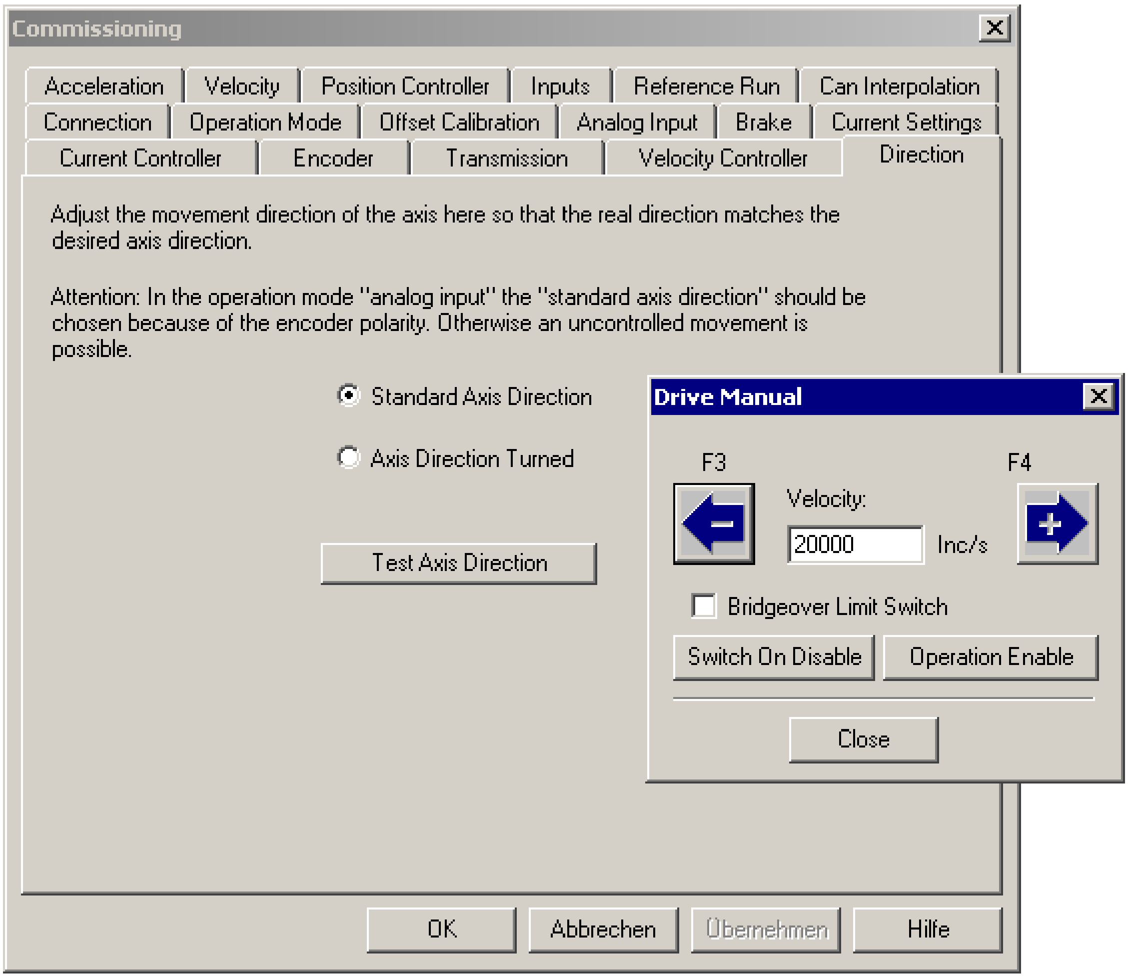
Some basic setting rules:

- In case of a strong following, reduce kp and / or increase kd.
- Too high a D proportion leads to a rough motor operation and to loud noises.
- An increase of td corresponds to a duplication of the D proportion (half kd).
- ki should not be greater than kp. Often, very small values suffice.
- For the setting of the following limitation, a higher target speed should be chosen. The reduction of the following limitation leads to a reduction of the following and a flattening of the ramps. Reduce the following limitation up to the point, at which the ramp slope is still as high as in case of a high following limitation. It might be possible to increase kp further. Then, the “Hard limit” option can be activated in order to achieve a higher acceleration.

The requirements for the controller can differ considerably from application to application. A smooth course of the actual speed with a small following (see Figure) e.g. is a good compromise for many application purposes.



Direction dialogue box



Every axis (linear axis, rotary axis) driven with the positioning module should have a defined axis direction.

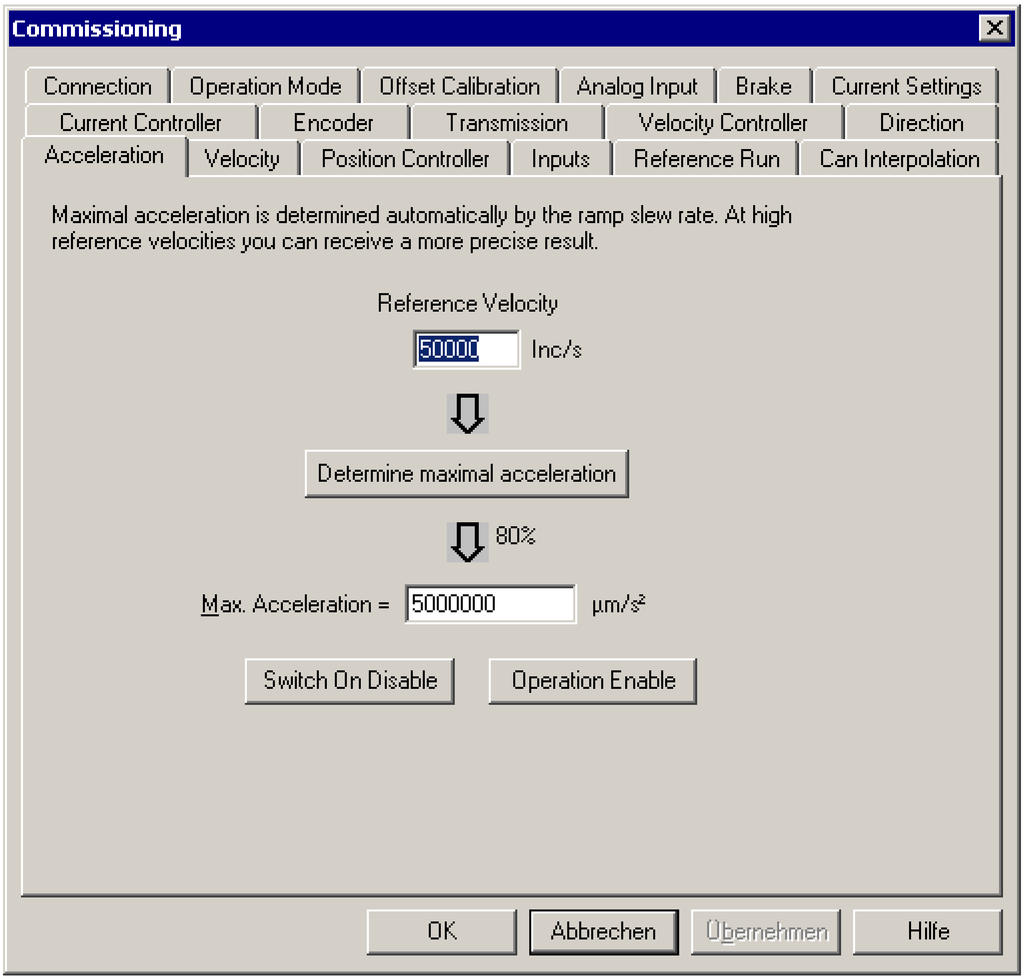
If the motor is connected correctly, it should rotate to the right in case of a positive direction and to the left in case of a negative direction (viewed from the front).

Here, the rotation direction of the motor can be adapted so that it corresponds to the defined axis direction of the connected mechanical facilities. For this purpose, a dialogue box can be displayed by means of which the axis is moved into the positive or negative direction.

If the analogue mode of operation (current or velocity controller) is activated, the default axis direction has to be chosen. Otherwise, the movement of the motor during the switch-on is uncontrolled.

Note: If the axis direction is changed, the limit switch inputs of the positive and negative limit switch are also inverted.

Acceleration dialogue box

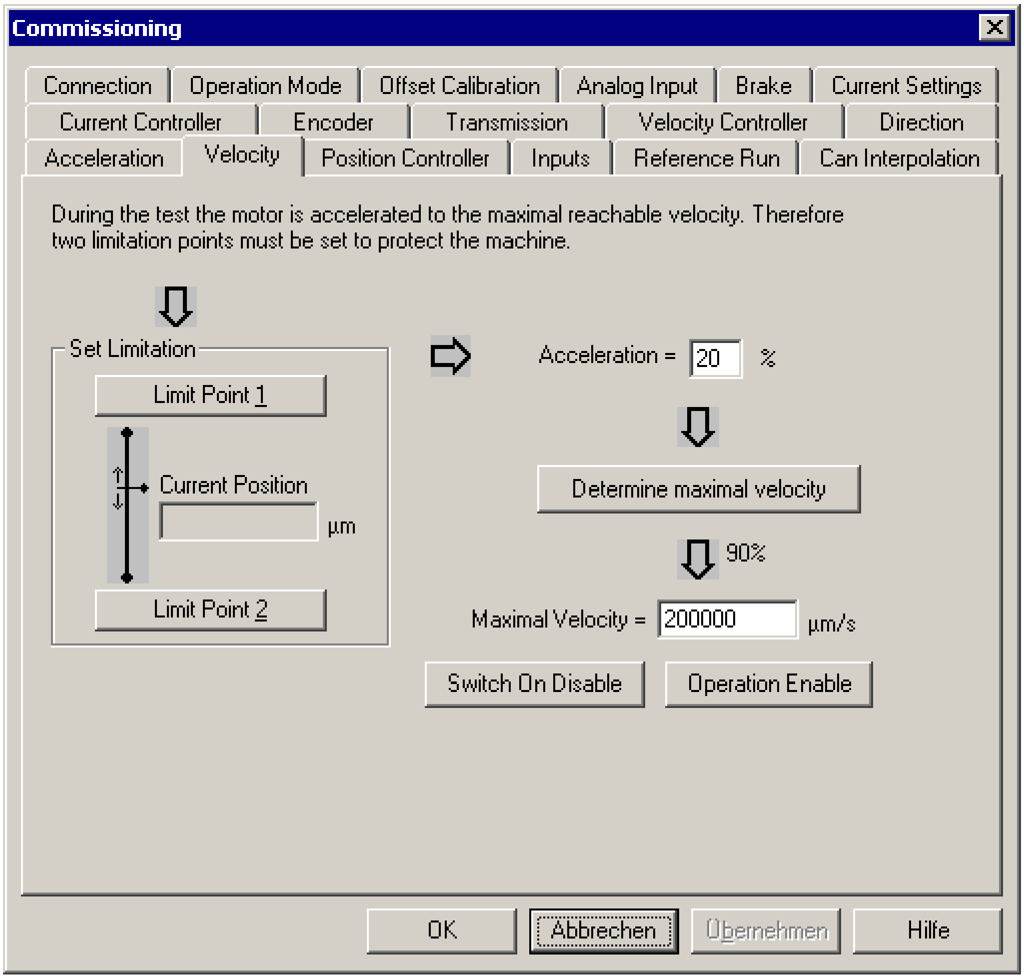


The maximum acceleration is the limitation for the acceleration of all motion functions with a ramp (profile velocity mode, profile position mode). I.e. the acceleration cannot become greater than this value, even if a different parameter (e.g. profile acceleration) contains a higher value.

The determination of the maximum acceleration can be carried out automatically. For this, a rectangular signal is used in order move the motor back and forth. At the same time, the build-up times of the ramps are measured and averaged. This value is limited to 80 % in order to have a certain reserve for the control.

It has to be taken into account that the calculation becomes very imprecise in case of relatively small speeds. The target speed should come to about ¼ of the maximum speed or more.

Velocity dialogue box



The maximum speed is the limitation for the speed of all motion functions with a ramp (profile velocity mode, profile position mode). I.e. the speed cannot become greater than this value, even if a different parameter (e.g. Target velocity 60FFh) contains a higher value.

In the mode of operation “Velocity controller with analogue input”, this value corresponds to the maximum achievable speed (-10 V …-+10 V –max. velocity ... +maximum speed). See “Operation Mode ”on page 44.

The determination of this parameter can be carried out automatically. Here, it is, however, necessary to define a motion sector, since the axis might cover a long way during the acceleration to the maximum speed and the subsequent braking and may not drive into the mechanical stop.

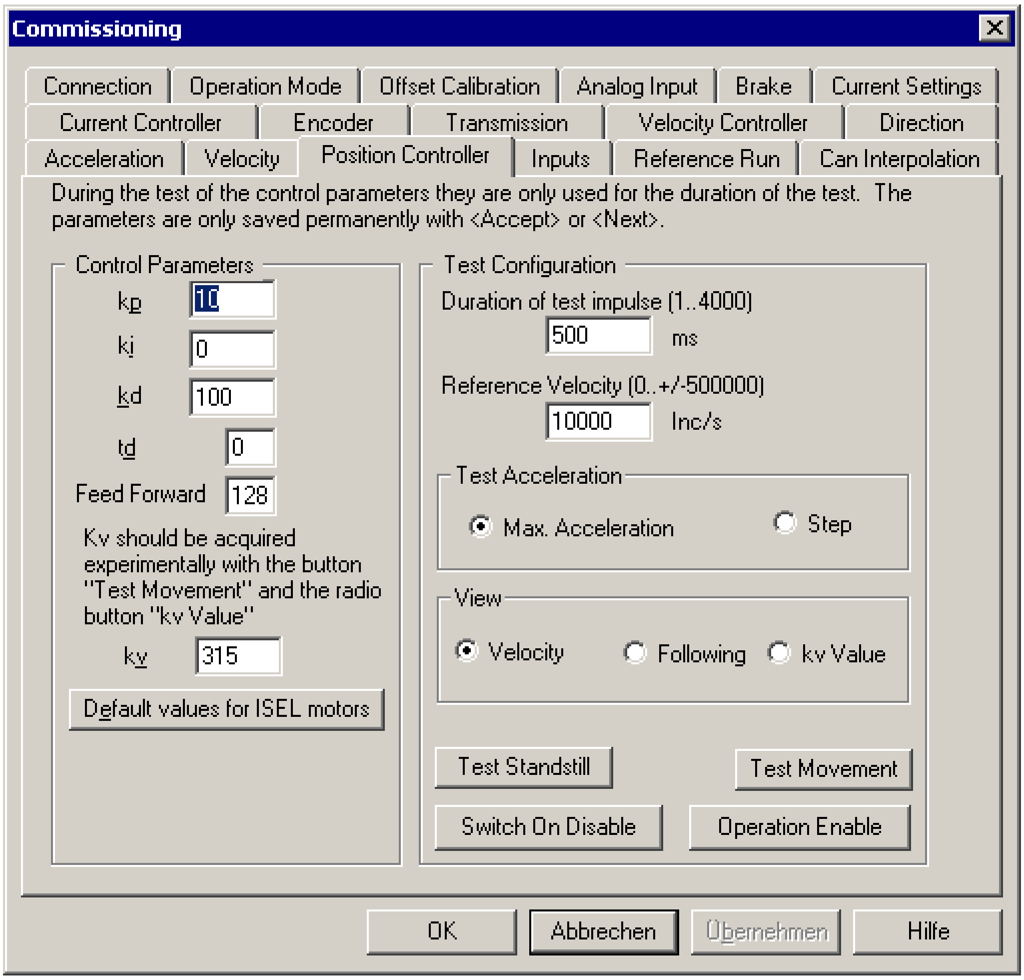
See Acceleration dialogue box on page 57.

In order to determine this motion area, at least one limit point has to be set in which the axis is driven to the corresponding spot via a Teach dialogue. In the beginning, both limit points contain the actual position.

The acceleration is by default set to 20 % of the maximum acceleration in order to go easy on the mechanical parts, if e.g. the motion sector becomes too small.

The maximum speed is determined internally and then limited to 95 %, in order to dispose of a good reserve for the regulation.

Position Controller dialogue box



The position controller is a PID controller with velocity feed forward. This is superimposed on the velocity controller, i.e. the velocity controller has to be set first. By default, the I proportion is set to zero in order to avoid an following of the control. A feed forward always leads to better dynamics, but also to an following, as does an I proportion unequal to zero. Normally, the default values for the I proportion and the feed forward should be applied.

For the test of the settings, a test signal is again available consisting of three impulses with different signs. The ramp slope can be chosen between the (previously determined) maximum acceleration and a maximum ramp (jump).

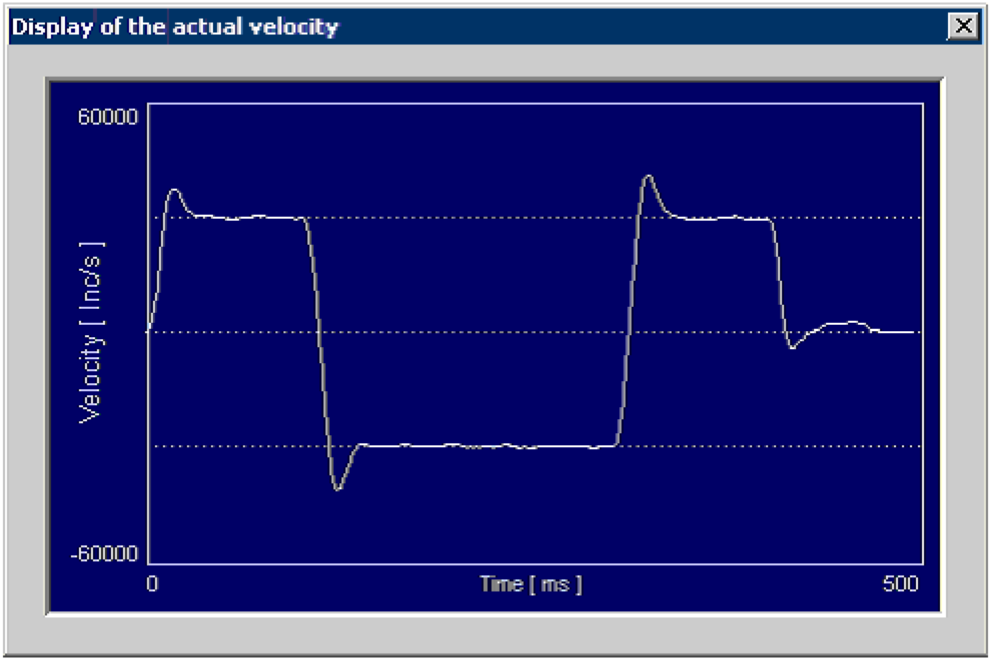
In order to be able to assess the settings, the display options speed and following error (control deviation) are available.

The setting of the regulation parameters can be realised as follows:

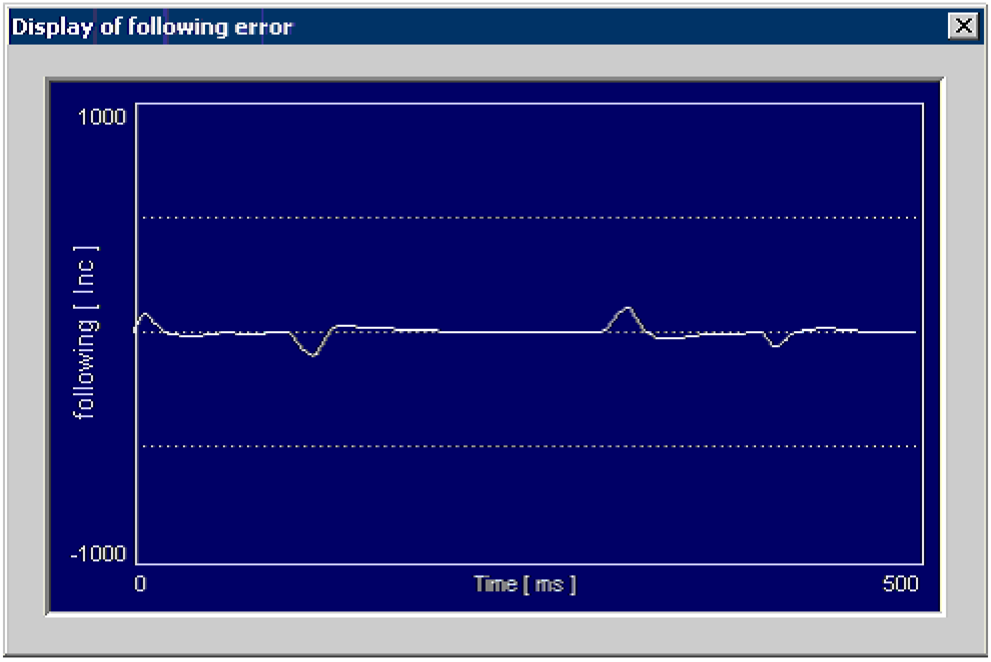
- Load default values.
- Set test acceleration to maximum acceleration and control the course of speed.
- Increase kp until a considerable following can be noticed.
- Increase (and / or td) and adapt kp, if necessary, in order to smoothen the course of speed. Test different target speeds. Normally, td should be set equal to 0. In case of severe faults such as a toothed belt feed forward, it is sometimes reasonable to increase td instead of kd in order to achieve a high stability at a low noise level. It has to be taken into account that an increase of td by 1 leads to a duplication of kd.
- Set the display to following and change the forward feed factor (and kp), until the following error becomes minimal. For this, increase the target speed.
- Set the test acceleration to jump and control the transient behaviour of the control cycle even at higher target speeds (speed display). If the axis switches to a permanent vibration, limit the following limitation of the velocity controller.

Set the test acceleration to jump and control the transient behaviour of the control cycle even at higher target speeds (speed display). If the axis switches to a permanent vibration, limit the following limitation of the velocity controller.

It can be useful for the optimisation to once again vary the velocity controller in order to achieve a better course of speed and a smoother motor operation. During the commissioning in a random order, however, one has to make sure that the parameters of the velocity controller have to be stored with <Apply> before every switch-over to the position controller.



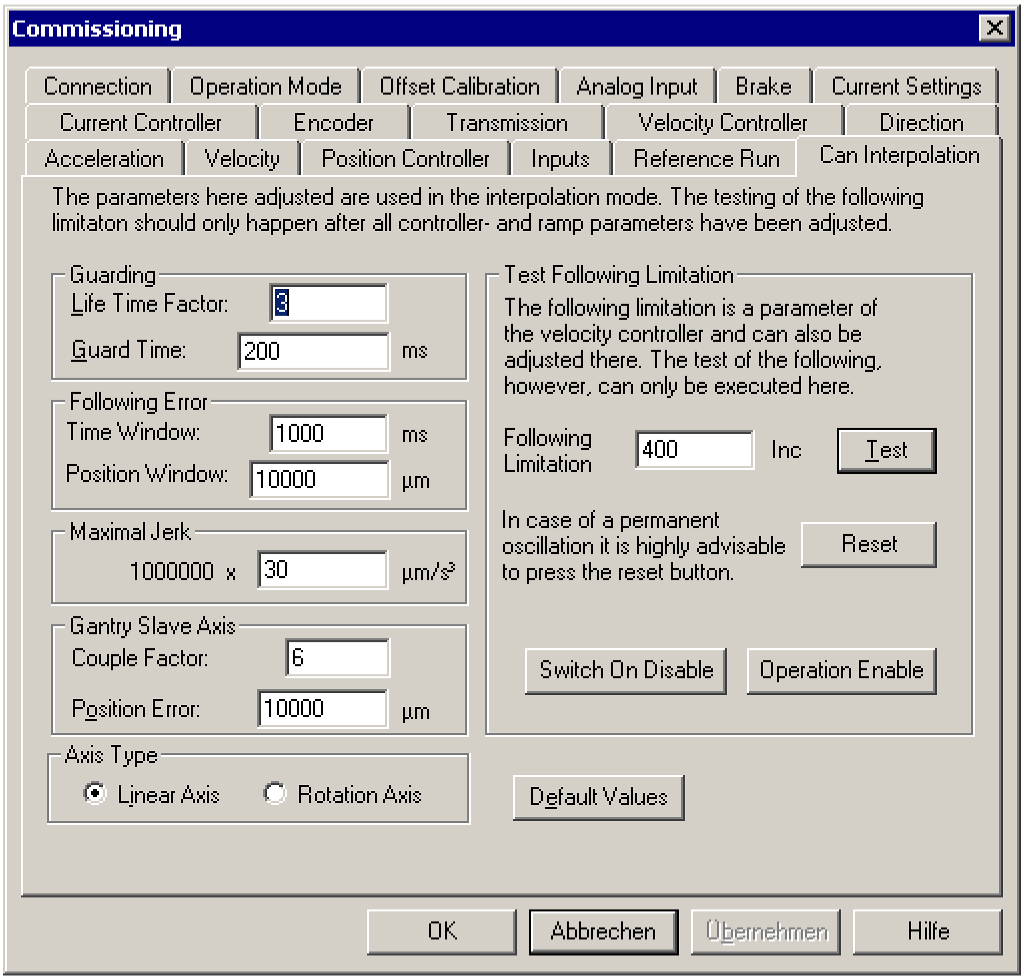
Course of the actual velocity (speed)



Course of the control deviation

The speed amplification is a characteristic parameter for the dynamism of an axis. This parameter is not used within the positioning module. A superordinate CNC control such as the ISEL control can read this parameter and use it for the optimisation of the sequence of motions. The kv factor is to be determined experimentally by selecting the “kv_value” radio button.

Can Interpolation dialogue box



The parameters set here were designed for the modes of operation “Position controller with ramp profile” and “Position controller for interpolation” (see “Operation Mode dialogue box” on page 44).

The two parameters “Life Time Factor“ and “Guard Time“ can be used to activate the watchdog function of the positioning module (see Monitoring – Guarding on page 76). If the value of one the two parameters equals 0, the watchdog function is switched off.

The monitoring of the following error is carried out by means of the two parameters “Time window” and “Position window”. If the actual position for the “Time window” time deviates from the target value by the “Position window” amount, an internal flag is set. A superordinate control can query this flag and react accordingly.

The axis’ vibration behaviour can be influenced by the “Maximum jerk” parameter. The smaller the jerk value, the less the vibration of the axis during the accelerating and braking processes. In return, the acceleration and braking process takes longer accordingly. The jerk value can be read in and used by the CNC control. Within the final power stage, this parameter is not used. The CNC control of ISEL uses this parameter.

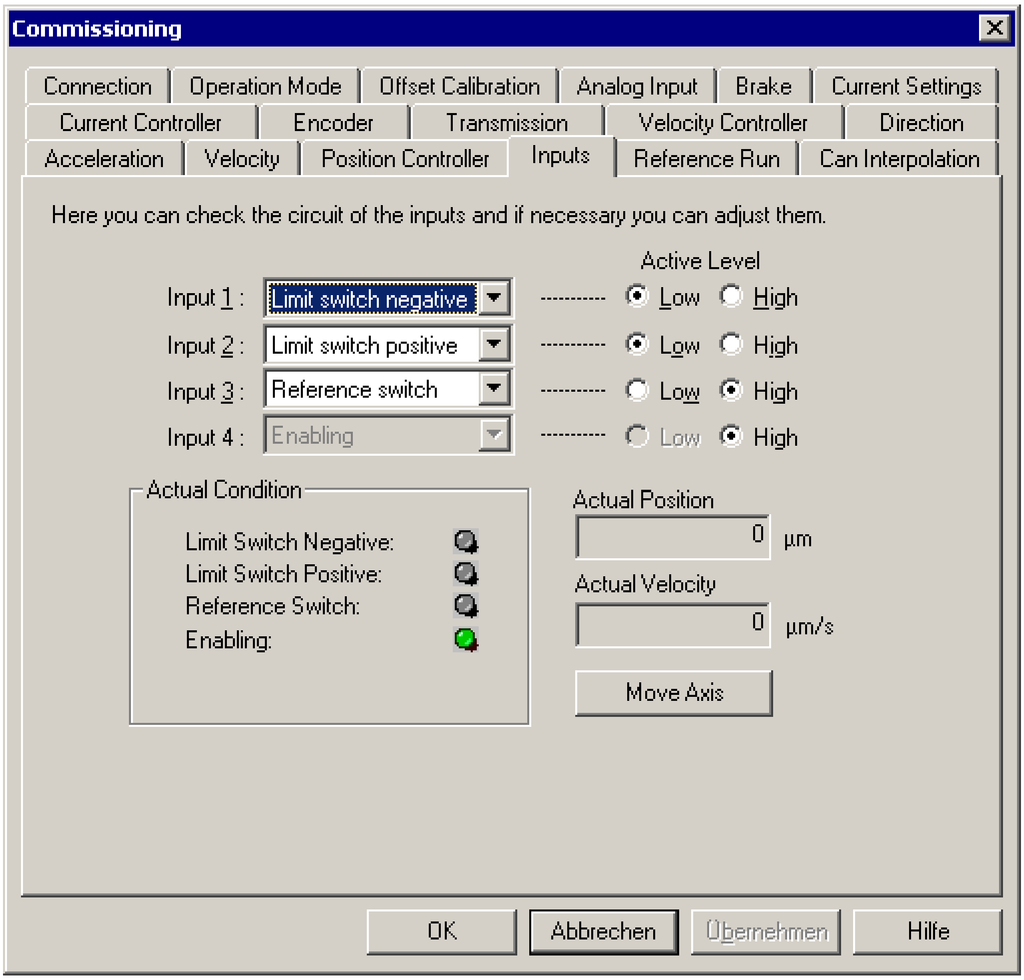
Concerning the axis type, the axis can either be set as a linear or as rotary axis. The CNC control of ISEL uses this parameter for the display. Otherwise, the axis type is not used.

In case of a gantry axis, where two axes are to be moved synchronously, the maximum position error can be freely defined. This limit value is monitored only by the slave axis. The master axis does not take into account this limit value. If the deviation of position between the master axis and the slave axis exceeds this value, the slave axis is immediately stopped with the error code 22 (see section on fault states). The related master axis is informed of this error via the CAN bus and stops immediately with the error code 28.

In case of a gantry axis’ slave axis, the user also has the possibility to define the coupling factor. The greater the factor, the “closer” the slave axis is connected with the master axis. Thus, the deviation of position between the two axes is also smaller during the gantry operation. This means: the greater the coupling factor, the better the gantry operation. But the regulation on the slave axis is unfortunately “rougher” The user should always apply the default value as initial value and change it by steps of 1 upwards or downwards. In the declaration mode, the user can observe the actual and maximum deviation of position between the master and the slave axis by means of the “Settings\Object Directory” menu.

Too large an following limitation value can lead to a permanent vibration during interpolation. Too small a value restricts the motion speed of the axis. It is recommendable to test the set value. After the “Test” button has been activated, the test runs automatically. A potential permanent vibration of the axis can be interrupted by the “Reset” button. An optimum value is the maximum possible value at which the permanent vibration was not yet noticed. The value of the following limitation can also be changed via the “Following limitation” data entry field (see also “Velocity Controller dialogue box” on page 54).

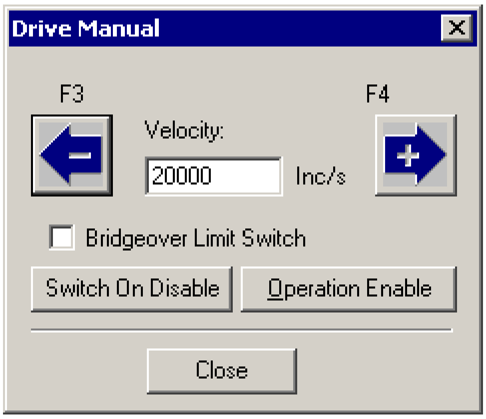
Inputs dialogue box



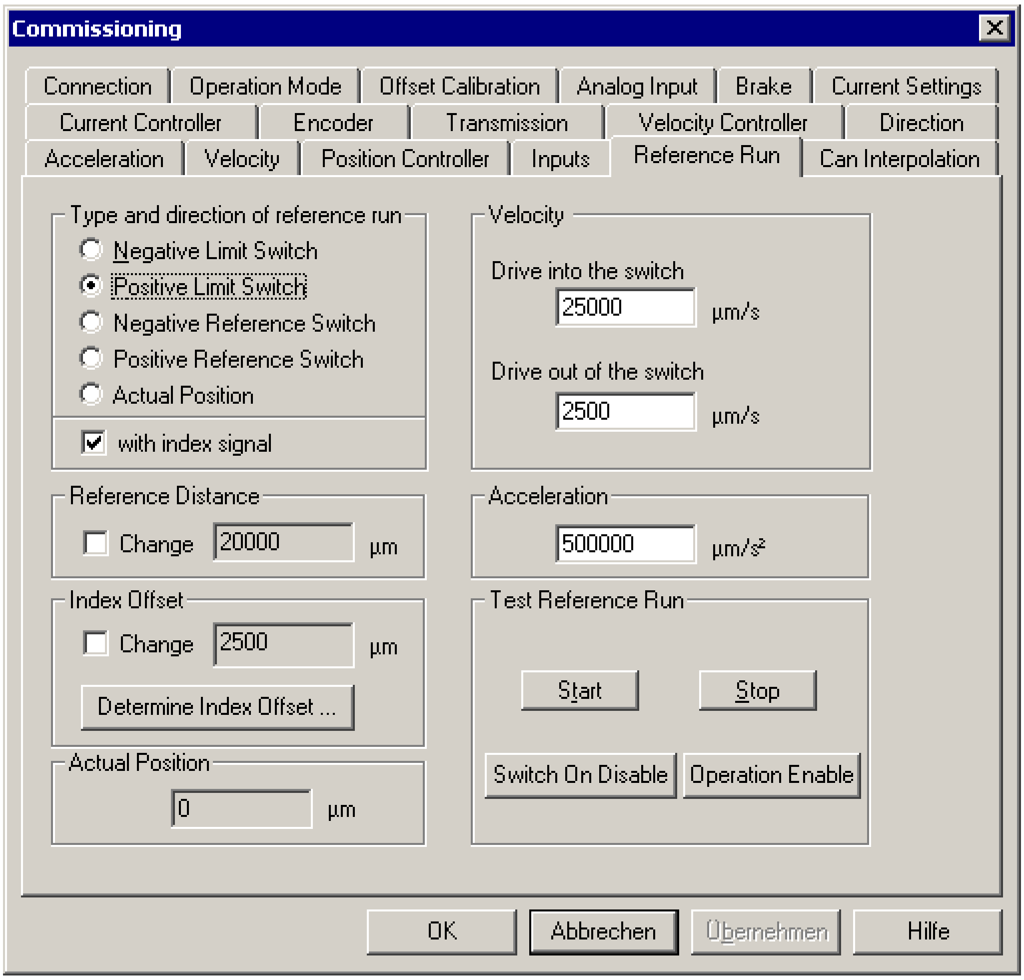
The positioning module disposes of four digital inputs for two limit switches, a reference switch and an enabling signal (see “Digital inputs 1 to 4” on page 14). The allocation of the signals to the inputs can be made at random, apart from the enabling signal which is always to be led to input 4 (hardware connection of input 4 with final stage release).

The actual status of the inputs can be read via the LEDs. This display corresponds to the state as it is evaluated by the control (light green = input is active, grey = input is not active).

In order to test the limit switch on one axis, said axis can be displaced to the limit switches by means of a Teach dialogue (move axis). If the activation of the limit switches via the emergency stop chain of the plant would lead to the switch-off of the final stages, the option <Shunt limit switch> can be used to set an output of the final stage that should be used to separate the limit switch of this axis from the emergency stop chain.



Reference Run dialogue box



The reference run serves to define the zero point of an axis. For this, the axis moves with the “Drive into the switch” speed in one direction, until the switch indicated becomes active. Afterwards, the axis (slowly) moves with the “Drive out of the switch” speed in the other direction, until the switch switches off again. Finally, the axis is brought to a defined distance (reference distance) from the switching point. This is then the zero point of the axis. The reference acceleration should amount to about 1/10 of the axis acceleration (see “Acceleration dialogue box” on page 57) in order to go easy on the mechanical axis during the reference run. In case of a reference run with index signal first the axis moves after switch off to the index offset without considering the index signal. After this the axis moves as long as it recognizes the index signal. After recognition of the index signal the axis moves to the reference distance and then sets the zero point of the axis. To determine the index offset first you have to drive a reference run without index signal and then a reference run with index signal (Dialog "Determine Index Offset" over button "Determine Index Offset ..."). After completion of the both reference runs the index offset is determined by the positions the reference runs have reached. For this the difference position of the two reference runs is calculated. If the difference position is between 1/4 of the encoder resolution and 3/4 of the encoder resolution the index offset is set to zero. If the difference position is out of this range the index offset is set to 1/2 encoder resolution. With "Assume Index Offset" you can assume the set respectivly changed index offset into the dialog "Reference Run".

The dialogue box comprises all parameters influencing the execution of the reference run. By means of <Type and direction of the reference run>, you define e.g. which switch is used and in which direction the axis is to go. In order to achieve a high precision of the machine's zero point, a combination with the index signal is possible, provided the index signal is connected to the encoder. Likewise, the speeds and the reference distance can be set here. Furthermore, there is the possibility to test the reference run.

Internally, the module is switched to the Homing Mode – Reference run CanOpen mode of operation for the execution of the reference run; then the reference run is started and, after its termination, the previous mode of operation is switched back to.

In case of a reference run on one of the two limit switches, note that, during the reference run, the output for the shunting of the limit switch (safety circuit) is set in order to prevent the final stages from switching off, when the limit switch is activated.

In case of a gantry axis, the master and slave axis may have different motion parameters and reference methods. Only the directions of the reference run have to be identical for both axes. On the basis of the values indicated, the control calculates the optimum motion behaviour for both axes. The reference distances can be used in order to adjust the deviation of position of both axes in the zero point.

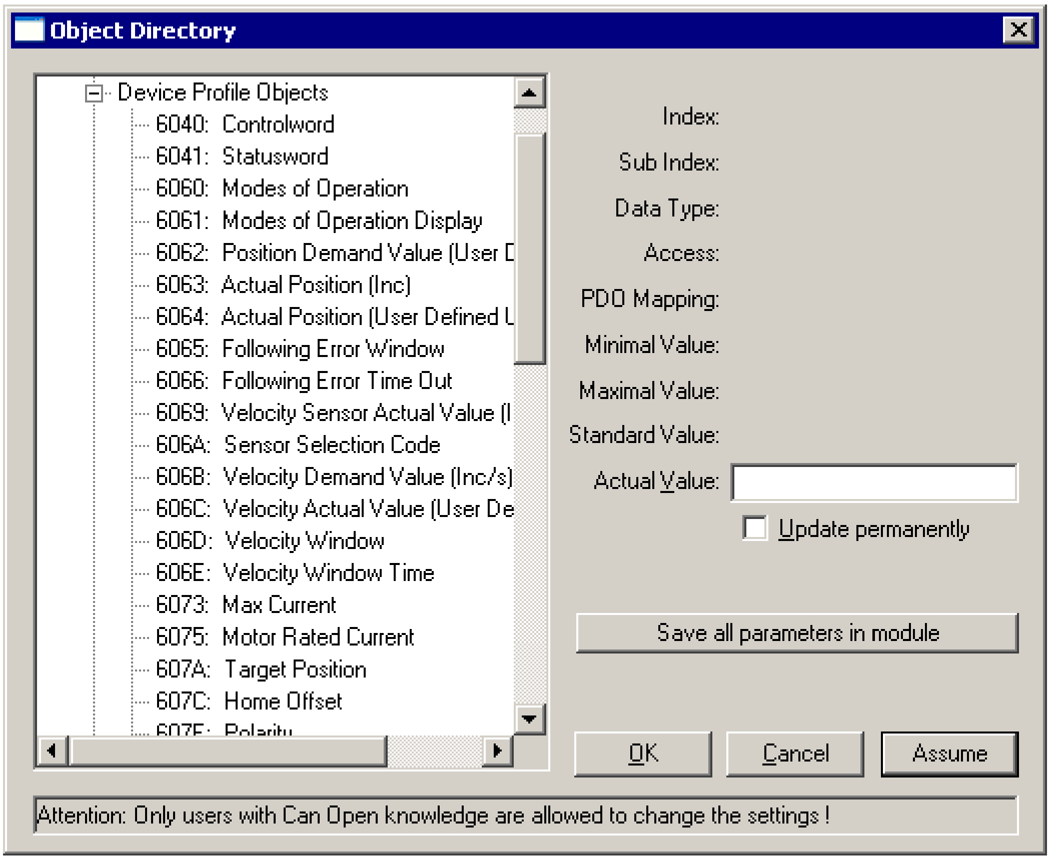
Object Directory dialogue box

The Object Directory dialogue box shows a tree view of all parameters (CanOpen objects) of the drive module. Via this structure, you can directly access all objects and read and / or write the parameter according to the respective characteristics. A description of all objects can be found in the "Object Directory" chapter on page 78. Only users experienced in CanOpen standards may use this dialogue box in order to set parameters. All other users shall realise the setting of parameters via the "Commissioning" dialogue box (see Commands of the Commissioning menu on page 33).

The tree view on the left side is subdivided into Communication (DS301), Device Profile (DS402) and Manufacturer-Specific Objects. On the right side, the characteristics and the content of the currently selected object are displayed. By means of the <Assume> button, you can realise changes in the current parameters. <OK> also confirms the current change and closes the dialogue box.

The "**Always update**" option has the effect that the value of the currently selected parameters is read about 5 times per second from the drive module (in case of an enabled online mode).

Via the "**Save current values in module**" button, you can permanently store the currently set values (set of parameters) in the drive module (the storing of individual parameters is not possible).


Object Directory
Device Profile Objects
6040: Controlword
6041: Statusword
6060: Modes of Operation
6061: Modes of Operation Display
6062: Position Demand Value (User D
6063: Actual Position (Inc)
6064: Actual Position (User Defined U
6065: Following Error Window
6066: Following Error Time Out
6069: Velocity Sensor Actual Value (I
606A: Sensor Selection Code
606B: Velocity Demand Value (Inc/s)
606C: Velocity Actual Value (User De
606D: Velocity Window
606E: Velocity Window Time
6073: Max Current
6075: Motor Rated Current
607A: Target Position
607C: Home Offset
Index:
Sub Index:
Data Type:
Access:
PDO Mapping:
Minimal Value:
Maximal Value:
Standard Value:
Actual Value:
Update permanently
Save all parameters in module
OK
Cancel
Assume
Attention: Only users with Can Open knowledge are allowed to change the settings !

Firmware update

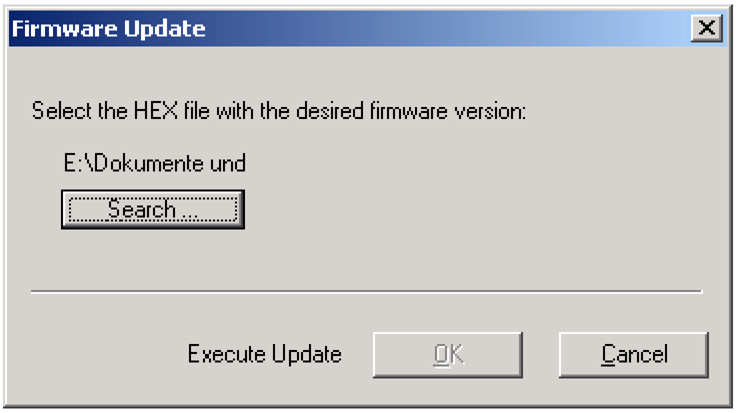
The software of the drive module is stored in a rewritable Flash memory. Thus, the programme can be loaded easily from the outside into the module, without memory modules having to be replaced. The loading of a new software version can only be realised via the serial interface.

Apart from the programme, the configurable set of parameters is also in the Flash memory. The memory sector is composed of four segments, the fourth segment being used to store the current parameters. A segment-wise deletion leads to the fact that only the first three memory banks are newly written during the normal update, i.e. the set of parameters is maintained. If new parameters are added due to the update, these are configured with default values.

Contrary to the normal update, the update via the bootstrap loader deletes the complete Flash memory and the stored parameter settings are lost. This possibility to load the programme also works, if no executable software exists in the module, e.g. after a failed normal update.

It is recommendable in any case to safe the set of parameters as a DCF file before carrying out an update.

The programme which is to be loaded into the module is read in from a hex file (e.g. DCCON.HEX). This file must be selected via a corresponding dialogue box. Afterwards you can launch the update process.



CanOpen protocol

Overview

The communication in CanOpen networks is based upon CAN data packages the content (max. 8 bytes) and target address of which are used by the CanOpen protocol.

The main part of the communication is handled via the communications objects SDO (Service Data Object) and PDO (Process Data Object).

Each CanOpen participant has a reserve of variables and parameters that are arranged in an object directory with defined addresses and that can be read or written via the network.

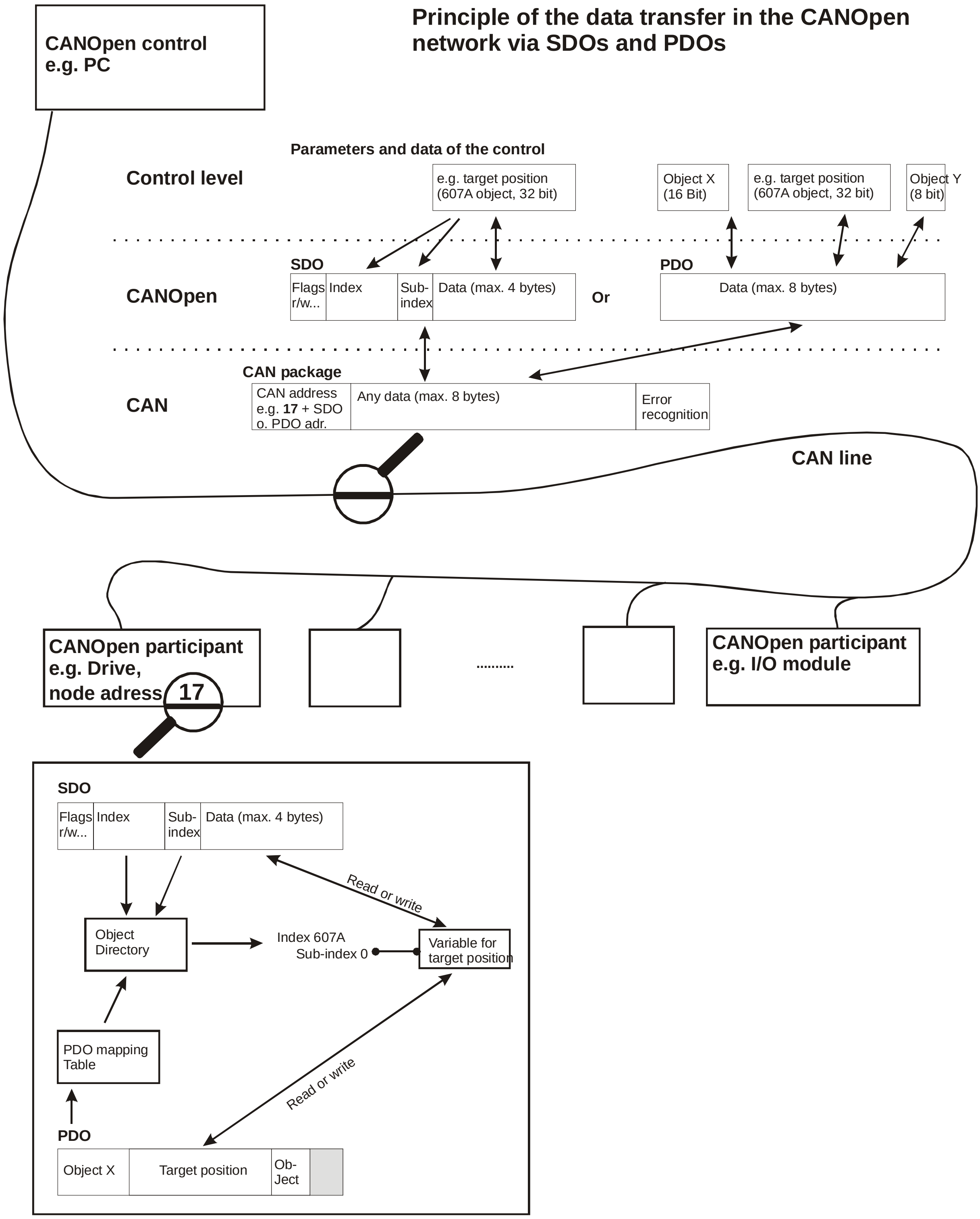
A CanOpen participant can be subdivided into three functional blocks:

Communication	Object directory	Application
Sending and receiving of communication objects. SDO, PDO, SYNC, etc.	Administration of communications parameters. Interface to internal variables and parameters.	Application programme, e.g. control of a drive, I/O administration. Implementation of the device profile.

Apart from the two objects for the data transfer, there are still further communications objects, e.g. for the synchronisation or error messages. All in all, the following CanOpen objects are supported:

Communications object	Short description
SDO	Universal communications channel for the reading and writing of all objects contained in an objects directory. Slower than PDO since the object address has always also to be transferred and since a feedback via a second SDO has to be realised.
PDO	Communications channel for the exchange of process data. Fast transfer, since the data are sent without protocol overhead.
EMCY	Emergency object for the transfer of error messages.
SYNC	The synchronisation object allows a synchronous operation of several bus participants.
NODE GUARDING	Monitoring of the bus participants through the exchange of cyclical messages.
NMT OBJECT	Control of the basic states of all bus participants.

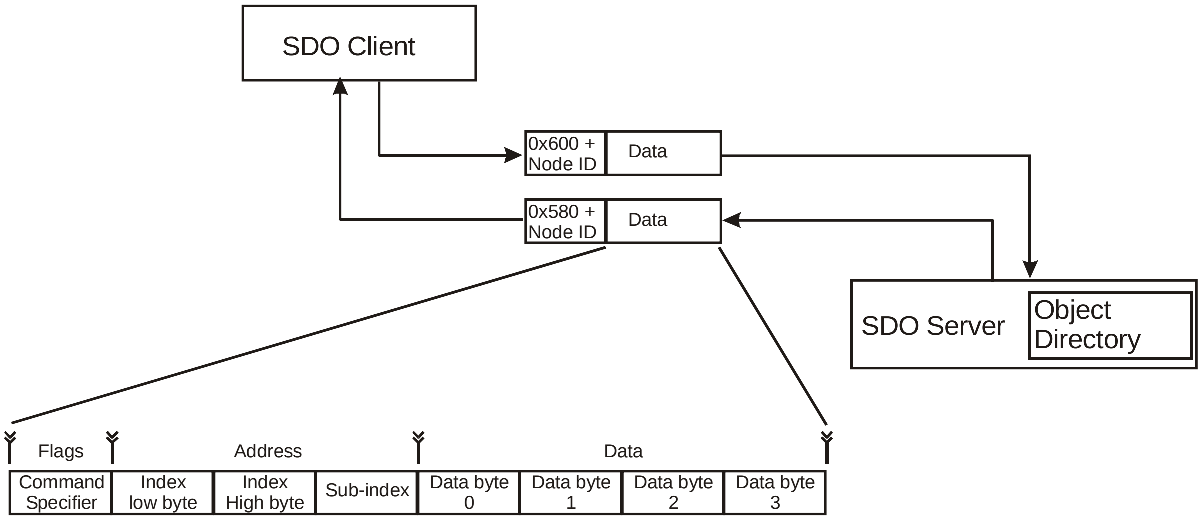
The following figure tries to display the communication with a CanOpen device via SDO or PDO.


Principle of the data transfer in the CANOpen network via SDOs and PDOs
CANOpen control e.g. PC
Parameters and data of the control
Control level
e.g. target position (607A object, 32 bit)
Object X (16 Bit)
e.g. target position (607A object, 32 bit)
Object Y (8 bit)
SDO
CANOpen
Flags r/w...
Index
Sub-index
Data (max. 4 bytes)
Or
PDO
Data (max. 8 bytes)
CAN package
CAN
CAN address e.g. 17 + SDO o. PDO adr.
Any data (max. 8 bytes)
Error recognition
CAN line
CANOpen participant e.g. Drive, node adress 17
....
CANOpen participant e.g. I/O module
SDO
Flags r/w...
Index
Sub-index
Data (max. 4 bytes)
Read or write
Object Directory
Index 607A Sub-index 0
Variable for target position
PDO mapping Table
Read or write
PDO
Object X
Target position
Ob-Ject

SDO

The Service Data Object (SDO) makes it possible to access the object directory of a CanOpen device. An SDO always uses two CAN objects with different IDs, since this protocol is always confirmed. An SDO creates a communications channel between two CanOpen participants. The device the object directory of which is accessed is the server of this SDO.

Each CanOpen device should dispose of a default SDO. The identifier of the Can objects related to this default SDO result from a defined Can address plus the node number (node ID) of the device.



The content of the first byte of an SDO (Command Specifier) controls the communication and defines e.g., if an object is read or written.

An SDO message always consists of 8 bytes, irrespective of the fact, how many data bytes are really transferred. The unused data bytes can contain any values and have to be ignored. The number of data bytes can be given in the Command Specifier; it is, however, also defined by the data type of the object to be transferred.

Basically, there are two different types of an SDO transfer:

- parameters with a length of 1-4 bytes are transferred with an SDO telegram (Expedited Transfer).
- data with a length of more than 4 bytes are transferred in several successive SDO telegrams (Normal Transfer).

In case of the UVE8112 drive module, all parameters can be transferred in the framework of the “Expedited Transfer”, with the exception of the objects 1008_h (device name), 1009_h (hardware version), $100A_h$ (software version) and 2081_h (trace data). The objects given do not have to be used in the normal operation.

Overview of the possible values of the command specifiers in case of the "Expedited Transfer":

<table>
<tr><th colspan="3">Requirement (Client)</th><th colspan="3">Response (Server)</th></tr>
<tr><td rowspan="6">Read object (upload request)</td><td colspan="2">Command specifier</td><td rowspan="6">Content of the object (upload response)</td><td>Command specifier</td><td>Number of data bytes</td></tr>
<tr><td colspan="2" rowspan="5">0x40</td><td>0x42</td><td>Not specified</td></tr>
<tr><td>0x4F</td><td>1</td></tr>
<tr><td>0x4B</td><td>2</td></tr>
<tr><td>0x47</td><td>3</td></tr>
<tr><td>0x43</td><td>4</td></tr>
<tr><td rowspan="6">Write object (download request)</td><td>Command specifier</td><td>Number of data bytes</td><td rowspan="6">Confirmation (download response)</td><td colspan="2">Command specifier</td></tr>
<tr><td>0x22</td><td>No data</td><td colspan="2" rowspan="5">0x60</td></tr>
<tr><td>0x2F</td><td>1</td></tr>
<tr><td>0x2B</td><td>2</td></tr>
<tr><td>0x27</td><td>3</td></tr>
<tr><td>0x23</td><td>4</td></tr>
<tr><td rowspan="2">Interruption of the SDO communication</td><td colspan="5">Command specifier</td></tr>
<tr><td colspan="5">0x80
Data bytes include an error code indicating the reason for the interruption.</td></tr>
</table>

The SDO transfer can be interrupted by a participant for different reasons. The following error codes can be indicated by the drive module.

SDO interruption error code	**Meaning**
0x05030000	Toggle Bit was not changed
0x05040000	SDO protocol timeout exceeded
0x05040001	Command specifier invalid or unknown
0x06010001	Read access to write-only object
0x06010002	Write access to read-only object
0x06020000	Object not existing in the object directory
0x06040041	Mapping for this object is admissible
0x06040043	Incompatibility of a parameter
0x06060000	Hardware error
0x06070012	Data type is not correct. The service parameter is too long
0x06090011	Sub-index not existing
0x06090030	Range of values of the parameter exceeded
0x06090031	Parameter value too high
0x06090032	Parameter value too small
0x06090042	Number and length of the parameters to be mapped exceeds the PDO length
0x08000000	General error
0x08000022	Parameter cannot be written or stored due to the current device status (mode of operation, etc.)

Example 1:

Query of the actual position (object 6064_h). As a response, the value 16900000_d = $0101DFA0_h$ is sent back.

	CS	Address			Data			
Client → Server	0x40	0x64	0x60	0x00	0x00	0x00	0x00	0x00
Server → Client	0x43	0x64	0x60	0x00	0xA0	0xDF	0x01	0x01

Example 2:

Set the control word (object 6040_h) to the value: $001F_h$.

	CS	Address			Data			
Client → Server	0x22	0x40	0x60	0x00	0x1F	0x00	0x00	0x00
Server → Client	0x60	0x40	0x60	0x00	0x00	0x00	0x00	0x00

PDO

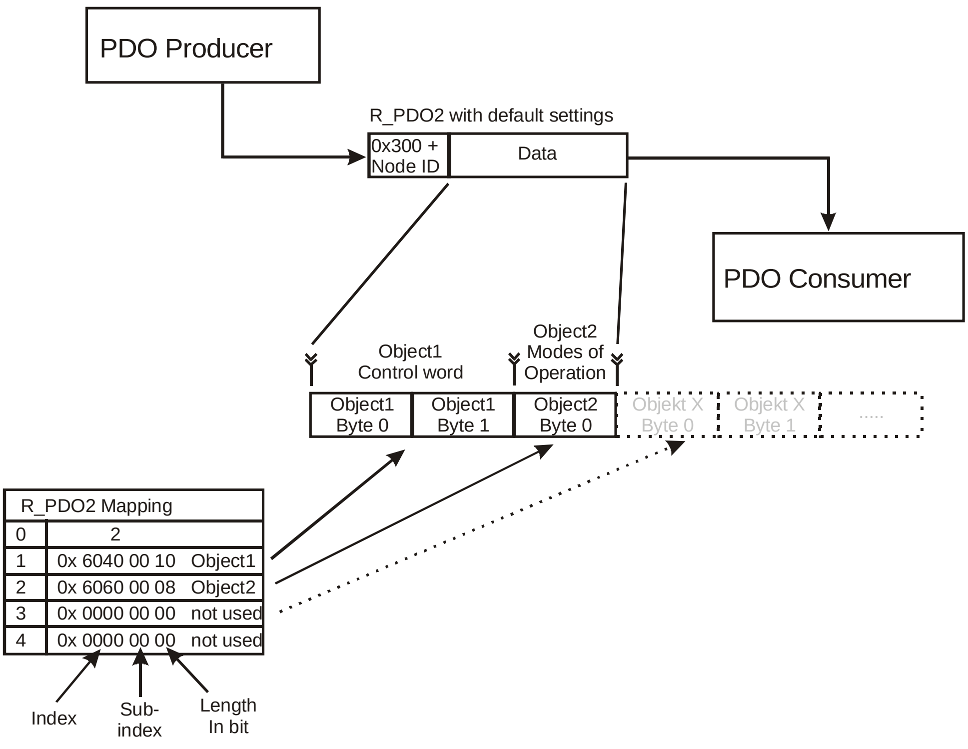
A Process Data Object (PDO) is a Can message with a defined identifier containing one or several objects with address information. The recipient (several recipients possible) knows the arrangement of the objects in this Can frame due to the Mapping table for this PDO in its object directory. The table is either preset firmly or written before the sending of the PDOs.

The receipt of a PDO is not confirmed by the recipients. The length of a PDO (1-8 bytes) depends on the number and length of the contained objects.

Mapping

A CanOpen device can dispose of several Receive PDOs and Transmit PDOs. The Mapping for the first R_PDO is contained in the object 1600h; for the next R_PDOs in the objects 1601h, 1602h, etc., the T_PDO mapping parameters as of Index 1A00h are given in the object directory.

In the figure, the default arrangement of the objects in the Receive PDO2 of the drive module is illustrated. The mapping can be changed or extended at all times. For all PDOs of the module, it is possible to map up to 4 objects.



Procedure for the setting of the mapping:

The setting of the mapping is carried out via SDO.

The entry under sub-index 0 in the mapping table defines the number of active objects in the PDO. Before the mapping can be changed, sub-index 0 **must** be set to 0 in order to deactivate the mapping.

Afterwards, the mapping entries can be described as of sub-index 1.

At the end, the corresponding number of objects is once again entered in the sub-index 0. If an object cannot be mapped or if the length of the PDO is exceeded, the SDO transfer is interrupted with a corresponding error message.

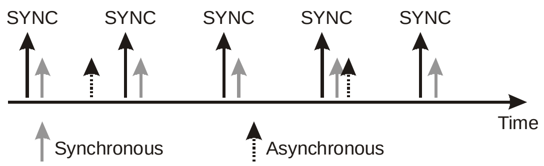
Types of transfer

The transfer of PDOs can be triggered by three different events:

- by an internal event (e.g. change of a status) or an internal timer.
- by the query of the PDO via a remote request (RTR Can object).
- by the receipt of a SYNC message.

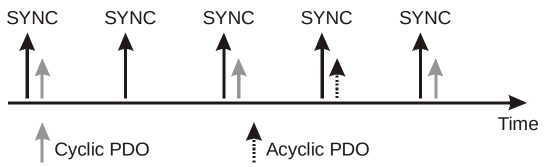
Furthermore, two types of transfer are distinguished:

- synchronous transfer. Synchronous PDOs are transferred directly after the SYNC object.
- asynchronous transfer. Asynchronous PDOs can be transferred at any given point in time.



The transfer of synchronous PDOs can again be subdivided into two types:

- Cyclic PDOs are transferred periodically according to an adjustable number of SYNC impulses (1-240).
- Acyclic PDOs are released by an internal event. They are transferred synchronously with the SYNC signal, but not periodically.



The setting of the type of transfer of PDOs is made via PDO communications parameters that are given in the object directory for R_PDOs as of index 1400h and for T_PDOs as of index 1800h. The following table shows an overview of the types of transfer and allocation to the parameter transmission type.

Type no.	Cyclic	Acyclic	Synchronous	Asynchronous	RTR only	Description
0		x	x			Transfer to SYNC, but not periodical
1-240	x		x			TPDO: transfer after every xth SYNC. RPDO: application of the data at the xth SYNC.
241-251						Reserved.
252			x		x	Objects in the PDO are updated with every SYNC, but only sent upon demand (RTR).
253				x	x	Objects in the PDO are updated and sent when the RTR is received.
254				x		Not used.
255				x		Event-controlled PDO. The transfer is triggered by a change of the mapped parameter or an adjustable timer.

SYNC

The SYNC signal provides a common cycle which can be used by all connected participants e.g. for the PDO transfer. The SYNC object is sent by a CanOpen participant and received by all devices participating in a synchronous operation.

The identifier of the SYNC object is by default set to 0x80 and has thus a very high priority within the Can network. No data bytes are transferred with the SYNC.

The SYNC signal plays an important role for the drive module in the Interpolation mode of operation. The superordinate control must communicate to the module beforehand the time interval of the SYNC signal; for this, the value (in µs) in the object "Communication Cycle Period" 1006_h has to be entered.

EMCY

Emergency messages are triggered by internal errors in the device and sent with a high priority to the Can bus. A superordinate control can thus react very quickly to an abnormal behaviour of individual components.

By default, the emergency message is characterised by the identifier 0x80 + node number.

With the emergency object, 8 data bytes are transferred for the error description. The structure of an emergency message looks as follows:

ID	2 bytes	1 byte	5 bytes
0x080 + Node-ID	Emergency error code EEC	Error register	Manufacturer-specific error field MEC

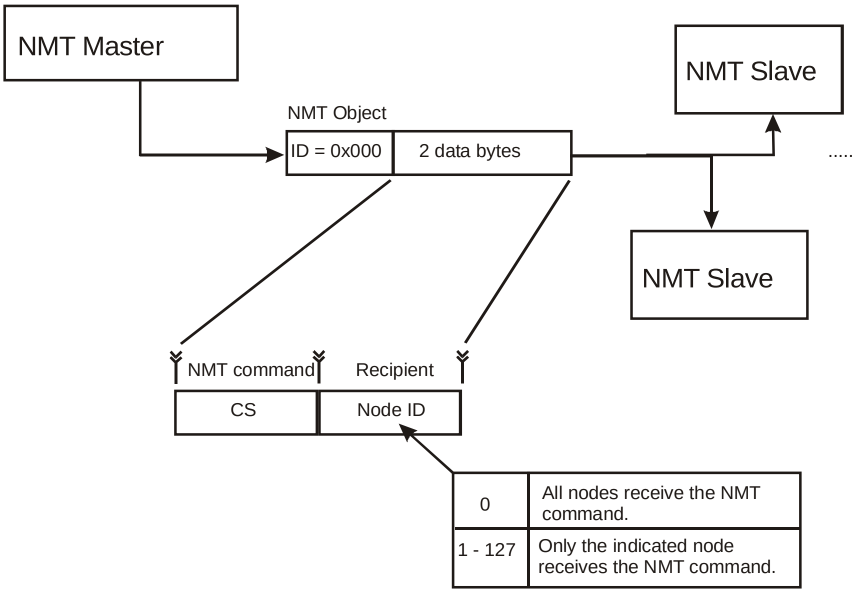
The first two bytes contain the error code of the fault that occurred. In the 3rd byte, the error register (object 1001_h) is transferred which only includes a rough classification of the type of error. The last 5 bytes may contain manufacturer-specific error information.

See Fault states on page 30.

Network management - NMT

The network management administers the communication’s basic functions of the participants in the CanOpen network. Here, a master-slave configuration is taken as a basis, in which an NMT master controls and regulates the state of all other participants (NMT slaves).

Status changes of individual or all NMT slaves (Start, Stop, Reset...) are triggered by an NMT object with the Can identifier 0x000. This object is sent by the NMT master and always possesses two data bytes. The first byte contains a command code determining the state of the NMT slave. The second data byte defines, if all slaves are addressed at the same time or if individual nodes only are used.

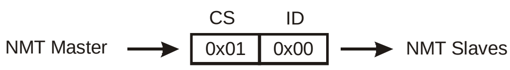


Command code (CS) of the NMT object

CS	Meaning	
0x01	“Start Remote Node“	Activate the Operational state
0x02	“Stop Remote Node“	Activate the Stopped state
0x80	“Enter Pre-Operational“	Activate the Pre-Operational state
0x81	“Reset Node“	Reset of all parameters and restart
0x82	“Reset Communication“	Reset of the communications parameters and restart

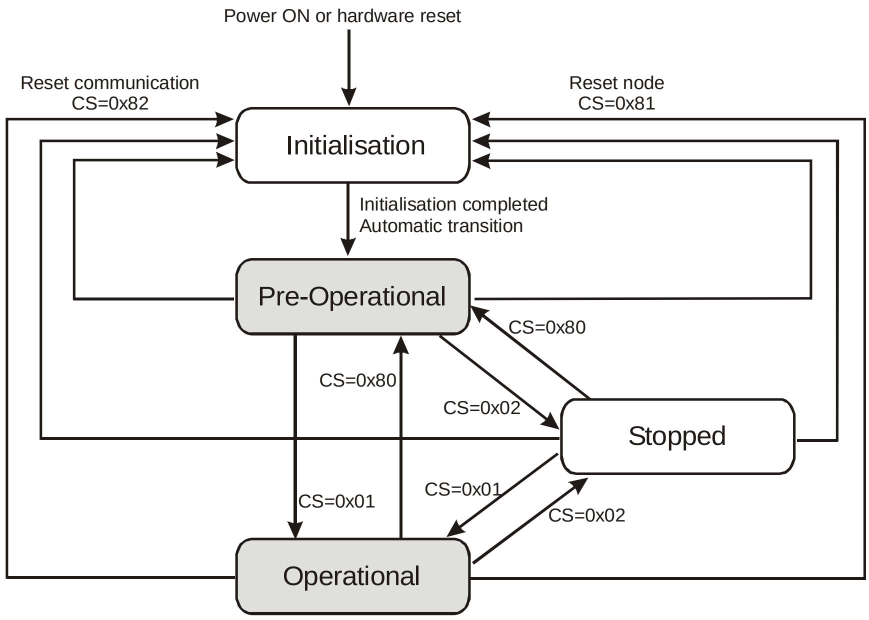
Example:

Bring all nodes into the Operational state



Status diagram

Every CanOpen participant disposes of an internal state machine which consists of four states and controls the communication behaviour of the node. The status transitions are triggered by internal events or by the receipt of NMT objects.



In the "Pre-Operational" state it is not possible to send or receive PDOs. In the "Stopped" state, no communication whatsoever is possible with the node, except for the Guarding.

Effects of the states on the processing of communications objects:

	Initialisation	**Pre-Operational**	**Operational**	**Stopped**
PDO			X	
SDO		X	X	
SYNC object		X	X	
EMCY object		X	X	
Boot-up object	X			
NMT objects inc. Node Guarding		X	X	X

Boot-Up object

After a successful initialisation, the module sends a boot-up object which notifies the NMT master of the fact that the node is available and has achieved the Pre-Operational state. The boot-up object uses the identifier of the guarding object and contains a data byte with the firm value 0.

ID	Byte 0
0x700 + Node ID	0x00

Monitoring – Guarding

The guarding includes the monitoring of all NMT slaves by the master (node guarding) as well as the automatic self-interruption of the slaves in case of a failure of the NMT master (life guarding).

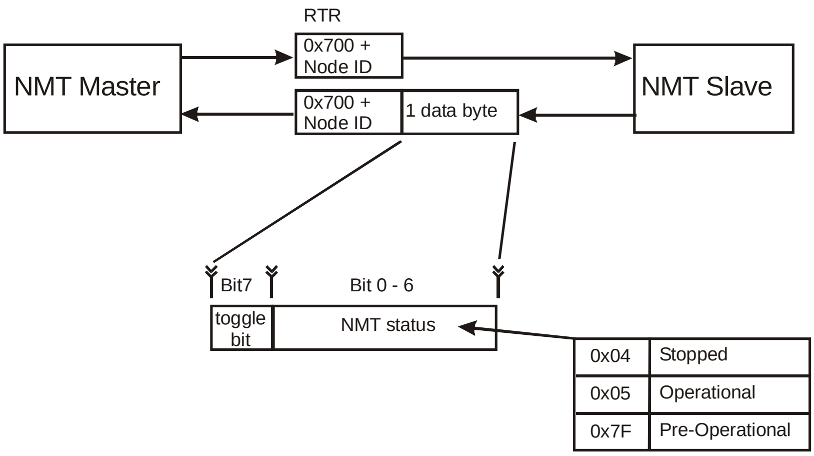
For this purpose, the master sends in regular time intervals (guard time), which can be set separately for every node, a request telegram (RTR) for the guarding object of the respective slave.

The NMT slave replies within the life time (= guard time x life time factor) with the guarding object, which contains – in one data byte - the current NMT state and a toggle bit.

Upon the first query, the toggle bit has a value of 0 and changes its value (0, 1, 0, 1…) with the following guarding telegrams. If, within the life time, no response with the correct toggle bit value is sent back or if the NMT state does not match the expected state, the master has to assume an error.

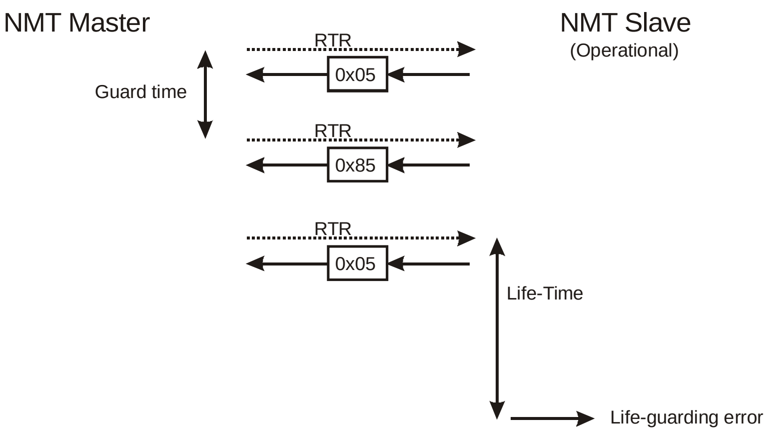
The guarding for a node is activated with the first request of the guarding object by the master and can only be switched off again through a reset of the node.

The NMT slave monitors the incoming queries of the master and switches to the fault state, if the telegrams for the life-time period fail to appear.



Example:

The module is in the Operational state. After the master does no longer sent a guarding request, the module changes to the fault state, when the life time elapsed.



Object Directory

The object directory contains all parameters and variables of the module that can be read or written via the Can network. The entries in the directory are characterised by a 16-bit index and a 8-bit sub-index.

Communications parameter

The range of 1000h to 1FFFh is reserved for the communication parameters of a CanOpen device. The communications part of the drive module software is structured according to the specification DS301 V4.0. In the following, all parameters are listed and described that are used for this and that control the communications behaviour.

General communications parameters

Device type

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
1000	00	Device type	Unsigned 32	RO	N	0x00020192	Device type

The device type indicates which device profile is used and which kind of device this is (e.g. servo drive = 02_h). Bit 0-15: device profile 402 = $192_{h.}$

Error register

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
1001	00	Error register	Unsigned 8	RO	Y	0x00	Error register

The error register contains a rough classification of an error that occurred. It is also sent as part of the emergency message. A more precise designation of the error is found in the error memory (1003_h).

Bit 7	6	5	4	3	2	1	0
-	-	Dev. prof.-specific	Commu-nication	-	Voltage	Current	general error

Manufacturer-specific status register

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
1002	00	Manufacturer-specific status register	Unsigned 32	RO	Y	0x00	Manufacturer-specific status register

Is currently not used.

Error memory

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
1003		Pre-defined error field	RECORD				Error memory (list of the last errors that occurred).
1003	00	Number of errors	Unsigned 8	RW	N	0	Number of the errors occurred.
1003	01	Standard error field	Unsigned 32	RO	N	0x00000000	Current error.
1003	02	Standard error field	Unsigned 32	RO	N	0x00000000	Last but one error.
1003	03	Standard error field	Unsigned 32	RO	N	0x00000000	Second from the last error.
1003	04	Standard error field	Unsigned 32	RO	N	0x00000000	Third from the last error.
1003	05	Standard error field	Unsigned 32	RO	N	0x00000000	A maximum of 5 errors is stored.

The error memory includes the errors that occurred in the device and where fed back via emergency object.

- The sub-index 0 contains the number of registered errors.
- Each new error is saved; the preceding errors are moved downwards in the list.
- The writing of an "0" on the sub-index 0 deletes the error memory.
- The 32-bit long error entries consist of the error code (EEC, see emergency object) in the lower 2 bytes and additional information in the upper 2 bytes (MEC).

Upper 16 bits	Lower 16 bits
Additional information	Error code

COB ID Sync message

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
1005	00	COB ID Sync message	Unsigned 32	CONST	N	0x00000080	Identifier of the SYNC message.

The lower 11 bits include the identifier of the Sync message which the module can receive.

SYNC interval

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
1006	00	Communication cycle period	Unsigned 32	RW	N	0	Length of the SYNC interval in microseconds.

Interval between two successive Sync telegrams. For the Interpolation mode of operation, the exact value in µsec has to be entered. In the other modes of operation, this parameter is not evaluated.

Device name

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
1008	00	Manufacturer's device name	Visible string	RO	N		Device name.

Hardware version

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
1009	00	Manufacturer's hardware version	Visible string	RO	N		Hardware version number.

Software version

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
100A	00	Manufacturer's software version	Visible String	RO	N		Software version number.

Node number

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
100B	00	Node ID	Unsigned 32	RO	N		Node number.

Guard time

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
100C	00	Guard time	Unsigned 16	RW	N	200	Interval between two guard telegrams in milliseconds.

See "Monitoring – Guarding" on page 76.

Life-time factor

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
100D	00	Life-time factor	Unsigned 8	RW	N	3	See below.

Life-time factor x guard time = time, until the module switches to the fault state, if guard telegrams fail to appear.

Store parameters

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
1010		Store parameters	RECORD				Store parameters.
1010	00	Largest supported sub-index	Unsigned 8	RO	N	1	Number of types of memory.
1010	01	Save all parameters	Unsigned 32	RW	N	0x00000001	Store all parameters.

By writing the four characters 's' 'a' 'v' 'e' in the ASCII code (0x 65 76 61 73) on the sub-index 1, all storable parameters of the module are saved permanently and internally.

This comprises:

- Communication parameter: PDO parameters (except PDO-ID), PDO mapping, guarding parameters, Sync interval.

- All writable device profile parameters except for Control word, Interpolation Data and Interpolation-Actual-Buffer-Size.
- Manufacturer-specific parameters: offset values of the analogue inputs.

Load default parameters

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
1011		Restore default parameters	RECORD				Load default parameters.
1011	00	Largest supported sub-index	Unsigned 8	CONST	N	1	Number of options.
1011	01	Restore all default parameters	Unsigned 32	RW	N	0x00000001	Load all parameters with default values.
1011	02	Restore communication default parameters	Unsigned 32	RW	N	0x00000001	Load communications parameters with default values.

By writing the four characters 'l' 'o' 'a' 'd' in the ASCII code (0x 64 61 6F 6C) on the sub-index 1, all parameters are set to their default values. By writing the signature on sub-index 2, only communications parameter with default values are loaded.

High-resolution time stamp

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
1013	00	High-resolution time stamp	Unsigned 32	RW	Y	0	High-resolution time stamp (µs).

Synchronisation of the interpolation cycle in the interpolation mode 2.

COB ID Error message

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
1014	00	COB ID emergency message	Unsigned 32	RO	N	0x080 + Node ID	Identifier of the error message.

Identity object

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
1018		Identity object	RECORD				SDO parameters of the module.
1018	00	Number of entries	Unsigned 8	CONST	N	3	Number of entries.
1018	01	Vendor ID	Unsigned 32	RO	N	0x00000031	Registered manufacturer ID.
1018	02	Product code	Unsigned 32	RO	N	0x00DC8112	Product number.
1018	03	Revision number	Unsigned 32	RO	N		Version number.

SDO parameters

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
1200		Server SDO parameters	RECORD				SDO parameters of the module.
1200	00	Number of entries	Unsigned 8	CONST	N	2	Number of entries.
1200	01	COB ID Client->Server (rx)	Unsigned 32	RO	N	0x600 + Node ID	Identifier of the Can object which is sent to the module.
1200	02	COB ID Server->Client (tx)	Unsigned 32	RO	N	0x580 + Node ID	Identifier of the Can object which is sent by the module.

See “SDO" on page 70.

PDO / Mapping parameters

The drive controller supports 4 Receive and 4 Transmit PDOs with variable mapping of up to 4 objects per PDO. The type of transfer can be any possible way admissible in CanOpen. The identifier of the PDOs can be changed, but not stored. All other parameters (type of transfer, mapping) can be stored and must therefore not be reconfigured every time a reset has been made. See "PDO" on page 72 for a description of the types of transfer and the mapping procedure.

Receive PDO1 – communications parameters

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
1400		R_PDO1 parameters	RECORD				Comm. parameters.
1400	00	Number of entries	Unsigned 8	CONST	N	2	Number of entries.
1400	01	COB ID	Unsigned 32	RO	N	0x200+Node no.	Identifier of the RxPDO1.
1400	02	Transmission type	Unsigned 8	RW	N	1	Transfer type of the PDO.

Receive PDO2 – communications parameters

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
1401		R_PDO2 parameters	RECORD				Comm. parameters.
1401	00	Number of entries	Unsigned 8	CONST	N	2	Number of entries.
1401	01	COB ID	Unsigned 32	RO	N	0x300+Node no.	Identifier of the RxPDO2.
1401	02	Transmission type	Unsigned 8	RW	N	1	Transfer type of the PDO.

Receive PDO3 – communications parameters

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
1402		R_PDO3 parameters	RECORD				Comm. parameters of the 3rd receive PDO.
1402	00	Number of entries	Unsigned 8	CONST	N	2	Number of entries.
1402	01	COB ID	Unsigned 32	RO	N	0x400+Node no.	Identifier of the RxPDO3.
1402	02	Transmission type	Unsigned 8	RW	N	1	Transfer type of the PDO.

Receive PDO4 – communications parameters

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
1403		R_PDO4 parameters	RECORD				Comm. parameters.
1403	00	Number of entries	Unsigned 8	CONST	N	2	Number of entries.
1403	01	COB ID	Unsigned 32	RO	N	0x500+Node no.	Identifier of the RxPDO4.
1403	02	Transmission type	Unsigned 8	RW	N	1	Transfer type of the PDO.

Receive PDO1 – mapping parameters

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
1600		R_ PDO1 mapping	RECORD				Mapping parameters.
1600	00	No. of obj. in PDO	Unsigned 8	RW	N	1	Number of objects.
1600	01	Mapping for 1st obj.	Unsigned 32	RW	N	0x60400010	Mapping of the 1st object.
1600	02	Mapping for 2nd obj.	Unsigned 32	RW	N	0x00000000	Mapping of the 2nd object.
1600	03	Mapping for 3rd obj.	Unsigned 32	RW	N	0x00000000	Mapping of the 3rd object.
1600	04	Mapping for 4th Obj.	Unsigned 32	RW	N	0x00000000	Mapping of the 4th object.

Receive PDO2 – mapping parameters

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
1601		R_ PDO2 mapping	RECORD				Mapping parameters.
1601	00	No. of obj. in PDO	Unsigned 8	RW	N	2	Number of objects.
1601	01	Mapping for 1st obj.	Unsigned 32	RW	N	0x60400010	Mapping of the 1st object.
1601	02	Mapping for 2nd obj.	Unsigned 32	RW	N	0x60600008	Mapping of the 2nd object.
1601	03	Mapping for 3rd obj.	Unsigned 32	RW	N	0x00000000	Mapping of the 3rd object.
1601	04	Mapping for 4th obj.	Unsigned 32	RW	N	0x00000000	Mapping of the 4th object.

Receive PDO3 – mapping parameter

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
1602		R_ PDO3 Mapping	RECORD				Mapping parameters.
1602	00	No. of obj. in PDO	Unsigned 8	RW	N	2	Number of objects.
1602	01	Mapping for 1st obj.	Unsigned 32	RW	N	0x60400010	Mapping of the 1st object.
1602	02	Mapping for 2nd obj.	Unsigned 32	RW	N	0x607A0020	Mapping of the 2nd object.
1602	03	Mapping for 3rd obj.	Unsigned 32	RW	N	0x00000000	Mapping of the 3rd object.
1602	04	Mapping for 4th obj.	Unsigned 32	RW	N	0x00000000	Mapping of the 4th object.

Receive PDO4 – mapping parameters

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
1603		R_ PDO4 mapping	RECORD				Mapping parameters.
1603	00	No. of obj. in PDO	Unsigned 8	RW	N	2	Number of objects.
1603	01	Mapping for 1st obj.	Unsigned 32	RW	N	0x60400010	Mapping of the 1st object.
1603	02	Mapping for 2nd obj.	Unsigned 32	RW	N	0x60810020	Mapping of the 2nd object.
1603	03	Mapping for 3rd obj.	Unsigned 32	RW	N	0x00000000	Mapping of the 3rd object.
1603	04	Mapping for 4th obj.	Unsigned 32	RW	N	0x00000000	Mapping of the 4th object.

Transmit PDO1 – communication parameters

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
1800		T_PDO1 parameters	RECORD				Comm. parameters.
1800	00	Number of entries	Unsigned 8	CONST	N	2	Number of entries.
1800	01	COB ID	Unsigned 32	RO	N	0x180+Node no.	Identifier of the TxPDO1.
1800	02	Transmission type	Unsigned 8	RW	N	1	Transfer type of the PDO.
1800	03	Inhibit time	Unsigned 16	RW	N	50	Minimal sending interval for TxPDO1 (in [100 µs]).
1800	04	Compatibility entry	Unsigned 8	RW	N	0	No function.
1800	05	Event timer	Unsigned 16	RW	N	0	Timer for cyclic sending (unit: [1 ms]).

Transmit PDO2 – communication parameters

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
1801		T_PDO2 parameters	RECORD				Comm. parameters.
1801	00	Number of entries	Unsigned 8	CONST	N	2	Number of entries.
1801	01	COB ID	Unsigned 32	RO	N	0x280+Node no.	Identifier of the TxPDO2.
1801	02	Transmission type	Unsigned 8	RW	N	1	Transfer type of the PDO.
1801	03	Inhibit time	Unsigned 16	RW	N	50	Minimal sending interval for TxPDO2 (in [100 µs]).
1801	04	Compatibility entry	Unsigned 8	RW	N	0	No function.
1801	05	Event timer	Unsigned 16	RW	N	0	Timer for cyclic sending (unit: [1 ms]).

Transmit PDO3 – communication parameters

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
1802		T_PDO3 parameters	RECORD				Comm. parameters.
1802	00	Number of entries	Unsigned 8	CONST	N	2	Number of entries.
1802	01	COB ID	Unsigned 32	RO	N	0x380+Node no.	Identifier of the TxPDO3.
1802	02	Transmission type	Unsigned 8	RW	N	1	Transfer type of the PDO.
1802	03	Inhibit time	Unsigned 16	RW	N	50	Minimal sending interval for TxPDO3 (in [100 µs]).
1802	04	Compatibility entry	Unsigned 8	RW	N	0	No function.
1802	05	Event timer	Unsigned 16	RW	N	0	Timer for cyclic sending (unit: [1 ms]).

Transmit PDO4 – communication parameters

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
1803		T_PDO4 parameters	RECORD				Comm. parameters.
1803	00	Number of entries	Unsigned 8	CONST	N	2	Number of entries.
1803	01	COB ID	Unsigned 32	RO	N	0x480+Node no.	Identifier of the TxPDO4.
1803	02	Transmission type	Unsigned 8	RW	N	1	Transfer type of the PDO.
1803	03	Inhibit Time	Unsigned 16	RW	N	50	Minimal sending interval for TxPDO4 (in [100 µs]).
1803	04	Compatibility entry	Unsigned 8	RW	N	0	No function.
1803	05	Event timer	Unsigned 16	RW	N	0	Timer for cyclic sending (unit: [1 ms]).

Transmit PDO1 – mapping parameters

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
1A00		T_ PDO1 mapping	RECORD				Mapping parameters.
1A00	00	No. of obj. in PDO	Unsigned 8	RW	N	1	Number of objects.
1A00	01	Mapping for 1st obj.	Unsigned 32	RW	N	0x60410010	Mapping of the 1st object.
1A00	02	Mapping for 2nd obj.	Unsigned 32	RW	N	0x00000000	Mapping of the 2nd object.
1A00	03	Mapping for 3rd obj.	Unsigned 32	RW	N	0x00000000	Mapping of the 3rd object.
1A00	04	Mapping for 4th obj.	Unsigned 32	RW	N	0x00000000	Mapping of the 4th object.

Transmit PDO2 – mapping parameters

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
1A01		T_ PDO2 Mapping	RECORD				Mapping parameters.
1A01	00	No. of obj. in PDO	Unsigned 8	RW	N	2	Number of objects.
1A01	01	Mapping for 1st obj.	Unsigned 32	RW	N	0x60410010	Mapping of the 1st object.
1A01	02	Mapping for 2nd obj.	Unsigned 32	RW	N	0x60610008	Mapping of the 2nd object.
1A01	03	Mapping for 3rd obj.	Unsigned 32	RW	N	0x00000000	Mapping of the 3rd object.
1A01	04	Mapping for 4th obj.	Unsigned 32	RW	N	0x00000000	Mapping of the 4th object.

Transmit PDO3 – mapping parameters

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
1A02		T_ PDO3 mapping	RECORD				Mapping parameters of the 3rd Transmit PDO.
1A02	00	No. of obj. in PDO	Unsigned 8	RW	N	2	Number of objects.
1A02	01	Mapping for 1st obj.	Unsigned 32	RW	N	0x60410010	Mapping of the 1st object.
1A02	02	Mapping for 2nd obj.	Unsigned 32	RW	N	0x60640020	Mapping of the 2nd object.
1A02	03	Mapping for 3rd obj.	Unsigned 32	RW	N	0x00000000	Mapping of the 3rd object.
1A02	04	Mapping for 4th obj.	Unsigned 32	RW	N	0x00000000	Mapping of the 4th object.

Transmit PDO4 – mapping parameters

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
1A03		T_ PDO4 Mapping	RECORD				Mapping parameters.
1A03	00	No. of obj. in PDO	Unsigned 8	RW	N	2	Number of objects.
1A03	01	Mapping for 1st obj.	Unsigned 32	RW	N	0x60410010	Mapping of the 1st object.
1A03	02	Mapping for 2nd obj.	Unsigned 32	RW	N	0x606C0020	Mapping of the 2nd object.
1A03	03	Mapping for 3rd obj.	Unsigned 32	RW	N	0x00000000	Mapping of the 3rd object.
1A03	04	Mapping for 4th obj.	Unsigned 32	RW	N	0x00000000	Mapping of the 4th object.

Device profile parameters

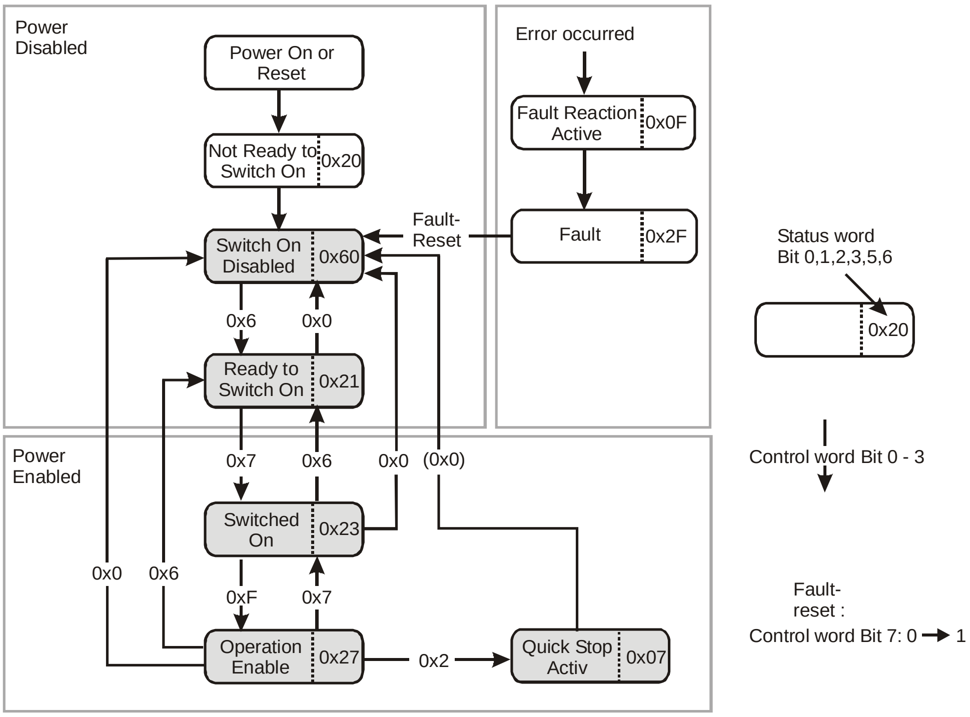
CanOpen comprises different devices (e.g. I/O modules, drives, encoders ...), so-called device profiles, in which pre-defined parameters and modes of operation are available for each device class. According to the functional scope of a device, more or less of these default objects can be implemented.

The drive module uses the DS402 V1.0 device module. In the following, all used parameters are listed and described according to functional groups. In this regard, all drive functions and the behaviour of the module are shown in the different modes of operation.

Device control

The drive module comprises a so-called state machine which defines which operating states may be applied and how the transition to other states can be triggered. This status diagram is controlled by a Control word or by an event (e.g. errors that occurred). Via the status word the current state can be queried.

State machine



Description of the individual states:

- **Not Ready to Switch On:** initialisation.
- **Switch On Disabled:** Initialisation completed, final stage blocked.
  Drive functions blocked. Brake in the automatic operation mode active.
- **Ready to Switch On:** Final stage blocked. Drive functions blocked.

  Brake in the automatic operation mode active.
- **Switched On:** Final stage released. Drive functions blocked.

  Brake in the automatic operation mode active.
- **Operation Enable:** Final stage released. Drive functions enabled. Motor is live.
- **Quick Stop Active:** Quick Stop function (braking with maximum ramp) is realised. The motor is live. After the motor achieved its standstill, automatic transition to the Switch On Disabled state.
- **Fault Reaction Active:** An error occurred. The motor is braked with the Quick-Stop ramp, then transfer to the Fault state. In case of critical faults, the system automatically switches to the Fault state. Brake in the automatic mode not active.

- **Fault:** Final stage blocked. Drive functions blocked.

Control word

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
6040	00	Control word	Unsigned 16	RW	Y	0x00	Control of all important drive functions.

Meaning of the bits in the control word

Bit	All modes of operation	Mode of operation: "Velocity control"	Mode of operation: "Profile position mode"	Mode of operation: "Homing mode"	Mode of operation: "Interpolation"
0-3	Control of the state machine.				
4	-	-	Apply new set of position.	Start reference run.	Release of interpolation mode.
5	-	-	The set of position immediately becomes valid.	-	-
6	-	-	0 = absolute / 1 = relative	-	-
7	Reset fault.				
8	Stop	Braking with normal ramp.	Interrupt positioning. (motion vector is maintained)	Interrupt reference run.	-
9-10	Not used.				
11	-	-	Cancel positioning. (motion vector is lost)	Cancel reference run.	-
12-15	Not used.				

Status word

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
6041	00	Status word	Unsigned 16	RO	Y	0x00	Status of all important drive functions.

Meaning of the bits in the status word

Bit	All modes of operation	Mode of operation: “Velocity control”	Mode of operation: “Profile position mode”	Mode of operation: “Homing mode”	Mode of operation: “Interpolation”
0-3	Status of the state machine.				
4	Final stage blocked (no enabling signal at input 4).				
5-6	Status of the state machine.				
7	I²t – current limitation is enabled.				
8	Main supply voltage is missing.				
9	Release of the controller via control word possible.				
10	Target value achieved.	Speed achieved.	Position achieved.	-	-
11	-				
12	-	Speed = 0	Set of position applied.	Reference run completed.	Interpolation released.
13	-	-	Following error.	Error during reference run.	Following error.
14	-	-	Motion active.	Motion active.	Buffer full.
15	-	-	-	-	Interpolation stopped (missing SYNC or buffer empty).

Mode of operation

The setting of the mode of operation is realised via the 6060h object. This parameter can only be written. The query of the active mode of operation is carried out via the 6061h object.

The following modes of operation can be set in the module via these parameters:

Mode of operation	Parameter value
Moving away from the limit switch	-3
Velocity control, analogue input	-2
Current controller, analogue input	-1
Positioning (profile position mode)	1
Velocity control (profile velocity mode)	3
Reference run (homing mode)	6
Interpolation (interpolated position mode)	7

Modes of operation

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
6060	00	Modes of operation	Signed 8	WO	Y	1	Setting of the mode of operation.

Display of the mode of operation

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
6061	00	Modes of operation display	Signed 8	RO	Y	1	Display of the current mode of operation.

Current control

The following objects refer to the current settings and the regulation of the motor current. During the commissioning, these parameters have to be set according to the motor used and the desired behaviour. For this, see "Current Settings dialogue box" on page 48.

Maximum current

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
6073	00	Maximum current	Unsigned 16	RW	N	12000	Maximum output current in mA.

The maximum current can amount to values between 1,000 and 25,000 (mA).

Rated current

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
6075	00	Motor’s rated current	Unsigned 32	RW	N	3000	Rated current in mA.

The admissible settings for the rated current are between 0 to 10,000 mA.

I^2t – turn-off time

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
6510	06	IIt – current limitation time	Unsigned 16	RW	N	500	I^2t – current limitation time in ms.

Values between 10 and 500 (ms) can be set.

Actual current

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
6078	00	Current: actual value	Signed 16	RO	Y	0	Actual current value (internal unit).

Current controller parameters

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
60F6		Torque control parameters	RECORD				Parameters of the current controller.
60F6	00	Number of entries	Unsigned 8	CONST	N	2	Number of entries.
60F6	01	kp	Unsigned 16	RW	N	8	Proportional amplification.
60F6	02	ki	Unsigned 16	RW	N	30	Integral amplification.
60F6	03	Fast sample	Unsigned 8	RW	N	0	Defines the limitation time for the current control. 0: Default 1: Fast

See "Current Controller" on page 49.

Profile Velocity Mode – velocity control with ramp profile

The objects given here are important for the velocity control mode of operation. Furthermore, the **Acceleration** parameter (6083$_h$, see Positioning mode) is used in this mode of operation.

The two parameters **Target velocity** and **Acceleration** define the speed profile which is implemented by the drive. As soon as the target speed parameter changes its value, the motor is accelerated with the given acceleration or braked, until the new target speed value is achieved. In the **Status word**, Bit10 (Target Reached) is set, if the actual velocity remains in the **Velocity Window** for the **Velocity Window Time**.

Target velocity

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
60FF	00	Target velocity	Signed 32	RW	Y	0	Target velocity in the profile velocity mode in user-defined units / s.

Actual velocity

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
606C	00	Velocity: actual value (user-defined units / s)	Signed 32	RO	Y	0	Actual velocity in user-defined units / s.

Actual target velocity value

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
606B	00	Velocity: target value (Incr./s)	Signed 32	RO	Y	0	Actual target velocity value in Incr./s.

Actual velocity sensor value

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
6069	00	Velocity sensor: actual value	Signed 32	RO	Y	0	Actual velocity in Incr/s.

Velocity sensor selection

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
606A	00	Sensor selection: Code	Signed 16	RO	N	0	Type of the velocity sensor (0 = velocity is determined on the basis of the position encoder).

Velocity window

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
606D	00	Velocity window	Unsigned 16	RW	N	100	Velocity window for the achievement of the target velocity in user-defined units / s.

Velocity window time

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
606E	00	Velocity window time	Unsigned 16	RW	N	1	Time frame for the achievement of the target velocity in milliseconds.

Velocity controller parameters

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
60F9		Velocity control parameters	RECORD				Parameters of the velocity controller.
60F9	00	Number of entries	Unsigned 8	CONST	N	6	Number of entries.
60F9	01	kp	Unsigned 16	RW	N	50	Proportional amplification.
60F9	02	ki	Unsigned 16	RW	N	5	Integral amplification.
60F9	03	kd	Unsigned 16	RW	N	1000	Differential amplification.
60F9	04	td	Unsigned 8	RW	N	0	Scan time of the D proportion in units of the controller scan time.
60F9	05	e_limit	Unsigned 16	RW	N	400	Following limitation.
60F9	06	hard_limit	Unsigned 8	RW	N	0	Method of the following limitation.

For the setting of the velocity controller, see "Velocity Controller dialogue box" on page 54. For the optimisation of the following limitation value, see “Current Controller dialogue box” on page 49.

Profile Position Mode – position control with ramp profile

In the Profile Position Mode, a motion segment can be allocated to the drive module which the controller automatically realises. During the processing of a segment, the parameters for the next movement can already be set, so that individual segments can be lined up seamlessly.

The setting of the individual movement segments is realised via the parameters **Target position, Segment velocity, Final velocity** and **Acceleration**, the same values being used for acceleration and braking ramps.

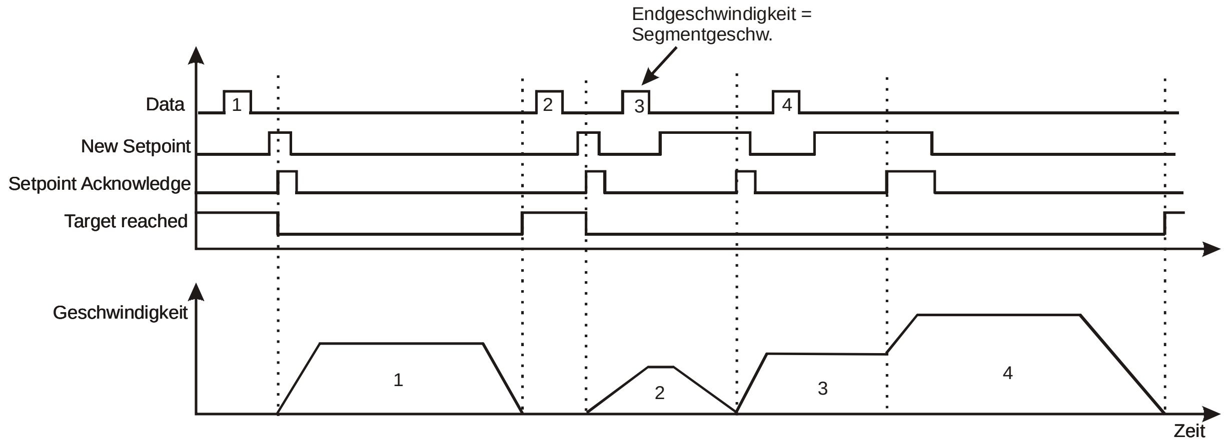
Via **Control word** and **Status word** a handshake is made during the transfer of the segments.

In the following illustration, 4 movement segments are executed after each other.

The data for the second segment are transferred only, after the first segment was completed. After the transfer of the motion parameter, the respective Bit 4 (New Setpoint) is set in Control word in order to tell the drive to apply the new movement parameters and to launch the movement. The controller confirms the acknowledgement with the Bit 12 in the Status word (Setpoint Acknowledge). If Bit 4 is reset in the control word, the drive replies with the revocation of the Setpoint Acknowledge and signalises thus that it is prepared to transfer further data.

After the second movement segment was launched, the master immediately begins with the transfer of data for the third movement segment and sets Bit 4 in the control word. The drive module applies the data at the end of the current segment and simultaneously starts the new movement.

Likewise, Segment 4 is directly put after Segment 3; the drive, however, does not brake, since in Segment 3 the final velocity is identical to the segment velocity.



Data: Daten für Bewegungssegment schreiben
New Setpoint: Controlword Bit 4
Setpoint Acknowledge: Statusword Bit 12
Target reached: Statusword Bit 10

If, in addition to Bit 4, Bit 5 is set in the control word (Change Set Immediately), the current movement is interrupted and the execution of a new segment is immediately started with. Here, the system accelerates or brakes down to the new segment velocity according to the ramp set.

Target position

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
607A	00	Target position	Signed 32	RW	Y	0	Target position of a movement segment in user-defined units.

Segment velocity

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
6081	00	Profile velocity	Unsigned 32	RW	N	10000	Velocity during the movement segment in user-defined units / s.

Final velocity

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
6082	00	End velocity	Unsigned 32	RW	N	0	Velocity at the end of the movement segment in user-defined units / s.

Acceleration

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
6083	00	Profile acceleration	Unsigned 32	RW	N	100000	Acceleration in user-defined units / s².

Actual position target value

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
60FC	00	Position demand value (Incr.)	Signed 32	RO	Y	0	Actual position target value.

Actual position (Incr.)

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
6063	00	Actual Position (Incr.)	Signed 32	RO	Y	0	Actual position value in increments.

Actual position (user-defined units)

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
6064	00	Actual position (user-defined units)	Signed 32	RO	Y	0	Actual position value in user-defined units.

Following error Position window

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
6065	00	Following error window	Unsigned 32	RW	Y	1000	Position window for the monitoring of the following error [Incr.].

The monitoring of the following error is realised by means of the two parameters “Following error position window” and "Following error time window". The following error is indicated in the status word (Bit 13), if the actual position deviates from the target value for the "Following error time window" by the "Following error position window”.

Following error time window

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
6066	00	Following error timeout	Unsigned 16	RW	Y	10	Time window for the monitoring of the following error [milliseconds].

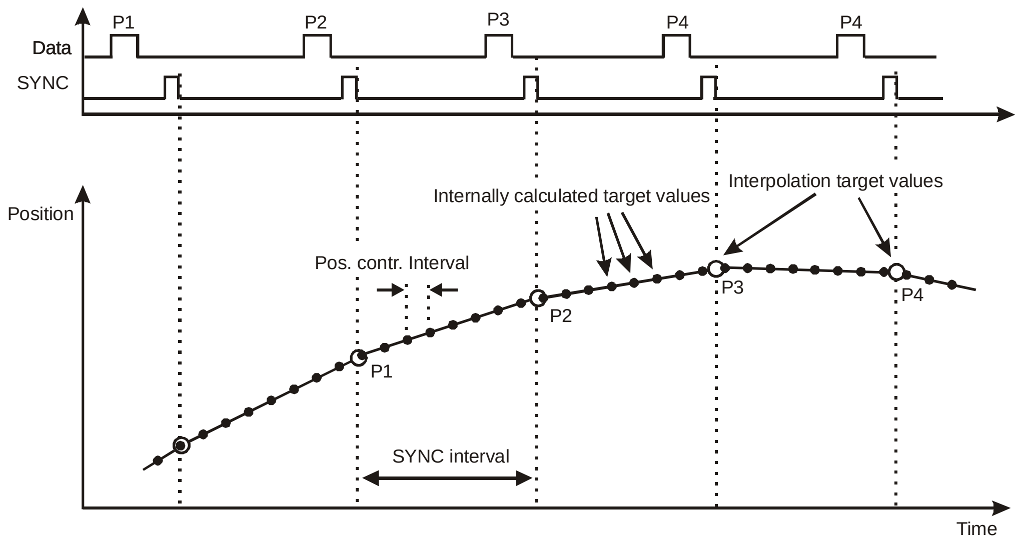
Position controller parameters

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
60FB		Position control parameters	RECORD				Parameters of the position controller.
60FB	00	Number of entries	Unsigned 8	CONST	N	6	Number of entries.
60FB	01	kp	Unsigned 16	RW	N	10	Proportional amplification.
60FB	02	ki	Unsigned 16	RW	N	0	Integral amplification.
60FB	03	kd	Unsigned 16	RW	N	100	Differential amplification.
60FB	04	td	Unsigned 8	RW	N	0	Scan time of the D proportion in units of the controller scan time.
60FB	05	kv	Unsigned 16	RW	N	100	Velocity amplification factor kv.
60FB	06	Ff	Unsigned 8	RW	N	0	Feed forward factor.

The parameterisation of the position controller can be carried out by means of the DcSetup programme. See "Position Controller dialogue box" on page 59.

Interpolated Position Mode – Position control with interpolation

The interpolation principle is easy to apply. At first, the "Interpolation Submode Select" ($60C0_h$) parameter is set to equal 0. Via the parameter "Interpolation Data" ($60C1_h$ – sub-index 1), the superordinate control hands over a new target position value (in user-defined units) to the drive controller in precisely defined time intervals (Sync time). The drive generates target values for its position controller so as to make sure that the given target position value is achieved in linear at the end of the time interval.



In order to launch the interpolation, the Sync signal has to be active and Bit 4 has to be set in the control word.

The length of the SYNC interval in µs has to be entered in the "Communication Cycle Period" (1006_h) object so as to make sure that the drive calculates the interim values correctly.

If the SYNC signal fails or comes in too late (0.5-1 ms), the interpolation process is stopped and the Bit 15 is set in the status word. For this, the last target position is maintained.

Selection of the interpolation method

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
60C0	00	Interpolation sub-mode select	Signed 16	RW	N	0	0: selection of the interpolation process.

Data record for interpolation

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
60C1		Interpolation data record	RECORD				Data record for interpolation.
60C1	00	Number of entries	Unsigned 8	CONST	N	1	Number of entries.
60C1	01	Interpolation data	Signed 32	RW	Y	0	Interpolation data (absolute value target position in user-defined-unit).

Interpolation interval

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
60C2		Interpolation time period	RECORD				Interpolation cycle in µs.
60C2	00	Number of entries	Unsigned 8	CONST	N	2	Number of entries.
60C2	01	ip time units	Unsigned 16	RW	N	3000	Scan time for the interpolation.
60C2	02	ip time index	Interger 8	RW	N	-6	Time unit -6 corresponds to µs (1 µs = 10^{-6} s).

Configuration of the interpolation buffer

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
60C4		Interpolation data configuration	RECORD				Configuration of the interpolation buffer.
60C4	00	Number of entries	Unsigned 8	CONST	N	2	Number of entries.
60C4	01	Max. buffer size	Unsigned 8	RW	N	31	Maximum size of the target value buffer.
60C4	02	Actual size	Unsigned 8	RW	N	15	Adjustable size of the target value buffer.

Homing Mode – Reference run

The objects listed here affect the execution of a drive module's reference run. The reference run is required for the positioning module in order to define the zero point of the axis.

See "Reference Run dialogue box" on page 64.

The type as well as the velocities and acceleration during the search for the zero point can be set.

For the execution of the reference run, the module has to be switched to the Homing Mode of operation. Via the control word, the reference run is launched and, with the help of the status word, the actual reference run status can be queried:

Control word Bit 4	Meaning
0	Reference run not active.
0 -> 1	Launch reference run.
1	Reference run active.
1 -> 0	Interrupt reference run.

Status word Bit 12	Status word Bit 13	Meaning
0	0	Reference run not yet completed.
0	1	Reference run successfully completed.
1	0	Error occurred during reference run.

Reference run method

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
6098	00	Homing method	Signed 8	RW	N	17	Type of reference run.

The following possibilities can be selected:

Reference run method	Direction	Target	Reference point for zero position
1	negative	limit switch	Next index signal after the limit switch during the moving out of the limit switch.
2	positive	limit switch	Next index signal after the limit switch during the moving out of the limit switch.
3	positive	reference switch	Next index signal after the limit switch during the moving out of the limit switch.
6	negative	reference switch	Next index signal after the limit switch during the moving out of the limit switch.
17	negative	limit switch	limit switch
18	positive	limit switch	limit switch
21	negative	reference switch	reference switch
19	positive	reference switch	reference switch
34		no run	current position

Reference run speeds

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
6099		Homing speeds	RECORD				Speeds during the reference run.
6099	00	Number of entries	Unsigned 8	CONST	N	2	Number of entries.
6099	01	Speed during search for switch	Unsigned 32	RW	N	10000	Speed during the movement to the switch in user-defined units / s.
6099	02	Speed during search for zero	Unsigned 32	RW	N	1000	Speed during the movement out of the switch in user-defined units / s.

Reference run acceleration

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
609A	00	Homing acceleration	Unsigned 32	RW	N	1000000	Acceleration during the reference run in user-defined units / s².

Reference offset

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
607C	00	Homing offset	Signed 32	RW	N	100	Reference offset in user-defined units.

The reference offset is the distance from the reference point of the zero position (e.g. switching point to the limit switch) to the zero position.

Moving the axis out of a limit switch

If a limit switch of the axis is triggered, the drive module switches to the fault state and the final stage cannot be released, until the limit switch is active.

The “Moving out of a limit switch” mode of operation has been designed in order to move the axis out of a limit switch (see Mode of operation on page 89). For this, output 2 is set with the activation of this mode of operation; it can be used for the shunting of the limit switches of this axis. Afterwards, the module can be switched to the Operation Enable state; by setting the Bit 4 in the control word, the movement out of the limit switch is launched.

Direction of movement	negative in case of an activated positive limit switch
	positive in case of an negative positive limit switch
Velocity	reference run velocity (out of the switch)
Acceleration	reference run acceleration
Target position	switching point of the limit switch + reference point offset

Control word Bit 4	Meaning
0	No meaning.
1	The movement out of the limit switch is carried out, if a limit switch is activated.

Factor Group – Conversion factors

The objects in the Factor Group are required for the conversion of the position, velocity and acceleration value from user-defined units (e.g. µm) into internal units (increments).

The only object which really is required for the conversion is the conversion factor (6093_h). All length, speed and acceleration data that are given in user-defined units are multiplied by this factor. For this, the units for the velocity and acceleration are “user-defined unit / second” or “user-defined unit / second²").

The conversion factor is calculated from the other objects according to the formula described under “Transmission dialogue box” on page 52.

The parameters dimensions index and unit index do not affect the conversion factors, but only serve for information purposes about the units. The DcSetup setting programme can present the right units e.g. for all length and velocity data.

Units index

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
6089	00	Position notation index	Signed 8	RW	Y	0	Index for positioning unit.

By means of the units index, the unit to be used can be set. The following values can be chosen:

Units index	Meaning
0	Not specified.
-3	Milli (metre)
-6	Micro (metre)
75	Arc seconds
76	Arc minutes
77	Degree

Dimensions index

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
608A	00	Position dimension index	Unsigned 8	RW	Y	1	Index for the dimensions of the position.

The dimensions index indicates which physical dimension is to be used. The following values can be chosen:

Dimensions index	Meaning
0	Not specified (1)
1	Length
12	Angular measurement

(1) Here, “increments” is used as user-defined unit. A superordinate control must then convert the unit of length or angular measurement into increments or vice versa by means of the 0x6093 object. Thus, the control or drive module can correctly analyse the corresponding data.

Incremental encoder resolution

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
608F		Position encoder resolution	RECORD				Resolution of the incremental encoder.
608F	00	Number of entries	Unsigned 8	CONST	N	2	Number of entries.
608F	01	Encoder increments	Unsigned 32	RW	N	4000	Number of encoder impulses (4 times the value of the line number).
608F	02	Motor revolutions	Unsigned 32	RO	N	1	Per number of motor revolutions.

As the encoder resolution, 4 times the value of the line number of the encoder has to be set, since the resolution is increased by the 4-edge evaluation. All internal calculations refer to the quadruple resolution. The number of motor revolutions is firmly set to 1.

Transmission ratio

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
6091		Transmission ratio	RECORD				Transmission ratio.
6091	00	Number of entries	Unsigned 8	CONST	N	2	Number of entries.
6091	01	Motor revolutions	Unsigned 32	RW	N	1	Revolutions at the transmission input.
6091	02	Shaft revolutions	Unsigned 32	RW	N	1	Revolutions at the transmission output.

If a transmission exists between the motor and the drive axis, the transmission ratio factor can be set in this object.

Feed forward constant

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
6092		Feed forward constant	RECORD				Feed forward constant.
6092	00	Number of entries	Unsigned 8	CONST	N	2	Number of Entries.
6092	01	Feed	Unsigned 32	RW	N	10000	Feed.
6092	02	Shaft revolutions	Unsigned 32	RO	N	1	Per revolution.

The forward feed constant defines how many user-defined units (e.g. µm) are covered per drive axis revolution.

Conversion factor

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
6093		Position factor	RECORD				Conversion factor for position units (is also used for velocity and acceleration).
6093	00	Number of entries	Unsigned 8	CONST	N	2	Number of Entries.
6093	01	Numerator	Unsigned 32	RW	N	2	Numerator.
6093	02	Divisor	Unsigned 32	RW	N	5	Divisor.

General parameters

Here, those objects are grouped that cannot be allocated to a certain group, since they affect several modes of operation or functional groups.

Maximum speed

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
607F	00	Max. profile speed	Unsigned 32	RW	N	100000	Maximum speed.

The maximum speed internally restricts all velocities in the modes of operation “Profile velocity mode” and “Profile position mode”. Furthermore, it indicates the range which is available as target value range in the mode of operation “Velocity controller with analogue input” (-10 ... +10 V = -Vmax ,.. +Vmax)

Maximum acceleration

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
60C5	00	Max. acceleration	Unsigned 32	RW	N	10000000	Maximum acceleration.

The maximum acceleration internally restricts all acceleration and braking ramps in the modes of operation “Profile Velocity Mode” and “Profile Position Mode”, even if the acceleration parameter (6083_h) has a greater value.

Axis direction

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
607E	00	Polarity	Unsigned 8	RW	N	0x00	Axis direction (0x00: positive, 0xC0: negative).

The Axis direction parameter makes it possible to invert the rotation direction of the motor at identical target values.

Digital inputs

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
60FD	00	Digital inputs	Unsigned 8	RO	Y	0x00	Current status of the digital inputs.

The first four bits contain the current status of the four digital inputs. The allocation of the inputs to the bits as well as the active range of the inputs is set by means of the 6510_h 01_h - 05_h parameters.

The meaning of the bits is allocated as follows:

Bit number	Meaning
Bit 0	Negative limit switch
Bit 1	Positive limit switch
Bit 2	Reference switch
Bit 3	Enabling signal

Digital outputs

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
60FE	00	Digital outputs	Unsigned 8	RW	Y	0x00	Actual state of the digital outputs.

With the help of the first bit of the object 0x60FE, output 3 can be set or reset for the brake control. Currently, only output 3 is shown on the object “Digital outputs”.

The meaning of the bits is internally allocated as follows:

Bit number	Meaning
Bit 0	Output 3 for the brake control
Bit 1 ... 7	Not yet used

Specific parameters of the final power stage

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
60F7		Power stage parameters	RECORD				Specific parameters of the final power stage.
60F7	00	Number of entries	Unsigned 8	CONST	N	1	Number of Entries.
60F7	01	Limit switch active: disable current	Unsigned 8	RW	N	0	Behaviour of the current in case of an activated limit switch.

Normally, the current is switched off by the hardware, if a limit switch is activated. By means of the 0x2054 object (“Limit switch shunt"), the safety circuit can be shunted in order to be able to switch on the current again. The motor can then be moved in both directions. There are, however, applications in which it is desired that the movement in the active direction of the limit switch is blocked. In this case, the parameter “Limit switch active: disable current” must be set to 1; otherwise movements in both directions are still admissible.

Parameter value	Meaning in case of an activated limit switch
0	The movement in both directions is possible.
1	Only the movement out of the active limit switch is admissible.

Drive data

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
6510		Drive data	RECORD				General parameters of the drive module.
6510	00	Number of entries	Unsigned 8	CONST	N	3	
6510	01	Digital input1 configuration	Unsigned 8	RW	N	0	Configuration of the digital input 1.
6510	02	Digital input2 Configuration	Unsigned 8	RW	N	1	Configuration of the digital input 2.
6510	03	Digital input3 Configuration	Unsigned 8	RW	N	2	Configuration of the digital input 3.
6510	04	Digital input4 Configuration	Unsigned 8	RW	N	3	Configuration of the digital input 4.
6510	05	Invert digital inputs	Unsigned 8	RW	N	0x00	Evaluation of the digital inputs (high- / low-active).
6510	06	IIt – current limitation time	Unsigned 16	RW	N	500	I^2t – current limitation time.
6510	07	Mode of brake control	Unsigned 8	RW	N	0	Mode of operation for the control of the brake.
6510	08	Init. value of digital outputs	Unsigned 8	RW	N	0x00	Initialisation values of the digital outputs.
6510	09	Invert digital outputs	Unsigned 8	RW	N	0x01	Evaluation of the digital outputs (high- / low-active).
6510	0A	Hardware Type	Unsigned 8	RO	N	0x00	Dummy
6510	0B	Standstill Tolerance Window	Unsigned 32	RW	N	10000	Position tolerance of the standstill supervision
6510	0C	Actual Standstll State	Unsigned 8	RO	N	0	Actual Standstill State 0 = Motor in silence, motor does not move 1 = Motor moves
6510	0D	Home Index Offset	Unsigned 32	RW	N	0	Offset at reference run with index signal (see reference run dialogue box)

Via the parameter "Digital InputX Configuration" it is defined on which bit (0 ... 3) in the “Digital Inputs“ parameter, input X is illustrated.

The parameter "Invert digital inputs" defines, if the inputs are to be evaluated low-actively or high-actively.

Invert digital inputs		Meaning
Bit 0 … 3	0	Input 1 ... 4 is high-active (i.e. no signal inversion).
	1	Input 1 ... 4 is low-active (i.e. signal inversion).

The meaning of the bits is internally allocated as follows:

Mode of operation	**Parameter value**
Bit 0	Negative limit switch
Bit 1	Positive limit switch
Bit 2	Reference switch
Bit 3	Enabling signal

The “Mode of brake control” determines how the brake is controlled via output 3. It has to be taken into account that the 0x60FE “Digital outputs” object is currently used only for output 3. This, however, can still be changed in the future.

In the manual mode, the brake has to be set or reset via the “Digital Outputs” parameter by means of SDO. The first bit of the “Init. value of digital outputs” object can be defined as the initial value of output 3.

In the automatic mode, the “Digital Output” parameter is blocked for a change from outside. The “Init. value of digital outputs” parameter does not have a meaning in this mode. The brake control is carried out internally. In the dead status, the brake is automatically set and in the live status, the brake is also automatically reset.

Parameter value	**Mode of the brake control**
0	Manual
1	Automatic

The "Invert digital outputs" parameter defines, if output 3 is to be evaluated low-actively or high-actively. This parameter is blocked for the automatic mode.

Invert Digital Outputs		**Meaning**
Bit 0	0	Output 3 is high-active (i.e. no signal inversion): Value 0 or 1 to Bit 0 from the object 0x60FE → value 0 or 1 at output 3)
	1	Output 3 is low-active (i.e. signal inversion): Value 0 or 1 to Bit 0 from the object 0x60FE → value 1 or 0 at output 3)

The different parameters of the brake control can very easily be defined and tested with the DcSetup programme in the “Brake” dialogue box.

The standstill monitoring was implemented to monitor the movement condition of the motor. If the motor changes from the condition "movement" into the condition "silence", the actual position is recorded. A accordant signal is given out to notify that the motor is in condition "silence". If the motor leaves the tolerance range of the previous recorded position another signal is given out to notify that the motor is in condition "movement". This tolerance range can be defined in the parameter "Standstill Tolerance Window". The standstill monitoring works in all movement operation modes (see frame connector X3). The object Actual Standstill Sate indicates if the motor is in silence or moves. The object Actual Standstill State indicates the condition like it determines the firmware of UVE8112, IMD10, not like it determines the PIC circuit.

Manufacturer-specific parameters

In the range of 2000_h to $5FFF_h$ of the object directory, there are those parameters that are not predefined by a CanOpen specification.

Can Baud Rate

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
2001		Can Baud Rate	RECORD				Displaying and changing of the der CAN-Bus baud rate in the modul
2001	00	Number of Entries	Unsigned 8	CONST	N	3	Number of entries
2001	01	Actual Baud Rate	Unsigned 8	RO	N		Current Baud Rate Code
2001	02	Actual High Speed Submode	Unsigned 8	RO	N		Current Baud Rate Code in High Speed Submode
2001	03	New High Speed Submode	Unsigned 8	RW	N	0	New Baud Rate Code in High Speed Submode

With a double click on the objet 2001-03 New High Speed Submode you can change the baud rate. The accordant DIP switch have to be set (see Chapter DIP switch in Description of the drive module).

The following baud rate code is used in the object Can Baud Rate:

Baud Rate Code	CAN-Bus Baud Rate
0	1000 kBit/s
1	800 kBit/s (not supported)
2	500 kBit/s
3	250 kBit/s
4	125 kBit/s
5	Reserviert (not supported)
6	50 kBit/s
7	20 kBit/s
8	10 kBit/s (not supported)

Relation between Baud rate and wire length of CAN bus:

Baud Rate (kBit/s)	20	50	125	250	500	1000
Wire length (m)	2500	1000	500	250	100	25

Maximum jerk

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
2040	00	Max. jerk	Unsigned 32	RW	N	10000000	Max. jerk in user-defined unit / s^3.

The maximum jerk limits the vibration behaviour of the mechanical axis. The jerk value can be read in and used by the CNC control. Within the final power stage, this parameter is not used. The CNC control of ISEL uses this parameter. The smaller the jerk value, the better the vibration behaviour of the axis. The acceleration or braking process, however, takes longer accordingly.

Axis type

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
2041	00	Axis type	Unsigned 8	RW	N	0	0 Linear axis. 1 Rotary axis.

The CNC control of ISEL uses this parameter for the display. Otherwise, this parameter does not have any particular application purpose.

Password

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
2042		Password	RECORD				Change of password
2042	00	Number of entries	Unsigned 8	CONST	N	3	Number of Entries.
2042	01	Current password	Unsigned 32	WO	N		Current password
2042	02	Modification mode	Unsigned 8	RO	N		0 → Wrong password 1 → Correct password
2042	03	New password	Unsigned 32	WO	N		New password

The “Password” object offers the possibility to protect the set parameters from unauthorised changes. It has to be taken into account that the protection mechanism does not work on the UVE 8112 level. An application software has to use the possibilities offered by this object in order to protect the parameters within the UVE 8112 module.

During the initialisation phase, the “Modification mode” parameter is always set to 0. The writing of the “Current password” parameter with the correct password sets the “Modification mode” parameter to 1. In this case, a new password can be defined via the "New password" parameter. A wrong password sets back the "Modification mode” parameter to 0. The setting of a new password is thus prohibited. Apart from the setting of a new password, an application software can control by means of the “Modification mode” parameter, if the password entered is correct or not, and respond accordingly. After a firmware update via the bootstrap loader, the module always has the default password “ISEL”.

Synchronous control

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
2043		Synchronous control	RECORD				Control of the gantry axis .
2043	00	Number of entries	Unsigned 8	CONST	N	3	Number of entries.
2043	01	Synchronous mode	Unsigned 8	RW	N	0	0 → No gantry axis 1 → Slave axis 2 → Master axis
2043	02	Slave minimal node ID	Unsigned 8	RW	N	127	Minimum CAN ID no. of the slave axis.
2043	03	Slave maximal node ID	Unsigned 8	RW	N	127	Maximum CAN ID no. of the slave axis.
2043	04	Slave error	Unsigned 8	RW	N	0	0 → Fault-free slave 1 → Faulty slave
2043	05	Master status word	Unsigned 16	RW	Y	0x0027	Master axis status word.
2043	06	Master: current position	Signed 32	RW	Y	0	Current position of the master axis.
2043	07	Following error window	Unsigned 32	RW	N	10000	Position window for the monitoring of the gantry axis.
2043	08	Following error	Signed 32	RO	Y	0	Current deviation of position between the master and the slave axis.
2043	09	Help variable	signed 32	RW	N	0	Help variable for the synchronisation process.
2043	10	Max. following error	signed 32	RW	N	0	Max. deviation of position between master and slave.
2043	11	Coupling factor	unsigned 8	RW	N	4	Coupling factor with master axis.
2043	12	Homing Follow Error Check (Dummy)	Unsigned 8	RW	N	1	Homing Follow Error Check (Dummy)

With the help of the synchronous control objects given above, the ISEL CNC control realises the operation of the gantry axis. The “Synchronous Mode” object is set during the CNC operation by the control, if applicable. Currently, up to 2 gantry axes are allowed per machine. The CAN node ID of the slave axes are stored in the “Slave minimal node ID” and “Slave maximal node ID” for the monitoring of errors. The “Slave Error” object is only used by one master axis. If the related slave axis is faulty, this object is set in the master. In a slave axis, this object is of no importance. During the gantry, the slave axis receives the status as well as the current position of the master axis in the objects “Master: status word” und “Master: current position”. By means of these two objects, the slave axis recognises, if the master axis is faulty or not and if the deviation of position between the master and the slave exceeds the limit value in the “Following error window” object or not. The current value of the deviation of position between the master and slave can be seen in the “Following error” object. The “Help variable” sub-index is intended for internal use only. The “Max. following error” sub-index of the slave axis is an auxiliary means for the commissioning. The user can launch the DCSetup in the declaration mode and this variable indicates the maximum deviation of position between the master and the slave. The seven objects mentioned last are used only by one slave axis. In the master operation, they are not taken into account.

With the exception of the two objects “Following error window” and “Coupling factor”, all objects of the synchronous control are commissioned during operation by the ISEL CNC control. In the “Can Interpolation” dialogue box of the slave axis, the position error of a gantry axis can be defined as a value for the “Following error window” object and the coupling factor of the slave as the value of the “Coupling factor" object. It has to be taken into account that these two objects of a slave axis, and not of a master axis, have to be set.

Active target value channel

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
2050	00	Active comm. interface	Unsigned 8	RO	N		Currently set target value channel

This parameter indicates which target value channel is currently activated; said channel is fixed by means of the DIP switch on the front side and queried during boot-up or reset.

Set bit	Designation	Meaning
Bit 0	CAN	Via CAN, data can be read.
Bit 1	RS232	Via RS232, data can be written.
Bit 2	Analogue input (current controller)	Control word and mode of operation cannot be changed. Control of the state machine via release signal.
Bit 3	Analogue input (velocity controller)	See current controller.

Fault reset

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
2051	00	Fault reset	Unsigned 8	WO	N		Fault reset or reset.

The writing of a 1 releases a fault reset (alternatively to control word Bit7 0→1). The writing of a 2 releases a complete reset of the module.

Error byte

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
2052	00	Error byte	Unsigned 8	RO	Y		Display of the current error code.

In this parameter, the internal error code is indicated, when an error occurred (Fault state). If case of several errors, the error code with the highest priority (smallest value) is given.

Limit switch shunt

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
2054	00	End switch bridge	Unsigned 8	RW	N	0	0: No shunting 1: Shunting of the limit switches

In case of an activated limit switch, the main supply voltage is switched off for safety reasons. A shunting of the limit switch deactivates the safety circuit. Thus, the supply voltage can be switched on. Afterwards, the axis can be moved out of the active limit switch. Concerning this subject, see Moving the axis on page 98.

Offset analogue input

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
2064		Calibrate analogue IO	RECORD				Offset adjustment of analogue inputs.
2064	00	Number of entries	Unsigned 8	CONST	N	3	Number of Entries.
2064	01	Calibrate analogue input	Unsigned 8	WO	N	0	A value unequal to zero launches the adjustment procedure.
2064	02	Analogue input offset	Signed 16	RW	N	0	Offset of the reference input (* 32)
2064	03	Internal current offset	Signed 16	RW	N	0	Offset or the current measurement input (* 32)

This object can be used to realise the offset adjustment of the internal A/D converter. The writing of a 1 on the sub-index 1 starts the automatic adjustment. For this, the module has to be in the "Switched On" (or "Operation Enable") state.

Under the sub-indices 2 and 3, the current offset values can be queried. These are given in units of 32 times the internal A/D converter resolution (11 bit, -1023 ... +1023). It is also possible to directly write the offset values.

Actual motor current

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
2070	00	Actual motor current	Signed 16	RO	Y	0	Actual motor current (internal unit).

Actual analogue target value

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
2071	00	Actual analogue input	Signed 16	RO	Y	0	Actual analogue target value -1023 ... +1023.

The following objects are required in order to configure and to launch test signals for the commissioning process.

Current control test signal

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
2090		Torque control test: input	RECORD				Test signal for current controllers.
2090	00	Number of entries	Unsigned 8	CONST	N	3	Number of entries.
2090	01	Test: input duration	Unsigned 8	RW	N	25	Duration of the test signal.
2090	02	Test: input value	Unsigned 8	RW	N	100	Amplitude of the test signal.
2090	03	Test: input status	Unsigned 8	RO	Y	0x00	Status of the execution (Bit 7: test signal active).

Velocity control test signal

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
2091		Velocity / Position control: test input	RECORD				Test signal for velocity and position controller.
2091	00	Number of entries	Unsigned 8	CONST	N	4	Number of entries.
2091	01	Test: input duration	Unsigned 16	RW	N	500	Duration of the test signal.
2091	02	Test: input velocity	Signed 32	RW	N	10000	Amplitude of the test signal (target velocity [Incr./s])(1).
2091	03	Test: input acceleration	Unsigned 32	RW	N	10000000	Ramp slope of the test signal (Incr./s²).
2091	04	Max Move Length	Signed 32	RW	N	50000	Maximal Move Length (inc)

(1) Attention: If the “Configure test input” object (index 20A0 and sub-index 01) amounts to the value of 6, the maximum speed is determined. In this case, the amplitude of the test signal is the maximum path length to be moved in [Incr.]. This object restricts the range of movement of the axis during the determination process.

Start test signal

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
20A0		Test input	RECORD				Choose test signal and start.
20A0	00	Number of entries	Unsigned 8	CONST	N	2	Number of entries.
20A0	01	Configure test input	Unsigned 8	RW	N	0	Selection of the test signal.
20A0	02	Start test input	Unsigned 8	WO	N	0	The writing of a 1 launches the test signal.

The sub-index 2 helps to carry out the following settings:

Configure test input	Internal mode of operation
0	Back to normal mode of operation
1	Test signal current control
2	Test signal velocity control
3	Test signal position control
4	Encoder test positive direction
5	Encoder test negative direction
6	Determine maximum velocity
7	Velocity control with ramp profile
8	Following error
9	Speed amplification factor kv

Trace Data

Index	Sub	Name	Type	Attrib	Map	Default value	Meaning
2081	00	Trace data	Domain	RO	N		Data field for internal registrations of courses of current and velocities etc.

EDS / DCF files

The drive module includes a so-called EDS file (Electronic Data Sheet) which contains a description of all objects of a module. This file has a form which is defined in CanOpen so as to make sure that it can be read in by different configuration tools. The DcSetup programme also uses this EDS file in order to create a list of existing objects.

In order to store the current settings of the objects, the DCF files (Device Configuration File) are created. They also contain the object descriptions from the EDS file, but also offer an additional entry, i.e. the current value of the object.

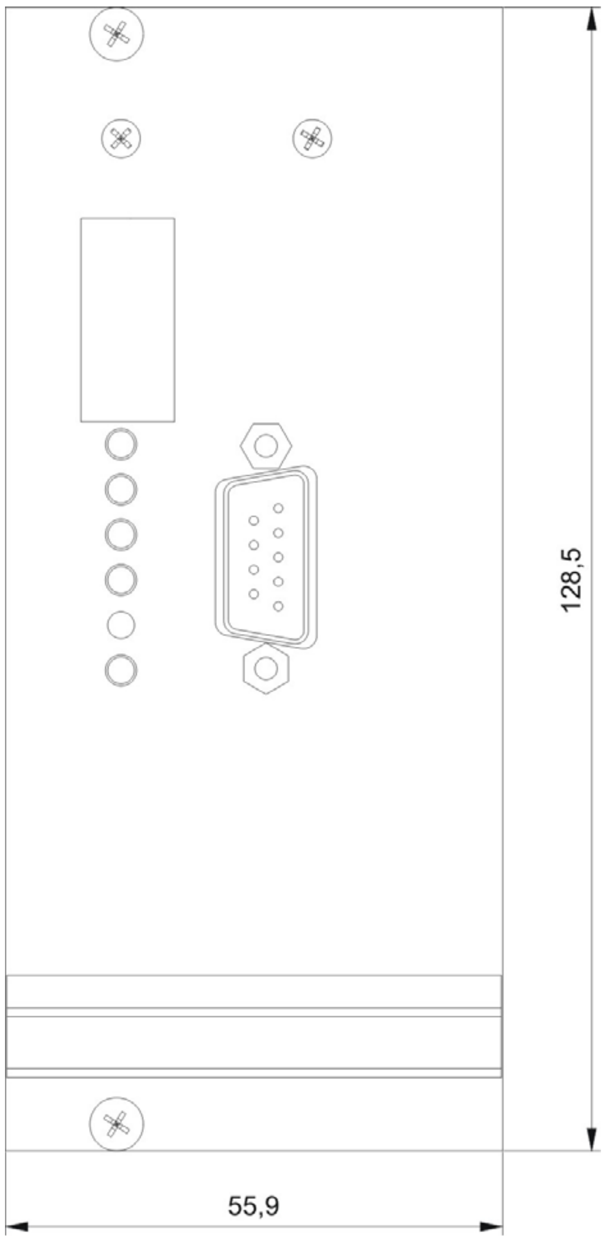
At the beginning of the file, general data on the device and the CanOpen characteristics are given. Then, the object descriptions are given which have the following structure:

e.g. "Reference run velocity" object (6099_h sub-index 1)

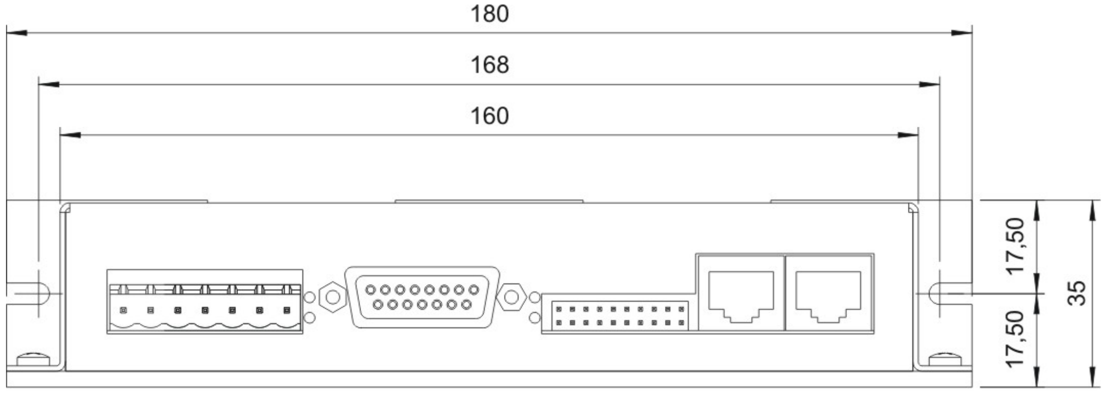
File entry	**Meaning**
[6099sub1]	Index and sub-index
ParameterName=Speed During Search For Switch	Name of the parameter
ObjectType=0x7	Object type (0x7 = variable)
DataType=0x0007	Data type (0x0007 = UNSIGNED32)
AccessType=RW	Access type (RW = read and write admissible)
DefaultValue=10000	Default value
PDOMapping=0	Mapping admissible? (0 = no Mapping)
ParameterValue=15000	Current value, **given only in the DCF file**

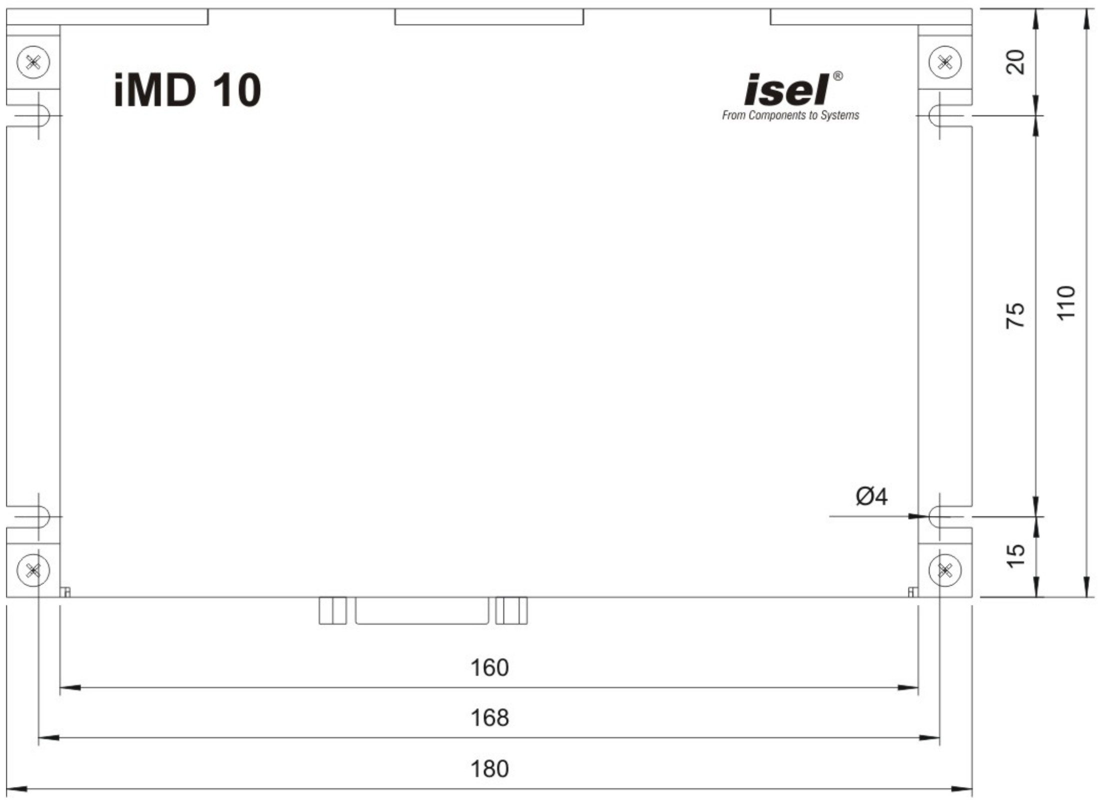
Appendix

UVE8112 package dimensions



IMD10 package dimensions





Glossary

DCF file

Device Configuration File

Descriptions of the module’s objects and current value of the objects.

EDS file

Electronic Data Sheet File

Descriptions of the module’s objects.

PDO

Process Data Object

Can message with a defined identifier which contains one or more objects without address information.

SDO

Service Data Object

Communication channel between two CanOpen participants.

Index