

Software manual

SDP

SCHUNK Drive Protocol V3.03

Imprint

Copyright:

This manual is protected by copyright. The author is SCHUNK GmbH & Co. KG. All rights reserved. Any reproduction, processing, distribution (making available to third parties), translation or other usage - even excerpts - of the manual is especially prohibited and requires our written approval.

Technical changes:

We reserve the right to make alterations for the purpose of technical improvement.

Document number: 1006503

Version: 03.00 | 10/08/2020 | en

© SCHUNK GmbH & Co. KG

All rights reserved.

Dear Customer,

thank you for trusting our products and our family-owned company, the leading technology supplier of robots and production machines.

Our team is always available to answer any questions on this product and other solutions. Ask us questions and challenge us. We will find a solution!

Best regards,

Your SCHUNK team

SCHUNK GmbH & Co. KG
Clamping and gripping technology
Bahnhofstr. 106 - 134
D-74348 Lauffen/Neckar
Tel. +49-7133-103-0
Fax +49-7133-103-2399
info@de.schunk.com
schunk.com

Table of Contents

1	General	6
1.1	Presentation of Warning Labels	6
1.2	Applicable documents	7
1.3	User administration	7
1.3.1	USER	7
1.3.2	PROFI	7
1.3.3	Advanced	7
1.3.4	ROOT	7
1.3.5	SCHUNK	7
1.3.6	DISABLED	8
1.4	Unit system	8
1.5	Booting	8
2	Software function	9
2.1	Pseudo-absolute encoder	9
2.1.1	Precondition	9
2.1.2	Function	9
2.2	Standstill commutation	10
2.2.1	Precondition	10
2.2.2	Function	11
3	Communication	12
3.1	Cyclical data exchange	12
3.1.1	Incoming process data	12
3.1.2	Outgoing process data	15
3.2	Status bits	18
3.2.1	Module ready for operation	18
3.2.2	The module moves	18
3.2.3	Target position reached	18
3.2.4	Motion blocked	18
3.2.5	Brake has been engaged	18
3.2.6	Module referenced	19
3.3	Acyclical data exchange	20
3.3.1	Acyclically read the data	20
3.3.2	Acyclically write the data	21
3.4	Diagnostic alarm	21
3.5	Configuration via HW configuration	22
4	Configuration parameters	23
4.1	Data types	23
4.2	Value range	24
4.3	Parameter code display	24
4.4	Parameter	25

4.4.1	Device	25
4.4.2	Motor.....	31
4.4.3	Controller.....	38
4.4.4	Referencing.....	43
4.4.5	Positioning	50
4.4.6	Gear	57
4.4.7	Brake.....	58
4.4.8	Voltage.....	60
4.4.9	Communication	62
4.4.10	General	64
4.4.11	Info	65
4.4.12	Ethernet.....	71
4.4.13	Asynchronous	73
5	Info and error messages.....	85
5.1	Detailed error information	85
5.2	Info codes	85
5.2.1	INFO NO RIGHTS	85
5.2.2	INFO UNKNOWN COMMAND.....	85
5.2.3	INFO FAILED.....	85
5.2.4	NOT REFERENCED	85
5.2.5	INFO COMMUNICATION ERROR.....	85
5.2.6	INFO TIMEOUT.....	86
5.2.7	INFO WRONG DATA TYPE	86
5.2.8	INFO RESTART.....	86
5.2.9	INFO CHECKSUM.....	86
5.2.10	INFO VALUE LIMIT MAX.....	86
5.2.11	INFO VALUE LIMIT MIN.....	86
5.2.12	INFO MESSAGE LENGTH	86
5.2.13	INFO WRONG PARAMETER.....	86
5.2.14	INFO UNKNOWN PARAMETER.....	87
5.3	Error codes	87
5.3.1	ERROR FILE NOT FOUND.....	87
5.3.2	ERROR FILE IS CORRUPT	87
5.3.3	ERROR FILE TYPE WRONG.....	87
5.3.4	ERROR FILE SYSTEM WRONG.....	87
5.3.5	ERROR FILE READ	87
5.3.6	ERROR FILE IS NOT CREATED	87
5.3.7	ERROR FILE WRITE	87
5.3.8	ERROR REBOOT.....	88
5.3.9	ERROR MOTOR PHASE.....	88
5.3.10	ERROR WRONG RAMP TYPE	88
5.3.11	ERROR WRONG DIRECTION	88
5.3.12	ERROR CONFIG MEMORY	88

5.3.13	ERROR SOFT LOW	88
5.3.14	ERROR SOFT HIGH	88
5.3.15	ERROR SERVICE	89
5.3.16	ERROR FAST STOP	89
5.3.17	ERROR TOW	89
5.3.18	ERROR VPC3	89
5.3.19	ERROR FRAGMENTATION	89
5.3.20	ERROR COMMUTATION	89
5.3.21	ERROR I2T	90
5.3.22	ERROR CURRENT	90
5.3.23	ERROR TOO FAST	90
5.3.24	ERROR POS SYSTEM	90
5.3.25	ERROR RESOLVER CHECK FAILED	90
5.3.26	ERROR MATH	90
5.3.27	ERROR CALIB CURRENT	91
5.3.28	ERROR INITIALIZE	91
5.3.29	ERROR INTERNAL	91
5.3.30	ERROR CONNECTION TEMP LOW	91
5.3.31	ERROR CONNECTION TEMP HIGH	91
5.3.32	ERROR MOTOR TEMP LOW	92
5.3.33	ERROR MOTOR TEMP HIGH	92
5.3.34	ERROR TEMP LOW OPTION	92
5.3.35	ERROR TEMP HIGH OPTION	92
5.3.36	ERROR TEMP LOW	92
5.3.37	ERROR TEMP HIGH	92
5.3.38	ERROR LOGIC LOW	93
5.3.39	ERROR LOGIC HIGH	93
5.3.40	ERROR MOTOR VOLTAGE LOW	93
5.3.41	ERROR MOTOR VOLTAGE HIGH	93
5.3.42	ERROR CABLE BREAK	94
5.3.43	ERROR LIFE SIGN	94
5.3.44	ERROR CUSTOM DEFINED	94
5.3.45	ERROR OVERSHOOT	94
5.3.46	ERROR HARDWARE VERSION	94
5.3.47	ERROR SOFTWARE VERSION	94
6	Appendix	95
6.1	Sample travel commands	95
6.1.1	Acknowledge error	95
6.1.2	Referencing	97
6.1.3	Prepositioning	99
6.1.4	Grip	101
6.1.5	Gripping force	103
6.1.6	Release	105

1 General

1.1 Presentation of Warning Labels

To make risks clear, the following signal words and symbols are used for safety notes.



DANGER

Danger for persons!

Non-observance will inevitably cause irreversible injury or death.



WARNING

Dangers for persons!

Non-observance can lead to irreversible injury and even death.



CAUTION

Dangers for persons!

Non-observance can cause minor injuries.

CAUTION

Material damage!

Information about avoiding material damage.

1.2 Applicable documents

- General terms of business *
- "SCHUNK Motion Tool (MTS)" software manual *
- Documentation for the products used *

The documents marked with an asterisk (*) can be downloaded on our homepage **schunk.com**

1.3 User administration

This module is equipped with user management. User rights can be changed via the "User" parameter, [User](#) [► 84].

NOTE

After restarting the module, the user remains active as "USER".

1.3.1 USER

The standard user, active after the module has been switched on. The module can only be parameterized to a limited extent, but has full operational features.

1.3.2 PROFI

Many important parameters can be changed.

The standard password for the professional rights is "Schunk".

NOTE

Incorrect parameterization can result in an unanticipated module behavior. However, the module cannot be destroyed.

1.3.3 Advanced

The most important parameters can be changed.

NOTE

Incorrect parameterization can result in destruction of the module.

1.3.4 ROOT

Full access to the module, all parameters can be changed.

NOTE

Incorrect parameterization can result in destruction of the module.

1.3.5 SCHUNK

Parameters can only be changed by SCHUNK.

1.3.6 DISABLED

Parameters with cannot be changed.

1.4 Unit system

For linear and gripper products, the following unit system applies:

- Position [mm]
- Speed [mm/s]
- Acceleration [mm/s²]
- Jerk [mm/s³]
- Current values [A]
- Times [s]

For rotary units, the following unit system applies:

- Position [degree]
- Speed [degree/s]
- Acceleration [degree/s²]
- Jerk [degree/s³]
- Current values [A]
- Times [s]

1.5 Booting

Default values for motions are predefined for the module as standard values. This allows the module to be commissioned directly without having to set the parameters beforehand. The following default values apply after the restart:

- "Target speed"
as [%] of the maximum value -> 10%, [Max. speed](#) [► 33]
- "Target acceleration"
as [%] of the maximum value -> 10%, [Max. acceleration](#) [► 34]
- "Target jerk"
as [%] of the maximum value -> 50%, [Max. jerk](#) [► 34]
- "Target current"
Rated current, [Nom. Current](#) [► 33]
- User is set to "User" [User administration](#) [► 7]

2 Software function

2.1 Pseudo-absolute encoder

2.1.1 Precondition

If the following preconditions are met, then the modules support the "pseudo absolute encoder" function:

- Brake available
- Position measuring system resolver [Positioning type](#) [► 50]
- OR: position measuring system encoder with index track and
 - DC motor type, [Motor Type](#) [► 32]
 - OR: BLDC motor type
 - OR: PMSM motor type

2.1.2 Function

When the brake is applied, the current position is saved in a non-volatile memory. If the logic voltage is switched off, an attempt is made to save the current position with the remaining residual energy.

Resolver

When the module is switched on again, the position saved beforehand will then be compared with a control value. Should this check be successful, then the position saved will be compared with the current position of the resolver. If these positions are also equal, then the module does not need to be re-referenced.

If the resolver is turned by precisely one revolution when deenergized, the displayed position will be faulty when reactivation takes place.

Encoder with index track

When the module is switched on again, the position saved beforehand will be compared with a control value. Should this check be successful, then the saved position will become the current position. The interval between the next index pulse is calculated at the same time.

In the first motion command that follows, the calculated interval is compared with the measured interval when the index pulse is reached. The module is deemed as referenced if the two values agree with each other. In addition, the index pulse must be reached within a certain period of time after the first motion command has been sent.

If an error occurs during the movement to the index pulse, then the referencing will be deleted.

After a successful reference movement, the index pulse has to be run over at least once in order to activate the function.

If the encoder is moved in a deenergized state, it is possible that the module will carry out a motion of no more than one motor revolution to the next index pulse after it has been switched on again with the incorrect position.

If the encoder is turned by precisely one revolution when deenergized, the displayed position could be faulty when reactivation takes place.

2.2 Standstill commutation

2.2.1 Precondition

If the following preconditions are met, then the modules support the "standstill commutation" function:

- DC motor type, [Motor Type](#) [► 32]
- OR: BLDC motor type
- OR: PMSM motor type
 - Position measurement system "Encoder with index track" and available hall sensors [Positioning type](#) [► 50]
 - OR: position measuring system resolver

NOTE

The direction of motion for block commutation and sine commutation must agree. If the directions of rotation are different, then the phases must be changed and the commutation table [Commutation table](#) [► 36] adjusted.

2.2.2 Function

If all prerequisites are met, then the module will attempt to carry out the standstill commutation. In modules with absolute-value measuring systems, the sine commutation can be activated directly after switch-on, since the position of the "sine pointer" is known.

In modules with an encoder measuring system, the position of the "sine pointer" is not known until the index pulse has been reached. Therefore, the module is moved with block commutation up to the first index pulse and then converted to sine commutation.

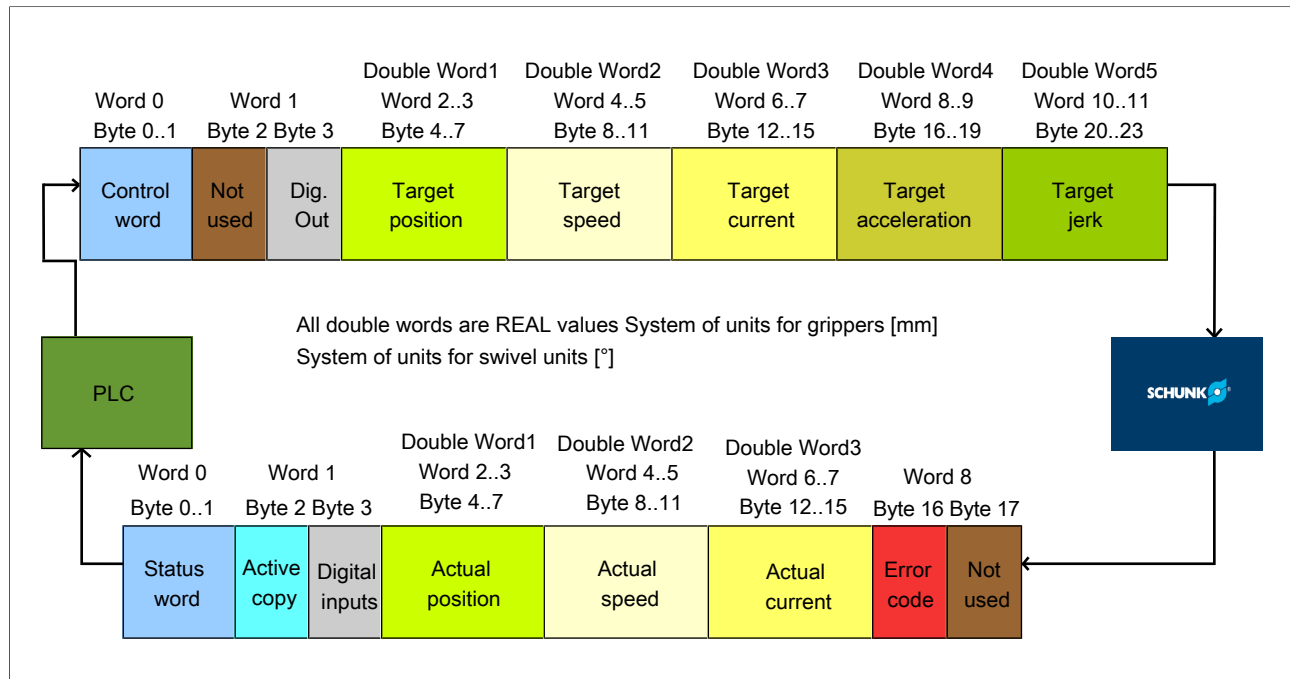
The position of the "sine pointer" to the index pulse is set or readjusted using the "Positioning offset" parameter, . If this value is set to "0", then a search and save process for a sine pointer occurs in the next motion command when energizing the motor phases, . The referencing is deleted in the process, [Pseudo-absolute encoder](#) [► 9].

NOTE

The module should be able to move freely in all directions for the pointer search. The module is moved in a jerking fashion up to two motor revolutions. Communication with the module is not possible during this time.

3 Communication

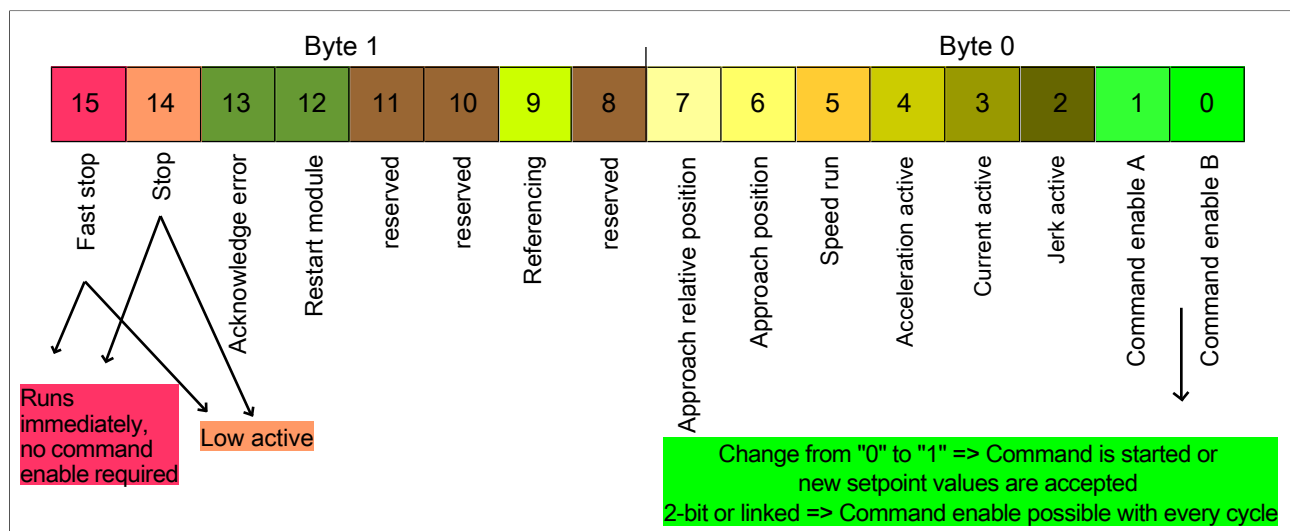
3.1 Cyclical data exchange



Cyclical data exchange

3.1.1 Incoming process data

Control word (bytes 0-1)



Performance priority increases with increasing bit value – "fast stop" has the highest priority.

Bit	Designation	Description
0	Command enable B	Enables the selected command for execution. The most significant active bit is interpreted as a command request. PROFINET: Regardless of the wiring of the other bits, further setpoint values are accepted. PROFIBUS: Additional less significant bits accept the corresponding setpoint values. Default values are used for bits that are not set.
1	Command enable A	
2 ²	Jerk active	PROFINET: Do not use PROFIBUS: When the bit is set, the "Jerk" setpoint value is accepted as the upper limit.
3 ²	Current active	PROFINET: Do not use PROFIBUS: When the bit is set, the "Current" setpoint value is accepted as the upper limit.
4 ²	Acceleration active	PROFINET: Do not use PROFIBUS: When the bit is set, the "Acceleration" setpoint value is accepted as the upper limit.
5 ²	Speed run	The run is only started after the command is enabled. The "Speed" setpoint value is accepted as the target speed and must be within the limits of "0" and "max. speed". The sign determines the direction of travel. The "Current", "Acceleration" and "Jerk" target values are used as maximum values for the run. The motor current is increased up to the setpoint value if the movement is blocked. The "Motion blocked" bit is set in the status word. The run continues once the block is removed. In this case, the "Motion blocked" bit is reset. When the end stop is reached, the motor current is increased up to the setpoint value. The "Motion blocked" bit is set in the status word.
6 ²	Approach position	The run is only started after the command is enabled. The "Position" setpoint value is accepted as the absolute target position and must be within the limits of "min. position" and "max. position". The "Speed", "Current", "Acceleration" and "Jerk" target values are used as maximum values for the run. If successful, the "Target position reached" bit in the status word is set.
7 ²	Approach relative position	The run is only started after the command is enabled. The "Position" setpoint value is accepted as the travel distance. The signs "+" and "-" determine the direction of travel. The "Speed", "Current", "Acceleration" and "Jerk" target values are used as maximum values for the run. If successful, the "Target position reached" bit in the status word is set.
8	reserved	

Bit	Designation	Description
9	Referencing ¹	The referencing run is started immediately, bit 9 of the control word must remain set to "1" until the referencing run has completed. During the referencing run, the zero position of the gripper jaws is determined again according to the configuration, Referencing [► 43]. If the parameter "Approach zero after referencing" is activated, the referencing run is only considered complete once the zero position is reached. If successfully completed, the "Module referenced" bit in the status word is set.
10	reserved	
11	reserved	
12 ²	Restart the module ³	The module will restart.
13 ²	Acknowledge error	The "Error" module status is reset. The cause of the error must be eliminated in advance.
14	Stop	The bit is low active. It runs immediately; no command enable (Release a / B) is required. The motor is driven to speed 0 in a controlled manner and regulated.
15	Fast stop ⁴	The bit is low active. It runs immediately; no command enable (Release a / B) is required. The "Module ready for operation" status bit is reset and the message D9 requiring acknowledgment is triggered. The motor voltage is set to zero, the motor phases are short-circuited.
¹ Referencing is only considered complete once the "zero position" has been reached, provided that this has been parameterized, Approach 0 after referencing [► 48].		
² Is only accepted after the command enable.		
³ Stop and fast stop do not have to be set for a restart.		
⁴ Fast stop does not generate an error message; it only resets the "Module ready for operation" bit.		

NOTE

Can be used for a gripping command:

- Speed run: Brake will never engage.
Check "Gripping successful": "Module blocked" status word bit and end stop not reached.
 - Position run: Target position must be within the workpiece.
Brake does not engage when the workpiece is gripped. Brake engages when no workpiece is present.
Check "Gripping successful": "Module blocked" status word bit set.
- ⇒ Brake is normally only used during positioning (pre-positioning / releasing workpiece / approaching intermediate position).

Digital output (byte 3)

See the option card and the Assembly and Operating Manual for the product.

Setpoint values (bytes 4-23)

Setpoint values for "Position", "Speed", "Current", "Acceleration" and "Jerk" are transferred as REAL values.

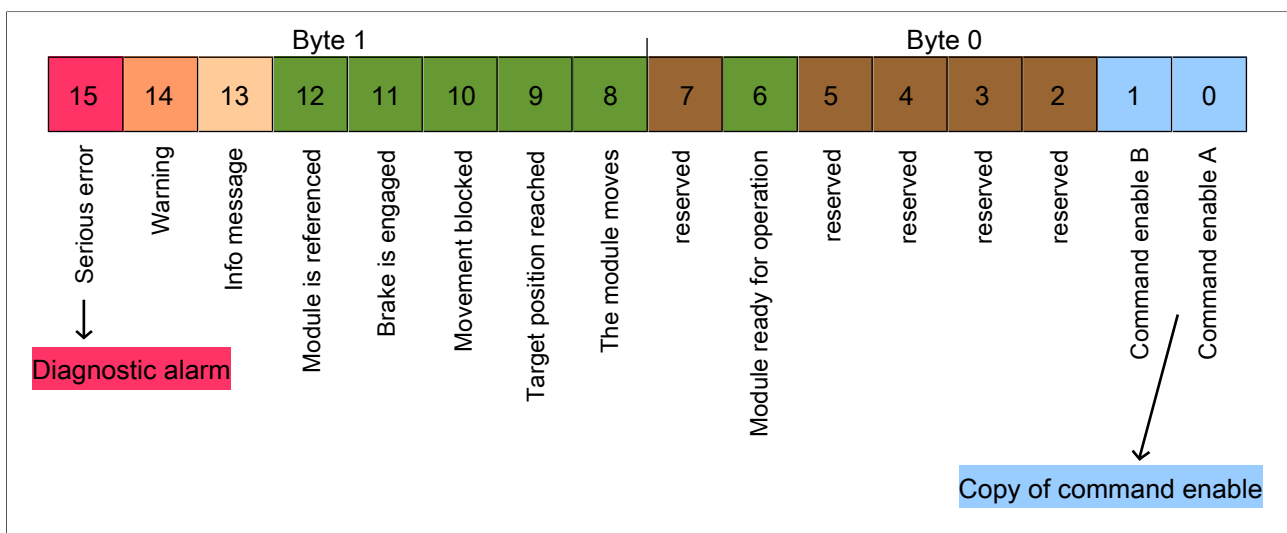
By setting the command enable, modified parameters are accepted if the control bit is active. If the control bit for the corresponding parameter is not active, the stored default value will be accepted.

If the minimum or maximum permissible value is not met when entering a parameter, the minimum or maximum value is set.

Exemplary operation commands are listed in the appendix, [Sample travel commands](#) [► 95].

3.1.2 Outgoing process data

Status word (bytes 0-1)

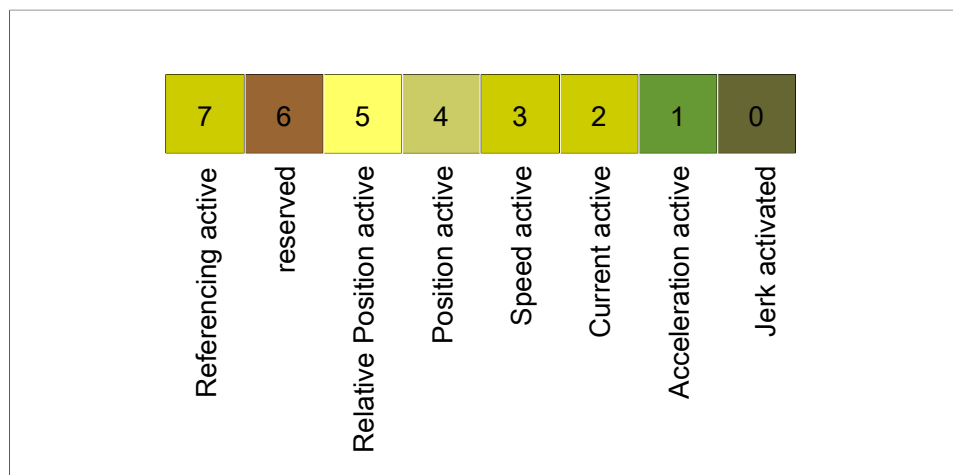


Bit	Designation	Description
0, 1	Command enable A and B	Copy of the command enable from the control word
2	reserved	
3	reserved	
4	reserved	
5	reserved	

Bit	Designation	Description
6	Module ready for operation ³	When the module is completely ready for operation, the bit is set. The prerequisite for this is that all voltages are present and no message requiring acknowledgment is present. If the module is not ready for operation and no error is reported, the current message code can be read from the status word.
7	reserved	
8	The module moves	If the movement threshold is exceeded, the bit is set, Motion threshold [► 54].
9	Target position reached ¹	The target position from the "Approach relative position" command or the "Approach position" command has been reached, Position waiting time reached [► 54]. and the module stops, Motion threshold [► 54].
10	Movement blocked ²	Further movement of the module was blocked, Motion waiting time blocked [► 56].
11	Brake engaged	The product brake is engaged, Brake timeout [► 60].
12	Module is referenced	The last referencing run was successfully completed.
13	Info message ³	An info message is present in the module. Byte 16 of the outgoing process data "Error code" contains the code of the current message. If an error message or warning is present simultaneously, only the code for the error message or warning is displayed.
14	Warning ³	There is a warning in the module. Byte 16 of the outgoing process data "Error code" contains the code of the current warning. If an error message is present simultaneously, only the code for the error message is displayed.
15	Serious error ³	There is a serious error in the product. Byte 16 of the outgoing process data "Error code" contains the code of the current message. Error details may be stored into a file via a diagnostic interrupt.
¹ Setting of the "Target position reached" flag can be delayed with the "Waiting time position reached" parameter, Position waiting time reached [► 54].		

Bit	Designation	Description
² With the "Movement blocked" bit, the grip of a workpiece can be determined. Further plausibility tests, e.g. whether the jaw position corresponds to the workpiece dimensions, help to ensure that the correct workpiece has been gripped. When the motor blocks, the bit is set. The following preconditions must be met in order for the motor to be blocked: <ul style="list-style-type: none"> • The motor turns at a speed below the movement threshold, Motion threshold [► 54]. • The target current is reached (+/-15%). • The time specified under the "Waiting time movement blocked" parameter has expired Motion waiting time blocked [► 56]. If a block is detected, the set current will be reduced by root 2 in sine commutation.		
³ The error code of the last generated error is output in the "Error code" byte 16, Info and error messages [► 85].		

Active copy (byte 2)



The byte copies the values of the corresponding bits from the control word.

Digital inputs (byte 3)

See the option card and the Assembly and Operating Manual for the product.

Actual values (bytes 4-15)

Existing actual values for position, speed and current are returned as REAL values.

Error code (byte 16)

Error code of the present errors, [Info and error messages](#) [► 85].
Exemplary operation commands are listed in the appendix, [Sample travel commands](#) [► 95].

3.2 Status bits

3.2.1 Module ready for operation

When there is no error, the module is ready for operation, [Motion threshold](#) [► 54].

3.2.2 The module moves

When the module moves, the bit is set.

3.2.3 Target position reached

When the position movement has reached the target position and the module is stationary, the bit is set.

3.2.4 Motion blocked

NOTE

It is not possible to achieve reliable detection of whether an object has been gripped when a "Movement blocked" bit is set. For gripping, it is recommended to check plausibility with further data. For example, the actual position can be compared with the workpiece diameter and it can be determined whether the two values are the same.

When the motor blocks, the bit is set.

The following preconditions must be met in order for the motor to be considered blocked:

- The motor turns at a speed below the movement threshold, [Motion threshold](#) [► 54].
- The target current is reached (+/-15%).
- The time parameterized under the parameter "Waiting time movement blocked" has expired, [Motion waiting time blocked](#) [► 56].

NOTE

When the blocking is detected, the set current will be reduced by root 2 in sine commutation.

3.2.5 Brake has been engaged

When the brake has been engaged, the bit is set.

3.2.6 Module referenced

During a reference movement, parameter values are taken over from the configuration of the module. These parameter values can be customized with the user "Profi":

- max. reference current (0x7D45) [Referencing max. current](#) [► 48]
- speed (0x7D46), [Speed referencing](#) [► 49]
- acceleration (0x7D47), [Acceleration referencing](#) [► 49]

Optionally, a reference movement can be completed with the movement to the zero point: if the configuration parameter "Move to referencing 0" (0x7D44) is activated, the module moves to the "0" position after the reference movement. The same process values apply as with the reference movement.

After successful completion of the referencing process, the "Module is referenced" status bit is set.

Under certain conditions, a successfully performed referencing process is retained even after the module has been switched off [Pseudo-absolute encoder](#) [► 9].

NOTE

Prior to a reference movement, all workpieces must be removed for a gripper.

3.3 Acyclical data exchange

The acyclical data exchange is carried out with

- PROFIBUS via slot 1,
- PROFINET via the slot and sub-slot. The numbers of these slots are variable and depend on the bus configuration.

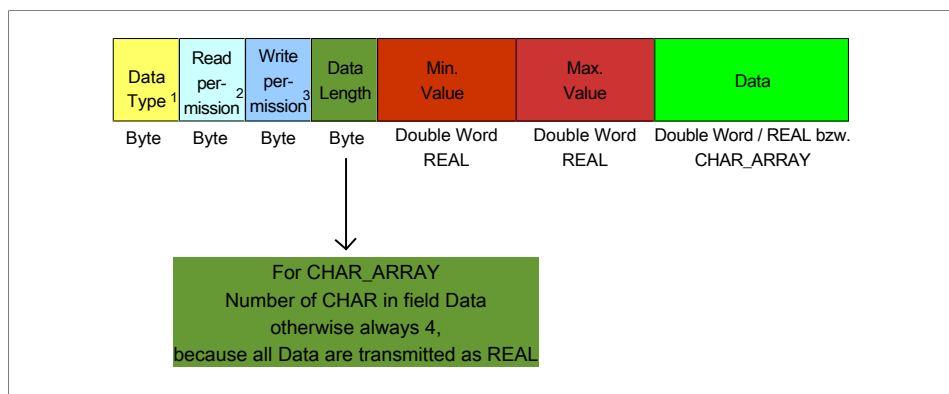
The index of configuration parameters, consisting only of the low byte of the parameter numbers, [Configuration parameters](#) [► 23].

This means that the value 0x7D00 can always be deducted from the specified parameter number.

Example:

Desired parameter	Max. speed
Parameter number	0x7D24
PROFIBUS slot	0x01
PROFIBUS / PROFINET index	0x24

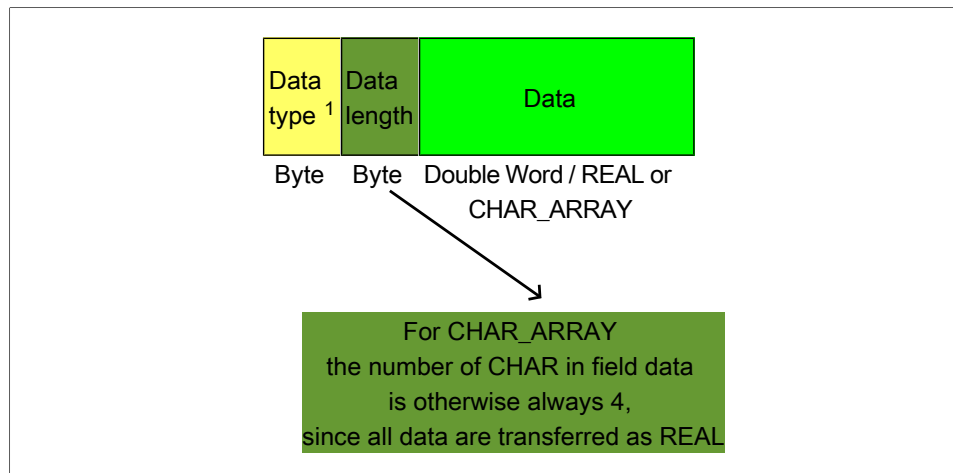
3.3.1 Acyclically read the data



Read user data when acyclical

- 1) Data types, [Data types](#) [► 23]
- 2) and 3) read and write permissions, [User](#) [► 84]

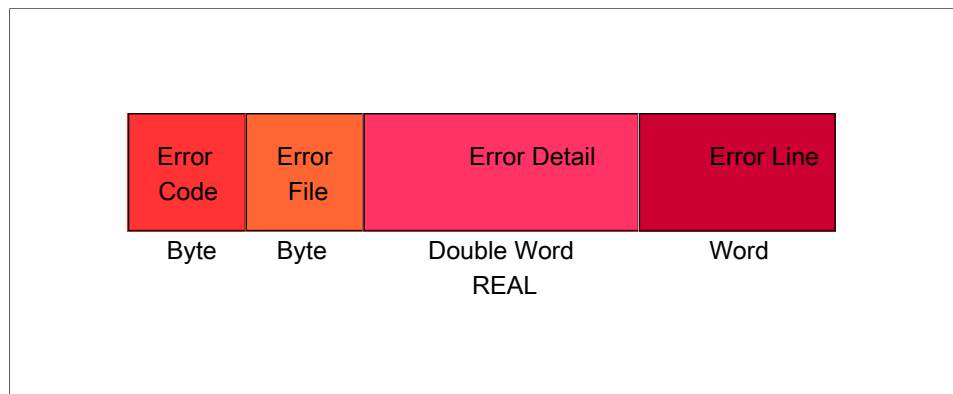
3.3.2 Acyclically write the data



Write user data when acyclical

1) Data types, [Data types](#) [► 23]

3.4 Diagnostic alarm



User data diagnostics alarm

3.5 Configuration via HW configuration

Parts of the module configuration can be made directly via the HW configuration using the PROFIBUS or PROFINET.

The configuration of the HW configuration contains parameters that can be written acyclically.

These parameters can be enabled via the "HW config enable" parameter. As standard, no check mark is placed.

- No check mark is placed for "HW config enable"
 - The parameter settings for the remaining parameters are ignored and are not transferred to the device by the control unit.
- Check mark is placed for "HW config enable"
 - For each new registration of the module on the PROFIBUS or PROFINET, the parameters affected by the configuration will be overwritten directly. Changes made or data that has been written acyclically will be overwritten.

The following HW config parameters can also be written and controlled acyclically:

- endless
 - [Endless](#) [► 27]
- change direction
 - [Inverted direction of rotation of motor](#) [► 26] and [Inverted position measurement](#) [► 26]
- energy save enable
 - [Saving energy](#) [► 31]
- wait pos reached
 - [Position waiting time reached](#) [► 54]
- reference type
 - [Referencing type](#) [► 43]
- move zero after reference
 - [Approach 0 after referencing](#) [► 48]
- use index
 - [Usage index](#) [► 46]

4 Configuration parameters

4.1 Data types

The following data types are used:

Name	Value	Description
UINT8	1	Unsigned integer with 8 bits (1 byte)
INT8	2	Signed integer with 8 bits (1 byte)
UINT16	3	Unsigned integer with 16 bits (2 bytes)
INT16	4	Signed integer with 16 bits (2 bytes)
UINT32	5	Unsigned integer with 32 bits (4 bytes)
INT32	6	Signed integer with 32 bits (4 bytes)
UINT64	7	Unsigned integer with 64 bits (8 bytes)
INT64	8	Signed integer with 64 bits (8 bytes)
FLOAT	9	Floating point number with single precision (4 bytes)
DOUBLE	10	Floating point number with double precision (8 bytes)
(CHAR_ARRAY) -	11	String
BOOL	12	Boolean value (1 byte)
BINARY	13	Byte sequence
ENUM	14	Enumeration (2 bytes)

4.2 Value range

The following value ranges are used:

- MAX_BOOL = 1
- MAX_INT8 = 127
- MAX_INT16 = 32767
- MAX_INT32 = 2147483647
- MAX_UINT8 = 255
- MAX_UINT16 = 65535
- MAX_UINT32 = 4294967295
- MAX_CHAR = 255
- MAX_ENUM = 65535
- MAX_FLOAT = 3.402823E+38
- MIN_BOOL = 0
- MIN_INT8 = -128
- MIN_INT16 = -32768
- MIN_INT32 = -2147483648
- MIN_UINT8 = 0
- MIN_UINT16 = 0
- MIN_UINT32 = 0
- MIN_CHAR = 0
- MIN_ENUM = 0
- MIN_FLOAT = -3.402823E+38

4.3 Parameter code display

The parameter code is displayed as a hexadecimal number and is structured as follows:

Example: System of units, code 0x7D75

0x	Hexadecimal display
7D	Parameter prefix
75	Parameter index

4.4 Parameter

NOTE

If a parameter value is exceeded or not reached during the writing process, an info message appears, [INFO VALUE LIMIT MIN](#) [► 86] or [INFO VALUE LIMIT MAX](#) [► 86].

4.4.1 Device

4.4.1.1 Device serial number

The parameter shows the serial number of the module.

Code: **0x7D73**
 Access rights, read - write: USER - SCHUNK
 Data type: UINT32
 Parameter value, min. - max.: 0 - MAX_UINT32

4.4.1.2 Actual gripper

The parameter shows whether the module is a gripper.

Code: **0x7D74**
 Access rights, read - write: USER - PROFI
 Data type: BOOL
 Parameter value, min. - max.: false - true

4.4.1.3 Unit system

The parameter shows the unit system of the module.

A restart of the module is required after the writing.

Code: **0x7D75**
 Access rights, read - write: USER - USER
 Data type: ENUM
 Parameter value, min. - max.: 0 - 9

Parameter value	Designation
0	mm
1	m
2	Inch
3	Rad
4	Degrees
5	INTERNAL
6	µm (integer)
7	µGrad (integer)
8	µInch (integer)
9	mGrad (integer)

4.4.1.4 Inverted direction of rotation of motor

The parameter shows the direction of rotation of the motor.
A restart of the module is required after the writing.

Code:	0x7D76
Access rights, read - write:	USER - PROFI
Data type:	BOOL
Parameter value, min. - max.:	false - true

NOTE

An incorrect setting may lead to unexpected effects, e.g. the module turns at an unexpectedly high speed.

The parameter has direct interaction with the following parameter:

- Parameter [Inverted position measurement](#) [► 26]
 - If the direction of rotation of the motor and the position measuring system are inverted at the same time, then a right-turning module can be configured from a left-turning one, or a positive closing gripper can be configured from a positive opening gripper.

4.4.1.5 Inverted position measurement

The parameter shows the measuring direction of the position measuring system.

A restart of the module is required after the writing.

Code:	0x7D77
Access rights, read - write:	USER - PROFI
Data type:	BOOL
Parameter value, min. - max.:	false - true

NOTE

An incorrect setting may lead to unexpected effects, e.g. the module turns at an unexpectedly high speed.

If the A and B encoder tracks are interchanged, this can be exchanged using the software.

The parameter has direct interaction with the following parameter:

- Parameter [Inverted direction of rotation of motor](#) [► 26]
 - If the direction of rotation of the motor and the position measuring system are inverted at the same time, then a right-turning module can be configured from a left-turning one, or a positive closing gripper can be configured from a positive opening gripper.

4.4.1.6 Endless

The parameter shows whether the axis rotates endlessly or if software end stops are taken into account.

Code: **0x7D78**
 Access rights, read - write: USER - PROFI
 Data type: BOOL
 Parameter value, min. - max.: false - true

With rotary modules, you can set whether or not the module rotates endlessly.

The parameter has direct interaction with the following parameters:

- Parameter [Min. position](#) [► 28]
 - Not taken into account when "Endless" is set.
- Parameter [Max. position](#) [► 28]
 - Not taken into account when "Endless" is set.

4.4.1.7 Digital outputs

The parameter shows how digital outputs are used.

Code: **0x7D7A**
 Access rights, read - write: USER - PROFI
 Data type: ENUM
 Parameter value, min. - max.: see the following table

Parameter value	Designation
0	Normal
1	Status + movement OUT2
2	Status + position reached OUT2
3	Status + brake OUT2
4	Status + warning OUT2
5	Status + program sequence OUT2

4.4.1.8 Min. position

The parameter shows the minimum position of the software stop.

Code:	0x7D7B
Access rights, read - write:	USER - PROFI
Data type:	FLOAT/INT32
Parameter value, min. - max.:	MIN_FLOAT - Max. Position [► 28]

If the position setting exceeds this value, an info message appears, [INFO VALUE LIMIT MIN](#) [► 86] and the position setting is automatically corrected according to this value.

The parameter has direct interaction with the following parameters:

- Parameter [Endless](#) [► 27]
 - Not taken into account when "Endless" is set.
- Parameter [Referencing type](#) [► 43]
 - Is used for referencing with stroke control.

4.4.1.9 Max. position

The parameter shows the maximum position of the software stop.

Code:	0x7D7C
Access rights, read - write:	USER - PROFI
Data type:	FLOAT/INT32
Parameter value, min. - max.:	Min. Position [► 28] - MAX_FLOAT

If the position setting exceeds this value, an info message appears, [INFO VALUE LIMIT MAX](#) [► 86] and the position setting is automatically corrected according to this value.

The parameter has direct interaction with the following parameters:

- Parameter [Endless](#) [► 27]
 - Not taken into account when "Endless" is set.
- Parameter [Referencing type](#) [► 43]
 - Is used for referencing with stroke control.

4.4.1.10 Min. main board temperature

The parameter shows the minimum permissible working temperature for the main board.

Code: **0x7D7D**
 Access rights, read - write: USER - ADVANCED
 Data type: FLOAT
 Parameter value, min. - max.: MIN_FLOAT - [Max. main board temperature](#) [► 29]

If the parameter value is not reached, an error message occurs, [ERROR TEMP LOW](#) [► 92].

4.4.1.11 Max. main board temperature

The parameter shows the maximum permissible working temperature for the main board.

Code: **0x7D7E**
 Access rights, read - write: USER - ADVANCED
 Data type: FLOAT
 Parameter value, min. - max.: [Min. main board temperature](#) [► 29] - MAX_FLOAT

If the temperature value exceeds the working temperature, there will be an warning. There will be an error message if the temperature does not drop within 1 min, [ERROR TEMP HIGH](#) [► 92].

4.4.1.12 Min. motor temperature

The parameter shows the minimum permissible operating temperature for the motor when the motor temperature sensor is connected.

Code: **0x7D7F**
 Access rights, read - write: USER - ADVANCED
 Data type: FLOAT
 Parameter value, min. - max.: MIN_FLOAT - [Max. motor temperature](#) [► 30]

If the parameter value is not reached, an error message occurs, [ERROR MOTOR TEMP LOW](#) [► 92].

4.4.1.13 Max. motor temperature

The parameter shows the minimum permissible operating temperature for the motor when the motor temperature sensor is connected.

Code:	0x7D80
Access rights, read - write:	USER - ADVANCED
Data type:	FLOAT
Parameter value, min. - max.:	Min. motor temperature [► 29] - MAX_FLOAT

If the parameter value is exceeded, an error message occurs, [ERROR MOTOR TEMP HIGH](#) [► 92].

4.4.1.14 Min. temperature of the communication board

The parameter shows the minimum permissible working temperature for the communication board.

Code:	0x7D81
Access rights, read - write:	USER - ADVANCED
Data type:	FLOAT
Parameter value, min. - max.:	MIN_FLOAT - Max. temperature of the communication board [► 30]

If the parameter value is not reached, an error message occurs, [ERROR CONNECTION TEMP LOW](#) [► 91].

4.4.1.15 Max. temperature of the communication board

The parameter shows the maximum permissible working temperature for the communication board.

Code:	0x7D82
Access rights, read - write:	USER - ADVANCED
Data type:	FLOAT
Parameter value, min. - max.:	Min. temperature of the communication board [► 30] - MAX_FLOAT

If the parameter value is exceeded, an error message occurs, [ERROR CONNECTION TEMP HIGH](#) [► 91].

4.4.1.16 Saving energy

The parameter shows whether the LEDs are switched off.

Code:	0x7D83
Access rights, read - write:	USER - USER
Data type:	BOOL
Parameter value, min. - max.:	false - true

4.4.2 Motor**4.4.2.1 Motor serial number**

The parameter shows the serial number of the motor.

Code:	0x7D1E
Access rights, read - write:	USER - SCHUNK
Data type:	UINT32
Parameter value, min. - max.:	0 - MAX_UINT32

The parameter value can only be written if the motor is de-energized.

4.4.2.2 Motor voltage

The parameter shows the nominal voltage of the motor [V].

Code:	0x7D1F
Access rights, read - write:	USER - ROOT
Data type:	UINT8
Parameter value, min. - max.:	24 - 48

The parameter value can only be written if the motor is de-energized.

CAUTION

An incorrect input can destroy the electronics.

4.4.2.3 Motor Type

The parameter shows the selected motor.

A restart of the module is required after the writing.

Code: **0x7D20**
 Access rights, read - write: USER - ADVANCED
 Data type: ENUM
 Parameter value, min. - max.: see the following table

Parameter value	Designation
0	DC Brush type DC motor
1	BLDC Electronically commutated brushless DC motor with block commutation.
2	PMSM Electronically commutated brushless DC motor with sine commutation.

The parameter value can only be written if the motor is de-energized.

CAUTION

An incorrect input can destroy the electronics.

4.4.2.4 I²T

The parameter shows the strength of the I²T monitoring [%].

Code: **0x7D21**
 Access rights, read - write: USER - ADVANCED
 Data type: UINT8
 Parameter value, min. - max.: 10 - 100

The parameter value can only be written if the motor is de-energized.

An I²T [error](#) **ERROR I²T** [► 90] will be triggered if the load is too high. With I²T monitoring, it is assumed that the maximum current may be present for three seconds (corresponds to 100%). If a value of <100% is entered, the time is reduced and the I²T monitoring triggers earlier.

4.4.2.5 Max. current

The parameter shows the maximum permissible current.

Code:	0x7D22
Access rights, read - write:	USER - ADVANCED
Data type:	FLOAT/INT32
Parameter value, min. - max.:	0 - Metering current range

The parameter value can only be written if the motor is de-energized.

CAUTION

An incorrect input can destroy the electronics.

If the value is exceeded for a period of several milliseconds, a fast stop is triggered and an error message appears, [ERROR CURRENT](#) [► 90].

4.4.2.6 Nom. Current

The parameter shows the maximum current that is allowed to flow permanently.

Code:	0x7D23
Access rights, read - write:	USER - ADVANCED
Data type:	FLOAT/INT32
Parameter value, min. - max.:	0 - Max. current [► 33]

The parameter value can only be written if the motor is de-energized.

CAUTION

An incorrect input can destroy the electronics.

If the value is exceeded for a period of several seconds, an error message appears, [ERROR I2T](#) [► 90].

4.4.2.7 Max. speed

The parameter shows the maximum permissible speed (output side).

Code:	0x7D24
Access rights, read - write:	USER - ADVANCED
Data type:	FLOAT/INT32
Parameter value, min. - max.:	0 - MAX_FLOAT/MAX_INT32

The parameter value can only be written if the motor is de-energized.

4.4.2.8 Max. acceleration

The parameter shows the maximum permissible acceleration (output side).

Code:	0x7D25
Access rights, read - write:	USER - ADVANCED
Data type:	FLOAT/INT32
Parameter value, min. - max.:	0 - MAX_FLOAT/MAX_INT32

The parameter value can only be written if the motor is de-energized.

4.4.2.9 Max. jerk

The parameter shows the maximum permissible jerk (output side).
The parameter value can only be written if the motor is de-energized.

Code:	0x7D26
Access rights, read - write:	USER - ADVANCED
Data type:	FLOAT/INT32
Parameter value, min. - max.:	0 - MAX_FLOAT/MAX_INT32

The jerk is the temporal alteration of acceleration.

The parameter is only analyzed if a position movement with jerk limitation is executed.

The parameter has direct interaction with the following parameter:

- Parameter [Position ramp](#) [► 55]

4.4.2.10 Pole pairings

The parameter shows the electrical pole pairing of the motor.

Code:	0x7D27
Access rights, read - write:	USER - ROOT
Data type:	UINT8
Parameter value, min. - max.:	0 - MAX_UINT8

The parameter value can only be written if the motor is de-energized.

The parameter is only required for brushless DC motors and affects the calculation of the speed, position and commutation pattern.

The parameter has direct interaction with the following parameter:

- Parameter [Motor Type](#) [► 32]

4.4.2.11 Terminal resistance

The parameter shows the value of the terminal resistance [Ohm].

Code:	0x7D28
Access rights, read - write:	USER - ROOT
Data type:	FLOAT
Parameter value, min. - max.:	0 - MAX_FLOAT

The parameter value can only be written if the motor is de-energized.

CAUTION

An incorrect input can destroy the electronics.

The terminal resistance is used for test functions to limit maximum currents as well as for automatic controller configuration.

4.4.2.12 Inductance

The parameter shows the value of the inductance [H].

Code:	0x7D29
Access rights, read - write:	USER - ROOT
Data type:	FLOAT
Parameter value, min. - max.:	0 - MAX_FLOAT

The parameter value can only be written if the motor is de-energized.

CAUTION

An incorrect input can destroy the electronics.

4.4.2.13 Motor constant

The parameter shows the value of the motor constant.

Code:	0x7D2A
Access rights, read - write:	USER - ROOT
Data type:	FLOAT/INT32
Parameter value, min. - max.:	0 - MAX_FLOAT/MAX_INT32

The parameter value can only be written if the motor is de-energized.

4.4.2.14 Commutation table

The parameter shows the number of the valid hall sensor table for block commutation.

Code:	0x7D2B
Access rights, read - write:	USER - ROOT
Data type:	UINT8
Parameter value, min. - max.:	0 - 5

The parameter value can only be written if the motor is de-energized.

If the entry is incorrect, the motor will not move at all or only generates very little torque.

4.4.2.15 Metering current range

The parameter shows the maximum metering current range for the current sensor used internally.

A restart of the module is required after the writing.

Code:	0x7D2C
Access rights, read - write:	USER - SCHUNK
Data type:	FLOAT/INT32
Parameter value, min. - max.:	0 - MAX_FLOAT/MAX_INT32

The parameter value can only be written if the motor is de-energized.

4.4.2.16 Max. measurement discrepancies

The parameter shows the maximum measurement difference [A].

Code:	0x7D2D
Access rights, read - write:	USER - ROOT
Data type:	FLOAT
Parameter value, min. - max.:	1 MAX_FLOAT

The parameter value can only be written if the motor is de-energized.

A third current sensor is necessary to calculate the maximum measurement difference. The maximum permitted deviation is $A + B + C = 0$. If the deviation is greater than the set value, an error message appears, [ERROR MOTOR PHASE](#) [► 88].

The value "-1" deactivates the third current sensor. This is not supported by all hardware variants. A motor phase break or motor phase short circuit is not detected by the deactivation of the sensor.

When the third current sensor is activated, the current sensors must be calibrated.

4.4.2.17 Offset phase A

The parameter shows the zero point alignment for the first current sensor.

Code:	0x7D2E
Access rights, read - write:	USER - SCHUNK
Data type:	UINT16
Parameter value, min. - max.:	1500 - 2600

The parameter value can only be written if the motor is de-energized.

NOTE

An incorrect value may lead to an unpredictable behavior in the drive (e.g., only travels in one direction, jerks badly).

4.4.2.18 Offset phase B

The parameter shows the zero point alignment for the second current sensor.

Code:	0x7D2F
Access rights, read - write:	USER - SCHUNK
Data type:	UINT16
Parameter value, min. - max.:	1500 - 2600

The parameter value can only be written if the motor is de-energized.

NOTE

An incorrect value may lead to an unpredictable behavior in the drive (e.g., only travels in one direction, jerks badly).

4.4.2.19 Offset phase C

The parameter shows the zero point alignment for the third current sensor.

Code:	0x7D30
Access rights, read - write:	USER - SCHUNK
Data type:	UINT16
Parameter value, min. - max.:	1500 - 2600

The parameter value can only be written if the motor is de-energized.

NOTE

An incorrect value may lead to an unpredictable behavior in the drive (e.g., only travels in one direction, jerks badly).

If the hardware does not support third current sensors, this value is always 2048.

4.4.3 Controller

4.4.3.1 KR current

The parameter shows the proportional part of current controller.

Code:	0x7D4B
Access rights, read - write:	USER - PROFI
Data type:	FLOAT
Parameter value, min. - max.:	0 - MAX_FLOAT

4.4.3.2 TN current

The parameter shows the integral part of the current controller.

Code:	0x7D4C
Access rights, read - write:	USER - PROFI
Data type:	FLOAT
Parameter value, min. - max.:	0 - MAX_FLOAT

This is not required for current limiting control, MCSL_Struktur.

4.4.3.3 TD current

The parameter shows the differential part of the current controller.

Code:	0x7D4D
Access rights, read - write:	USER - ROOT
Data type:	FLOAT
Parameter value, min. - max.:	0 - MAX_FLOAT

4.4.3.4 KC current

The parameter shows the correction factor of the current controller for the integral part (anti-windup).

Code:	0x7D4E
Access rights, read - write:	USER - ROOT
Data type:	FLOAT
Parameter value, min. - max.:	0 - MAX_FLOAT

4.4.3.5 KR speed

The parameter shows the proportional part of the speed controller.

Code:	0x7D4F
Access rights, read - write:	USER - PROFI
Data type:	FLOAT
Parameter value, min. - max.:	0 - MAX_FLOAT

4.4.3.6 TN speed

The parameter shows the integral part of the speed controller.

Code: **0x7D50**
 Access rights, read - write: USER - PROFI
 Data type: FLOAT
 Parameter value, min. - max.: 0 - MAX_FLOAT

4.4.3.7 TD speed

The parameter shows the differential part of the speed controller.

Code: **0x7D51**
 Access rights, read - write: USER - ROOT
 Data type: FLOAT
 Parameter value, min. - max.: 0 - MAX_FLOAT

4.4.3.8 KC speed

The parameter shows the correction factor of the current controller for the integral part (anti-windup).

Code: **0x7D52**
 Access rights, read - write: USER - ROOT
 Data type: FLOAT
 Parameter value, min. - max.: 0 - MAX_FLOAT

4.4.3.9 KR position

The parameter shows the proportional part of position controller.

Code: **0x7D53**
 Access rights, read - write: USER - PROFI
 Data type: FLOAT
 Parameter value, min. - max.: 0 - MAX_FLOAT

4.4.3.10 TN position

The parameter shows the integral part of position controller.

Code: **0x7D54**
 Access rights, read - write: USER - PROFI
 Data type: FLOAT
 Parameter value, min. - max.: 0 - MAX_FLOAT

4.4.3.11 TD position

The parameter shows the differential part of the position controller.

Code: **0x7D55**
 Access rights, read - write: USER - ROOT
 Data type: FLOAT
 Parameter value, min. - max.: 0 - MAX_FLOAT

4.4.3.12 KC position

The parameter shows the correction factor of the current controller for the integral part (anti-windup).

Code:	0x7D56
Access rights, read - write:	USER - ROOT
Data type:	FLOAT
Parameter value, min. - max.:	0 - MAX_FLOAT

4.4.3.13 Speed feed-forward control

The parameter shows the feed-forward control factor of the speed controller.

Code:	0x7D57
Access rights, read - write:	USER - ROOT
Data type:	FLOAT
Parameter value, min. - max.:	0 - MAX_FLOAT

4.4.3.14 Current feed-forward control

The parameter shows the feed-forward control factor of the current controller.

Code:	0x7D58
Access rights, read - write:	USER - ROOT
Data type:	FLOAT
Parameter value, min. - max.:	0 - MAX_FLOAT

4.4.3.15 Position deviation

The parameter shows the position window in which the position control is terminated.

Code:	0x7D59
Access rights, read - write:	USER - PROFI
Data type:	FLOAT/INT32
Parameter value, min. - max.:	0 - MAX_FLOAT/MAX_INT32

Whether the brake will be applied or the "Position reached" will be triggered is controlled depending on the brake configuration [Brake usage](#) [► 59].

4.4.3.16 Max. overshoot

The parameter shows how much overshoot the module is allowed during a movement.

Code: **0x7D5A**
 Access rights, read - write: USER - ADVANCED
 Data type: FLOAT/INT32
 Parameter value, min. - max.: 0 - MAX_FLOAT/MAX_INT32

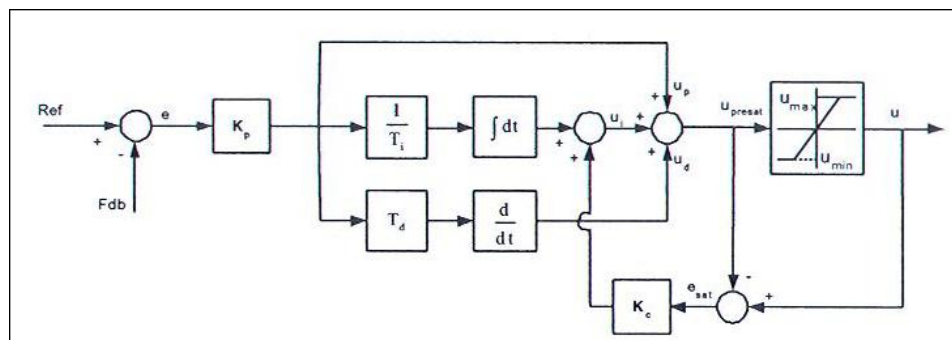
If the module moves past this position window during a positioning movement, then the error message "ERROR OVERSHOOT" is generated, [ERROR OVERSHOOT](#) [► 94]. This value must be set greater than the maximum permissible position variation. [Position deviation](#) [► 40].

4.4.3.17 Structure

The parameter shows the structural composition of the control circuit.

Code: **0x7D5B**
 Access rights, read - write: USER - PROFI
 Data type: ENUM
 Parameter value, min. - max.: see the following table

Parameter value	Designation
0	Current speed
1	Cascade
2	Speed with current limiting
3	Speed with PWM limiting
4	Position cascade



Controllers, structure

All controllers are realized as PID controllers with anti-windup functionality. The complete parameter set is only to be accessed via the root rights in SCHUNK mode.

The following parameters can be set:

- KR: proportional part of the respective controller
- TN: integral part of the respective controller
- TD: differential part of the respective controller
- KC: correction factor for the integral part
- Current - Speed
Current control and speed control operate independently of each other.
- Cascade
Position, speed, and current controllers are cascade connected => Current controlled (set current is not to be exceeded) positioning or speed movements are possible (e.g. no pre-positioning required for a gripping process necessary). In this mode, the specified current is not exceeded in all motion types.
- Speed with current limiting
Current control is not active. The specified current is limited for speed or position movements. In contrast to the cascade, the current is not controlled, but limited (current limiting control).
- Speed with PWM limiting
Current control is not active. The duty cycle of the PWM is limited for speed or position movements. The ratio of current to duty cycle is calculated using the terminal resistance of the motor, [Terminal resistance](#) [► 35].

Since the PWM's duty cycle is directly limited (voltage limitation), it is possible that the motor no longer reaches its full speed. Position movements may take considerably longer than anticipated.

If the controller structure is changed, the controller parameters may have to be adapted.

4.4.4 Referencing

4.4.4.1 Referencing type

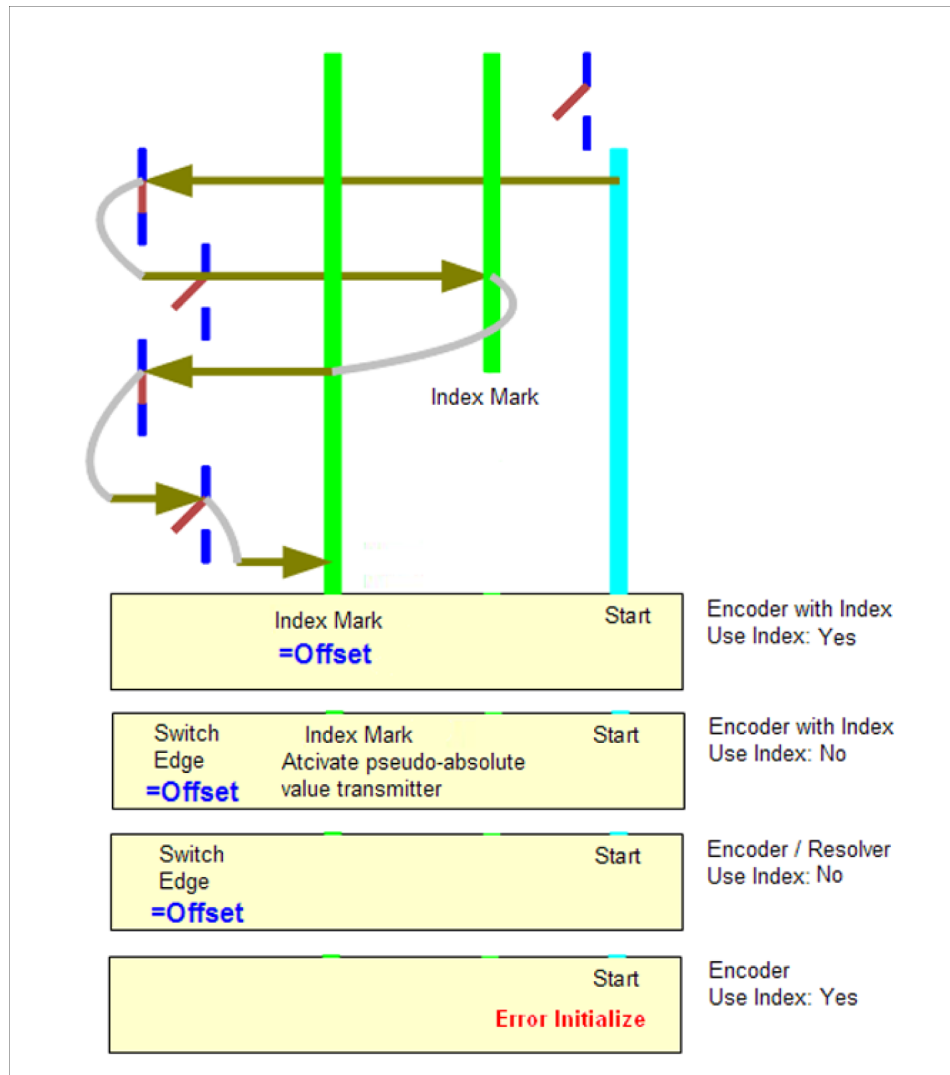
The parameter shows the type of referencing.

A restart of the module is required after the writing.

Code: **0x7D41**
 Access rights, read - write: USER - PROFI
 Data type: ENUM
 Parameter value, min. - max.: see the following table

Parameter value	Designation
0	Left side internal switch
1	Right side internal switch
2	Left side external IN0 switch
3	Right side external IN0 switch
4	Speed, left
5	Speed, right
6	Speed, left (stroke monitoring)
7	Speed, right (stroke monitoring)
8	Current, left
9	Current, right
10	Current, left (stroke monitoring)
11	Current, right (stroke monitoring)
12	None

When using an encoder with index track, observe the "Positioning type" parameter, [Positioning type](#) [► 50].



Referencing with switch

NOTE

The encoder type can be read out using the "Motion Tool SCHUNK" (MTS) software via the "Positioning Type" parameter (0x7D60). Readout is also possible via acyclic data traffic. Further information is contained in the chapter [Positioning type](#) [► 50] and the software manual "Motion Tool SCHUNK (MTS)".

- Internal switch left/right
 - The internal reference switch is used for referencing. The direction of motion when the reference switch is active is determined by the direction "left" or "right".
- External IN1 switch left/right
 - An external reference switch (IN1) is used for referencing. The direction of motion when the reference switch is active is determined by the direction "left" or "right".

When referencing with a switch, it must be ensured that the switch flack of the proximity switch is applied for at least 200 ms. If necessary, also adjust the referencing speed and the switching cams.

NOTE

SCHUNK recommends a basic referencing process after installing the machine/automated system on the module. If the position or load is changed during referencing, SCHUNK also recommends a basic referencing process. Without a basic referencing process, the error message "NOT REFERENCED" may occur, [NOT REFERENCED](#) [► 85].

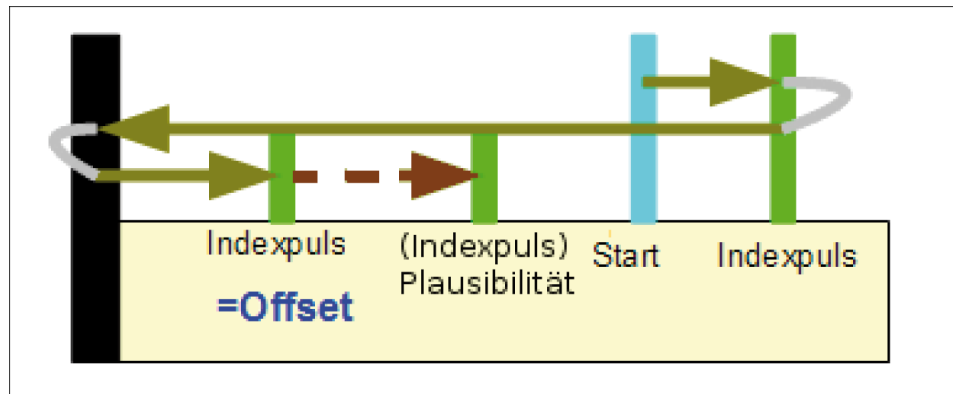
If a fixed end stop is available, SCHUNK recommends the following referencing types:

- Speed, left/right
 - A speed run is performed for referencing. If the module moves to a fixed stop, then this will be recognized as a reference point. The direction of rotation is defined via "left" or "right".
- Speed with stroke monitoring left/right
 - In addition to the procedure described above, a movement is made to the opposite fixed stop after the first fixed stop has been approached. The traveled distance must be greater than the difference in the software limit ranges => Referencing successful, [Min. position](#) [► 28].
- Current, left/right

NOTE

Jamming, sluggishness in the mechanical system, or a "forgotten" workpiece can also lead to the rated current being exceeded. This would then also be interpreted as a fixed end stop, although none are present.

- A current run is executed. The current is increased until the module moves. If the current exceeds the max. reference current, then it is assumed that a fixed stop has been reached which is recognized as a reference point, [Referencing max. current](#) [► 48].
- Current with stroke monitoring left/right
 - In addition to the procedure described above, a movement is made to the opposite fixed end stop after the first fixed stop has been approached. The traveled distance must be greater than the difference in the software limit ranges => Referencing successful, [Min. position](#) [► 28].
- None



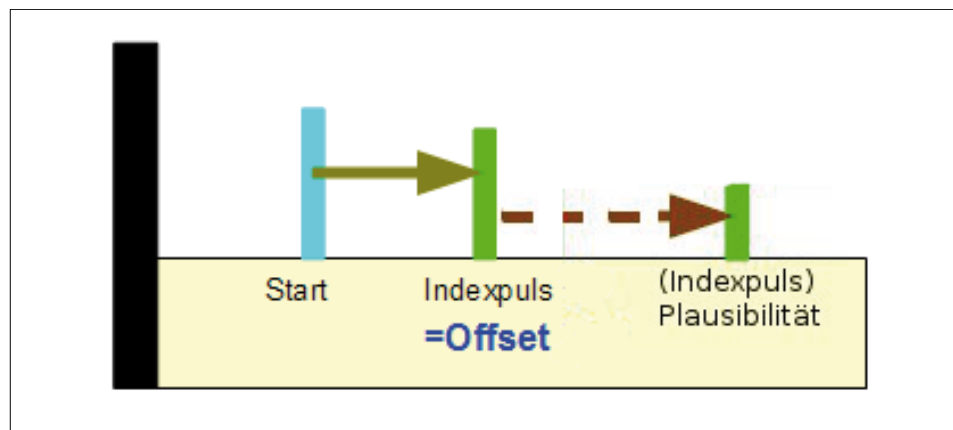
Referencing up to stop with activated index track

- The current position is seen as the reference position.

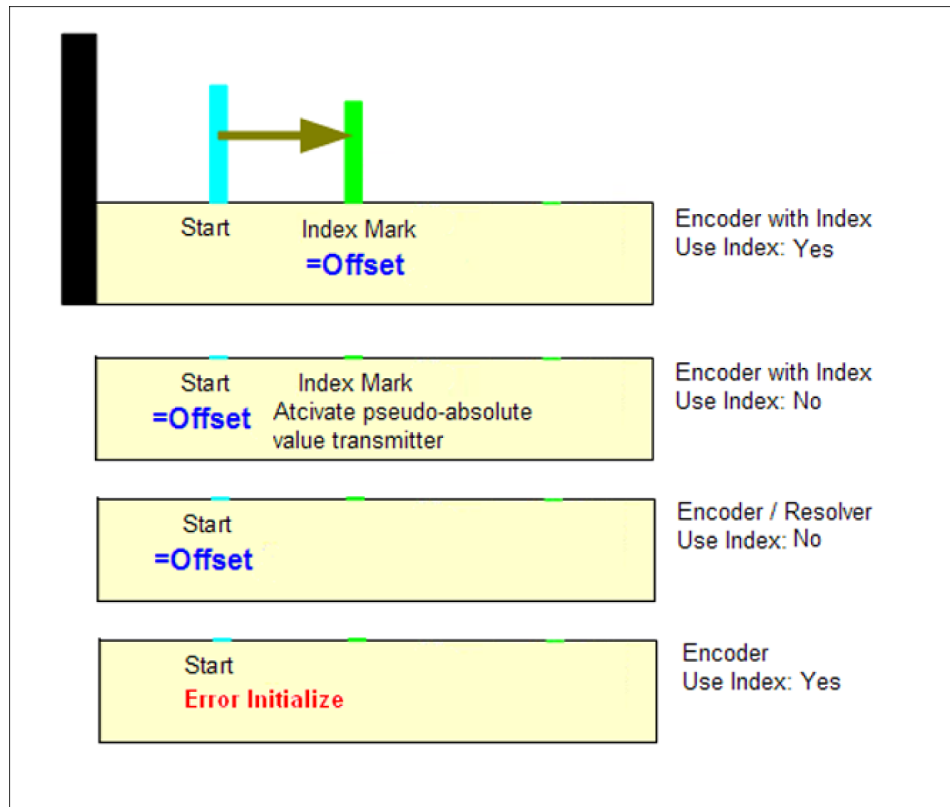
4.4.4.2 Usage index

The parameter shows whether the index track of the encoder is analyzed during referencing.

Code:	0x7D42
Access rights, read - write:	USER - PROFI
Data type:	BOOL
Parameter value, min. - max.:	false - true



Referencing of "none" with activated index track



Referencing of "none"

NOTE

The encoder type can be read out using the "Motion Tool SCHUNK" (MTS) software via the "Positioning Type" parameter (0x7D60). Readout is also possible via acyclic data traffic. Further information is contained in the chapter [Positioning type](#) [► 50] and the software manual "Motion Tool SCHUNK (MTS)".

NOTE

Should be referenced with index pulse. If the index pulse is in a poor position, it can happen that the positions differ by one motor rotation each after repeated referencing. Remedy: Shift the reference mark slightly. This applies to all reference marks except "internal switch" and "external switch".

4.4.4.3 Distance to the index

The parameter shows the distance from the reference event (switch flank detected) to the index pulse.

Code:	0x7D43
Access rights, read - write:	USER - PROFI
Data type:	UINT16
Parameter value, min. - max.:	0 - MAX_UINT16

If the parameter is set to "0", the distance is measured and stored during the next reference movement.

For subsequent reference movements, this is measured again and compared with the saved value. If the values are within a set tolerance range, then the referencing can be successfully completed, [Max. distance switch referencing](#) [► 50]. Furthermore, an index pulse at an unfavorable point (index pulse shortly before or after switch flanks) can be corrected.

If the distance between index pulse and reference event exceeded the permissible tolerance, then referencing will be aborted and "ERROR REFERENCED" displayed, [NOT REFERENCED](#) [► 85].

It is necessary to remeasure the distance from the reference event to the index pulse. To do that, the parameter "Distance to index" must be set to "0", [Distance to the index](#) [► 48].

4.4.4.4 Approach 0 after referencing

The parameter shows whether the position "0" is approached after successful referencing.

Code:	0x7D44
Access rights, read - write:	USER - PROFI
Data type:	BOOL
Parameter value, min. - max.:	false - true

4.4.4.5 Referencing max. current

The parameter shows the current setting from the nominal current of the motor [%]. The reference current does not exceed the specified value.

Code:	0x7D45
Access rights, read - write:	USER - PROFI
Data type:	UINT8
Parameter value, min. - max.:	0 - 200

If the maximum permissible reference current is not sufficient enough to move the module, the maximum reference current must be increased.

4.4.4.6 Speed referencing

The parameter shows the speed settings for reference movements.

Code:	0x7D46
Access rights, read - write:	USER - PROFI
Data type:	FLOAT/INT32
Parameter value, min. - max.:	0 - Max. speed [► 33]

4.4.4.7 Acceleration referencing

The parameter shows the acceleration settings for reference movements with internal or external reference switch, and speed reference movements.

Code:	0x7D47
Access rights, read - write:	USER - PROFI
Data type:	FLOAT/INT32
Parameter value, min. - max.:	0 - Max. acceleration [► 34]

4.4.4.8 Offset referencing

The parameter shows the position offset after successful referencing (zero offset).

Code:	0x7D48
Access rights, read - write:	USER - PROFI
Data type:	FLOAT/INT32
Parameter value, min. - max.:	MIN_FLOAT/MIN_INT32 - MAX_FLOAT/MAX_INT32

4.4.4.9 Timeout referencing

The parameter shows the time that a reference movement is permitted to last.

Code:	0x7D49
Access rights, read - write:	USER - PROFI
Data type:	FLOAT/INT32
Parameter value, min. - max.:	0 - MAX_FLOAT/MAX_INT32

If the time is exceeded, the motor is de-energized and an error message appears [NOT REFERENCED](#) [► 85].

4.4.4.10 Max. distance switch referencing

The parameter shows the maximum distance from the reference event (switch flack detected) to the index pulse.

Code: **0x7D4A**
 Access rights, read - write: USER - ADVANCED
 Data type: UINT16
 Parameter value, min. - max.: 0 - MAX_UINT16

If one encoder tick is specified.

4.4.5 Positioning

4.4.5.1 Positioning serial number

The parameter shows the serial number of the position measuring system.

Code: **0x7D5F**
 Access rights, read - write: USER - SCHUNK
 Data type: UINT32
 Parameter value, min. - max.: 0 - MAX_UINT32

The parameter value can only be written if the motor is de-energized.

4.4.5.2 Positioning type

The parameter shows the measuring system type.

A restart of the module is required after the writing.

Code: **0x7D60**
 Access rights, read - write: USER - ADVANCED
 Data type: ENUM
 Parameter value, min. - max.: see the following table

Parameter value	Designation
0	Encoder
1	Encoder Index
2	Resolver
6	Encoder differential
7	Encoder index differential
8	Analog

The parameter value can only be written if the motor is de-energized.

After a new measuring system has been selected, a restart of the module is required.

- Encoder
 - Encoder measuring system without index track.
- Encoder index
 - Encoder measuring system with index track.
For the reference movements, the index track is analyzed depending on the configuration, [Usage index](#) [► 46]. When using the index track, it is possible for the motor to move back and forth several times in short movements or to make small movements in the "wrong" direction during the referencing process. The modules also move with the "None" referencing type, due to the search for the next index pulse , [Pseudo-absolute encoder](#) [► 9].
- Resolver
 - Resolver system with adjustable exciting current
- Encoder differential
 - Differential encoder without index track
- Encoder index differential
 - Differential encoder with index track
For reference movements, the index track is analyzed depending on the configuration, [Usage index](#) [► 46]. When using the index track, it is possible for the motor to move back and forth several times in short movements or to make small movements in the "wrong" direction during the referencing process. The modules also move with the "None" referencing type, due to the search for the next index pulse , [Pseudo-absolute encoder](#) [► 9].

4.4.5.3 Positioning installation

The parameter shows the installation position of the position measuring system.

The parameter value can only be written if the motor is de-energized.

A restart of the module is required after the writing.

Code: **0x7D61**
 Access rights, read - write: USER - ADVANCED
 Data type: ENUM
 Parameter value, min. - max.: see the following table

Parameter value	Designation
0	Drive side
1	Output side
2	Between gear ratios

- Drive side
 - The position measuring system is mounted directly on the drive end.
- Output side
 - The position measuring system is mounted directly on the output end.
- Between gear ratios
 - The position measuring system is mounted in the middle of the gear.

4.4.5.4 Ticks per revolution

The parameter shows the ticks per revolution.

The parameter value can only be written if the motor is de-energized.

A restart of the module is required after the writing.

Code: **0x7D62**
 Access rights, read - write: USER - ADVANCED
 Data type: UINT16
 Parameter value, min. - max.: 512 - MAX_UINT16

4.4.5.5 Exciter amplitude

The parameter shows the amplitude of the input voltage at the exciter coil [%].

Code: **0x7D63**
 Access rights, read - write: USER - ADVANCED
 Data type: UINT8
 Parameter value, min. - max.: 0 - 100

The parameter value can only be written if the motor is de-energized.

4.4.5.6 Exciter frequency

The parameter shows the voltage frequency at the exciter coil [kHz].

Code: **0x7D64**
 Access rights, read - write: USER - ADVANCED
 Data type: ENUM
 Parameter value, min. - max.: see the following table

Parameter value	Designation
0	1 kHz
1	2 kHz
2	4 kHz
3	8 kHz

The parameter value can only be written if the motor is de-energized.

4.4.5.7 ADC offset positioning

The parameter shows the "centering" of the input signal at the resolver.

Code: **0x7D65**
 Access rights, read - write: USER - ADVANCED
 Data type: FLOAT
 Parameter value, min. - max.: MIN_FLOAT - MAX_FLOAT

The parameter value can only be written if the motor is de-energized.

4.4.5.8 Offset positioning

The parameter shows the twisting of the position measuring system opposite the motor phases.

Code:	0x7D68
Access rights, read - write:	USER - ADVANCED
Data type:	FLOAT/INT32
Parameter value, min. - max.:	MIN_FLOAT - MAX_FLOAT/ MIN_INT32 - MAX_INT32

The parameter value can only be written if the motor is de-energized.

This value may be automatically determined, [Standstill commutation](#) [► 10].

4.4.5.9 Motion threshold

The parameter shows the value in percent [%] of the maximum permissible speed.

Code:	0x7D69
Access rights, read - write:	USER - PROFI
Data type:	FLOAT
Parameter value, min. - max.:	1 - 100

The parameter value can only be written if the motor is de-energized.

If the current speed falls below this value, the module is treated as if it is standing still. The "Module is moving" status display is off.

4.4.5.10 Position waiting time reached

The parameter shows the reset time deceleration for the "Position reached" flag.

Code:	0x7D6A
Access rights, read - write:	USER - PROFI
Data type:	FLOAT/INT32
Parameter value, min. - max.:	0 - MAX_FLOAT

SCHUNK recommends setting this value somewhat larger than the PLC cycle.

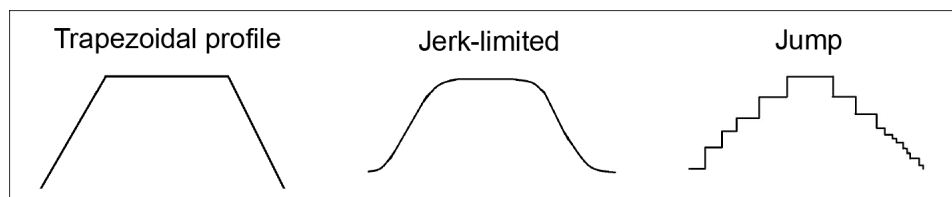
4.4.5.11 Position ramp

The parameter shows the ramp type for the position movement
A restart of the module is required after the writing.

Code: **0x7D6B**
 Access rights, read - write: USER - PROFI
 Data type: ENUM
 Parameter value, min. - max.: see the following table

Parameter value	Designation
0	Trapezoid V2
1	Jerk-limited
2	Trapezoid V3
3	Jump

The parameter value can only be written if the motor is de-energized.



- **Trapezoid V2**
A trapezoid is taken as the basis for the calculation of the motion profile. The traveling time is not calculated. Switching points are controlled according to the positions.
- **Jerk-limited**
A path with jerk-limiting is calculated for the position movement. The motion parameter "Jerk" is used for this ramp type.
- **Trapezoid V3**
A trapezoid is taken as the basis for the calculation of the motion profile. The traveling time is not calculated. Switching points are controlled according to the positions.
- **Jump**
A path profile is not calculated here, the position jump is directly specified instead. Internal path planning is switched off. Depending on the interpolation interval of the external interpolator, it may be necessary to adapt the "KC current" control parameter, [KC current](#) [► 38].

4.4.5.12 Towing error

The parameter shows the maximum permissible towing error value.

Code:	0x7D6C
Access rights, read - write:	USER - PROFI
Data type:	FLOAT
Parameter value, min. - max.:	0 - MAX_FLOAT

This towing error value must not be exceeded during a positioning movement. Exceeding this value results in an "ERROR TOW" error message, [ERROR TOW](#) [► 89].

The parameter value can only be written if the motor is de-energized.

4.4.5.13 Motion waiting time blocked

The parameter shows the time that has to elapse in order to trigger the "Motion was blocked" status display [s].

Code:	0x7D6D
Access rights, read - write:	USER - PROFI
Data type:	FLOAT
Parameter value, min. - max.:	0 - MAX_FLOAT

When using the brake, this value may not be less than half the brake timeout. Otherwise the blocking detection will interfere directly after the start of a movement command.

4.4.6 Gear

4.4.6.1 Serial number

The parameter shows the serial number of the gear.

Code:	0x7D37
Access rights, read - write:	USER - SCHUNK
Data type:	UINT32
Parameter value, min. - max.:	0 - MAX_UINT32

The parameter value can only be written if the motor is de-energized.

4.4.6.2 Ratio 1

The parameter shows the gear ratio 1: factor of the reduction from the motor to the drive.

Code:	0x7D38
Access rights, read - write:	USER - PROFI
Data type:	FLOAT
Parameter value, min. - max.:	0 - MAX_FLOAT

The parameter value can only be written if the motor is de-energized.

The parameter has direct interaction with the following parameter:

- Parameter [Positioning installation](#) [► 52]

4.4.6.3 Ratio 2

The parameter shows the gear ratio 2: factor of the reduction from the position measuring system to the drive.

Code:	0x7D39
Access rights, read - write:	USER - PROFI
Data type:	FLOAT
Parameter value, min. - max.:	0 - MAX_FLOAT

The parameter value can only be written if the motor is de-energized.

The parameter is only required if the position measuring system is installed between the gear ratios.

The parameter has direct interaction with the following parameter:

- Parameter [Positioning installation](#) [► 52]

4.4.7 Brake

4.4.7.1 Brake serial number

The parameter shows the serial number of the brake.

Code: **0x7D3C**
 Access rights, read - write: USER - SCHUNK
 Data type: UINT32
 Parameter value, min. - max.: 0 - MAX_UINT32

The parameter value can only be written if the motor is de-energized.

4.4.7.2 Brake type

The parameter shows the type of brake.

The parameter value can only be written if the motor is de-energized.

A restart of the module is required after the writing.

Code: **0x7D3D**
 Access rights, read - write: USER - ADVANCED
 Data type: ENUM
 Parameter value, min. - max.: see the following table

Parameter value	Designation
0	No brake
1	Magnetic 12V
2	Magnetic 24V
3	Magnetic 48V



⚠ DANGER

Danger due to non-functioning brake!

An error in configuration can lead to a non-functioning brake and cause serious injury.

- Check that the configuration is free of errors

The parameter is used to automatically determine the voltage for the brake control.

4.4.7.3 Brake usage

The parameter shows how the brake is used.

The parameter value can only be written if the motor is de-energized.

A restart of the module is required after the writing.

Code: **0x7D3E**
 Access rights, read - write: USER - ADVANCED
 Data type: ENUM
 Parameter value, min. - max.: see the following table

Parameter	Value	Designation	Effect and application with brake	Effect and application without brake
Not used	0	The brake is only applied in the event of voltage failure and is released once, immediately after starting the module.	-	-
Only in case of error	1	The brake is only engaged in case of error and is released with the first movement command. The motor is permanently controlled.	1. Motor control is deactivated in case of error 2. Motor phases short-circuited 3. no warming 4. Brake receives position 5. no force maintenance	1. Motor control is deactivated in case of error 2. Motor phases short-circuited 3. no warming 4. no position maintenance 5. no force maintenance
Normal	2	The brake is engaged in the event of an error and at the end of the movement and the controller goes off.	1. Controller is deactivated when position is reached 2. Motor no longer energized 3. no warming 4. Brake receives position 5. no force maintenance	1. Controller is deactivated when position is reached 2. Motor no longer energized 3. no warming 4. no position maintenance 5. no force maintenance

The parameter has direct interaction with the following function:

- Function [Pseudo-absolute encoder](#) [► 9]
 - If a brake is configured and additional conditions are met, then the pseudo-absolute encoder is active.

4.4.7.4 Brake timeout

The parameter shows the duration of the pause from the end of a traverse movement until the brake is engaged.

Code:	0x7D3F
Access rights, read - write:	USER - ADVANCED
Data type:	FLOAT/INT32
Parameter value, min. - max.:	0 - MAX_FLOAT

The parameter value can only be written if the motor is de-energized.

4.4.8 Voltage

4.4.8.1 Min. motor voltage

The parameter shows the minimum permissible motor voltage.

Code:	0x7D32
Access rights, read - write:	USER - ADVANCED
Data type:	FLOAT
Parameter value, min. - max.:	10 MotorMaxVolt

If the voltage falls below the parameter value, an error message occurs, [ERROR MOTOR VOLTAGE LOW](#) [► 93].

4.4.8.2 Max. motor voltage

The parameter shows the maximum permissible motor voltage.

Code:	0x7D33
Access rights, read - write:	USER - ADVANCED
Data type:	FLOAT
Parameter value, min. - max.:	MotorMinVolt 72

If the voltage falls below the parameter value, an error message occurs, [ERROR MOTOR VOLTAGE HIGH](#) [► 93]. If this error occurs repeatedly, the module is disabled and can only be put into operation again by SCHUNK. If the error occurs frequently due to the load, an external brake chopper can be used.

4.4.8.3 Min. logic voltage

The parameter shows the minimum permissible logic voltage.

Code:	0x7D34
Access rights, read - write:	USER - ADVANCED
Data type:	FLOAT
Parameter value, min. - max.:	5 - LogicMaxVolt

If the voltage falls below the parameter value, an error message occurs, [ERROR LOGIC LOW](#) [► 93].

4.4.8.4 Max. logic voltage

The parameter shows the maximum permissible logic voltage.

Code:	0x7D35
Access rights, read - write:	USER - ADVANCED
Data type:	FLOAT
Parameter value, min. - max.:	MotorMinVolt 30

If the voltage falls below the parameter value, an error message occurs, [ERROR LOGIC HIGH](#) [► 93].

4.4.9 Communication

4.4.9.1 Main communication

The parameter shows the active communication interface.

Code: **0x7D87** (deactivated under PROFIBUS)
 Access rights, read - write: USER - USER
 Data type: ENUM
 Parameter value, min. - max.: see the following table

Parameter value	Designation
0	Auto
1	Serial
2	CAN
3	PROFIBUS
4	Serial without spontaneous messaging
5	CANopen
6	None
7	Anybus
8	USB

4.4.9.2 Module ID

The parameter shows the current module ID.

Code: **0x7D88** (deactivated under PROFIBUS)
 Access rights, read - write: USER - USER
 Data type: UINT8
 Parameter value, min. - max.: 0 - MAX_UINT8

4.4.9.3 Baud rate CAN

The parameter shows the baud rate for the CAN interface.

Code: **0x7D89** (deactivated under PROFIBUS)

Access rights, read - write: USER - USER

Data type: ENUM

Parameter value, min. - max.: see the following table

Parameter value	Designation
0	50 kbit
1	100 kbit
2	125 kbit
3	250 kbit
4	500 kbit
5	1 Mbit

4.4.9.4 Baud rate RS232

The parameter shows the baud rate for the serial interface.

Code: **0x7D8A** (deactivated under PROFIBUS)

Access rights, read - write: USER - USER

Data type: ENUM

Parameter value, min. - max.: see the following table

Parameter value	Designation
0	1200 Baud
1	2400 Baud
2	4800 Baud
3	9600 Baud
4	19200 Baud
5	38400 Baud
6	57600 Baud
7	115200 Baud

4.4.9.5 Protocol mode

The parameter shows the communication protocol between the module and PLC.

The same communication protocol must be parameterized in the module and in the PLC, otherwise there is no communication.

A restart of the module is required after the writing.

Code: **0x7DF3**
 Access rights, read - write: USER - PROFI
 Data type: ENUM
 Parameter value, min. - max.: see the following table

Parameter value	Designation
0	Undefined
1	SMP Single Element
2	SMP Frame
3	SDP Single Element
4	SDP Frame
5	Ethernet IP
6	CANopen

4.4.10 General

4.4.10.1 EEPROM version

The parameter shows the version of the EEPROM.

Code: **0x7D9B**
 Access rights, read - write: USER - ROOT
 Data type: UINT16
 Parameter value, min. - max.: 0 - MAX_UINT16

4.4.10.2 EEPROM CRC

The parameter shows the checksum for all EEPROM data.

Code: **0x7D9C**
 Access rights, read - write: USER - ROOT
 Data type: UINT16
 Parameter value, min. - max.: 0 - MAX_UINT16

4.4.10.3 Data CRC

The parameter shows the checksum for all module-specific EEPROM data.

Code: **0x7D9D**
 Access rights, read - write: USER - ROOT
 Data type: UINT16
 Parameter value, min. - max.: 0 - MAX_UINT16

4.4.10.4 Configuration mode

The parameter shows which module communicates with the PLC. Different output parameters are stored for each corresponding module. The parameter must be the same in the module and in the PLC, otherwise this leads to an error in the hardware parameterization.

A restart of the module is required after the writing.

Code: **0x7DF4**
 Access rights, read - write: USER - PROFI
 Data type: ENUM
 Parameter value, min. - max.: see the following table

Parameter value	Designation
0	Undefined
1	ERS
2	EGN/EZN
3	Reserved
4	EGL
5	PR
6	PDU
7	PSM
8	PW
9	PEH
10	PRH
11	PRL

4.4.11 Info

4.4.11.1 Error 0

The parameter shows the last error n.

Code: **0x7DA0**
 Access rights, read - write: USER - ROOT
 Data type: UINT8
 Parameter value, min. - max.: 0 - MAX_UINT8

For further information on the info and error messages, see [Info and error messages](#) [► 85].

4.4.11.2 Error 1

The parameter shows the error n-1.

Code:	0x7DA1
Access rights, read - write:	USER - ROOT
Data type:	UINT8
Parameter value, min. - max.:	0 - MAX_UINT8

For further information on the info and error messages, see [Info and error messages](#) [► 85].

4.4.11.3 Error 2

The parameter shows the error n-2.

Code:	0x7DA2
Access rights, read - write:	USER - ROOT
Data type:	UINT8
Parameter value, min. - max.:	0 - MAX_UINT8

For further information on the info and error messages, see [Info and error messages](#) [► 85].

4.4.11.4 Error 3

The parameter shows the error n-3.

Code:	0x7DA3
Access rights, read - write:	USER - ROOT
Data type:	UINT8
Parameter value, min. - max.:	0 - MAX_UINT8

For further information on the info and error messages, see [Info and error messages](#) [► 85].

4.4.11.5 Error 4

The parameter shows the error n-4.

Code:	0x7DA4
Access rights, read - write:	USER - ROOT
Data type:	UINT8
Parameter value, min. - max.:	0 - MAX_UINT8

For further information on the info and error messages, see [Info and error messages](#) [► 85].

4.4.11.6 Error 5

The parameter shows the error n-5.

Code:	0x7DA5
Access rights, read - write:	USER - ROOT
Data type:	UINT8
Parameter value, min. - max.:	0 - MAX_UINT8

For further information on the info and error messages, see [Info and error messages](#) [► 85].

4.4.11.7 Error 6

The parameter shows the error n-6.

Code:	0x7DA6
Access rights, read - write:	USER - ROOT
Data type:	UINT8
Parameter value, min. - max.:	0 - MAX_UINT8

For further information on the info and error messages, see [Info and error messages](#) [► 85].

4.4.11.8 Error 7

The parameter shows the error n-7.

Code:	0x7DA7
Access rights, read - write:	USER - ROOT
Data type:	UINT8
Parameter value, min. - max.:	0 - MAX_UINT8

For further information on the info and error messages, see [Info and error messages](#) [► 85].

4.4.11.9 Error 8

The parameter shows the error n-8.

Code:	0x7DA8
Access rights, read - write:	USER - ROOT
Data type:	UINT8
Parameter value, min. - max.:	0 - MAX_UINT8

For further information on the info and error messages, see [Info and error messages](#) [► 85].

4.4.11.10 Error 9

The parameter shows the error n-9.

Code:	0x7DA9
Access rights, read - write:	USER - ROOT
Data type:	UINT8
Parameter value, min. - max.:	0 - MAX_UINT8

For further information on the info and error messages, see [Info and error messages](#) [► 85].

4.4.11.11 Error 10

The parameter shows the error n-10.

Code:	0x7DAA
Access rights, read - write:	USER - ROOT
Data type:	UINT8
Parameter value, min. - max.:	0 - MAX_UINT8

For further information on the info and error messages, see [Info and error messages](#) [► 85].

4.4.11.12 Error 11

The parameter shows the error n-11.

Code:	0x7DAB
Access rights, read - write:	USER - ROOT
Data type:	UINT8
Parameter value, min. - max.:	0 - MAX_UINT8

For further information on the info and error messages, see [Info and error messages](#) [► 85].

4.4.11.13 Error 12

The parameter shows the error n-12.

Code:	0x7DAC
Access rights, read - write:	USER - ROOT
Data type:	UINT8
Parameter value, min. - max.:	0 - MAX_UINT8

For further information on the info and error messages, see [Info and error messages](#) [► 85].

4.4.11.14 Error 13

The parameter shows the error n-13.

Code:	0x7DAD
Access rights, read - write:	USER - ROOT
Data type:	UINT8
Parameter value, min. - max.:	0 - MAX_UINT8

For further information on the info and error messages, see [Info and error messages](#) [► 85].

4.4.11.15 Error 14

The parameter shows the error n-14.

Code:	0x7DAE
Access rights, read - write:	USER - ROOT
Data type:	UINT8
Parameter value, min. - max.:	0 - MAX_UINT8

For further information on the info and error messages, see [Info and error messages](#) [► 85].

4.4.11.16 Error 15

The parameter shows the error n-15.

Code:	0x7DAF
Access rights, read - write:	USER - ROOT
Data type:	UINT8
Parameter value, min. - max.:	0 - MAX_UINT8

For further information on the info and error messages, see [Info and error messages](#) [► 85].

4.4.11.17 Error 16

The parameter shows the error n-16.

Code:	0x7DB0
Access rights, read - write:	USER - ROOT
Data type:	UINT8
Parameter value, min. - max.:	0 - MAX_UINT8

For further information on the info and error messages, see [Info and error messages](#) [► 85].

4.4.11.18 Error 17

The parameter shows the error n-17.

Code:	0x7DB1
Access rights, read - write:	USER - ROOT
Data type:	UINT8
Parameter value, min. - max.:	0 - MAX_UINT8

For further information on the info and error messages, see [Info and error messages](#) [► 85].

4.4.11.19 Error 18

The parameter shows the error n-18.

Code:	0x7DB2
Access rights, read - write:	USER - ROOT
Data type:	UINT8
Parameter value, min. - max.:	0 - MAX_UINT8

For further information on the info and error messages, see [Info and error messages](#) [► 85].

4.4.11.20 Error 19

The parameter shows the error n-19.

Code:	0x7DB3
Access rights, read - write:	USER - ROOT
Data type:	UINT8
Parameter value, min. - max.:	0 - MAX_UINT8

For further information on the info and error messages, see [Info and error messages](#) [► 85].

4.4.12 Ethernet

4.4.12.1 Activate web server

The parameter shows whether the web server is enabled.
A restart of the module is required after the writing.

Code:	0x7D92
Access rights, read - write:	USER - ADVANCED
Data type:	BOOL
Parameter value, min. - max.:	false - true

4.4.12.2 Activate FTP server

The parameter shows whether the FTP server is enabled.
A restart of the module is required after the writing.

Code:	0x7D93 (deactivated under PROFIBUS)
Access rights, read - write:	USER - ADVANCED
Data type:	BOOL
Parameter value, min. - max.:	false - true

4.4.12.3 MAC address

The parameter shows the MAC address.
A restart of the module is required after the writing.

Code:	0x7DE1
Access rights, read - write:	USER - DISABLED
Data type:	BINARY
Parameter value, min. - max.:	-

4.4.12.4 Host name

The parameter shows the host name.

Code:	0x7DE2 (deactivated under PROFIBUS)
Access rights, read - write:	USER - ADVANCED
Data type:	(CHAR_ARRAY) -
Parameter value, min. - max.:	-

4.4.12.5 IP address

The parameter shows the IP address.

Code:	0x7DE3 (deactivated under PROFIBUS)
Access rights, read - write:	USER - ADVANCED
Data type:	UINT32
Parameter value, min. - max.:	0 - MAX_UINT32

4.4.12.6 Subnet mask

The parameter shows the subnet mask.

Code:	0x7DE4 (deactivated under PROFIBUS)
Access rights, read - write:	USER - ADVANCED
Data type:	UINT32
Parameter value, min. - max.:	0 - MAX_UINT32

4.4.12.7 Gateway

The parameter shows the gateway.

Code:	0x7DE5 (deactivated under PROFIBUS)
Access rights, read - write:	USER - ADVANCED
Data type:	UINT32
Parameter value, min. - max.:	0 - MAX_UINT32

4.4.12.8 Preferred DNS server

The parameter shows the preferred DNS server.

Code:	0x7DE6 (deactivated under PROFIBUS)
Access rights, read - write:	USER - ADVANCED
Data type:	UINT32
Parameter value, min. - max.:	0 - MAX_UINT32

4.4.12.9 Alternative DNS server

The parameter shows the alternative DNS server.

Code:	0x7DE7 (deactivated under PROFIBUS)
Access rights, read - write:	USER - ADVANCED
Data type:	UINT32
Parameter value, min. - max.:	0 - MAX_UINT32

4.4.12.10 Activate DHCP

The parameter shows whether the DHCP is activated.

Code:	0x7DE8 (only PROFINET)
Access rights, read - write:	USER - ADVANCED
Data type:	BOOL
Parameter value, min. - max.:	false - true

4.4.12.11 Domain name

The parameter shows the domain names.

Code:	0x7DE9 (deactivated under PROFIBUS)
Access rights, read - write:	USER - ADVANCED
Data type:	(CHAR_ARRAY) -
Parameter value, min. - max.:	-

4.4.12.12 SMTP server

The parameter shows the SMTP server.

Code:	0x7DEA (deactivated under PROFIBUS)
Access rights, read - write:	USER - ADVANCED
Data type:	(CHAR_ARRAY) -
Parameter value, min. - max.:	-

4.4.13 Asynchronous**4.4.13.1 Current main board temperature**

The parameter shows the current temperature of the main board.

Code:	0x7DB9
Access rights, read - write:	USER - DISABLED
Data type:	FLOAT
Parameter value, min. - max.:	MIN_FLOAT - MAX_FLOAT

4.4.13.2 Current motor temperature

The parameter shows the current temperature of the motor.

Code:	0x7DBA
Access rights, read - write:	USER - DISABLED
Data type:	FLOAT
Parameter value, min. - max.:	MIN_FLOAT - MAX_FLOAT

4.4.13.3 Current OPT temperature Comm.

The parameter shows the current temperature of the communication board.

Code:	0x7DBB
Access rights, read - write:	USER - DISABLED
Data type:	FLOAT
Parameter value, min. - max.:	MIN_FLOAT - MAX_FLOAT

4.4.13.4 Error line

The parameter shows the last error triggered.

Code:	0x7DBC
Access rights, read - write:	USER - DISABLED
Data type:	UINT16
Parameter value, min. - max.:	0 - MAX_UINT16

NOTE

Important information for service purposes.

4.4.13.5 Error value

The parameter shows the detail that led to the last error.

Code:	0x7DBD
Access rights, read - write:	USER - DISABLED
Data type:	FLOAT
Parameter value, min. - max.:	MIN_FLOAT - MAX_FLOAT

NOTE

Important information for service purposes.

4.4.13.6 Error file

The parameter shows the name of the file in which the last triggered error was saved.

Code:	0x7DBE
Access rights, read - write:	USER - DISABLED
Data type:	(CHAR_ARRAY) -
Parameter value, min. - max.:	-

NOTE

Important information for service purposes.

4.4.13.7 Firmware type

The parameter shows the firmware type of the module.

The parameter value can only be written if the motor is de-energized.

A restart of the module is required after the writing.

Code: **0x7DBF**
 Access rights, read - write: USER - DISABLED
 Data type: ENUM
 Parameter value, min. - max.: see the following table

Parameter value	Designation
0	PTA

4.4.13.8 Order number

The parameter shows the order number of the module.

The parameter value can only be written if the motor is de-energized.

A restart of the module is required after the writing.

Code: **0x7DC0**
 Access rights, read - write: USER - SCHUNK
 Data type: UINT32
 Parameter value, min. - max.: 0 - MAX_UINT32

NOTE

Important information for service purposes.

4.4.13.9 Module name

The parameter shows the name of the module.

The parameter value can only be written if the motor is de-energized.

A restart of the module is required after the writing.

Code: **0x7DC1**
 Access rights, read - write: USER - SCHUNK
 Data type: (CHAR_ARRAY) -
 Parameter value, min. - max.: -

NOTE

Important information for service purposes.

4.4.13.10 Date firmware on the main board

The parameter shows the compile date of the main board firmware.

Code:	0x7DC2
Access rights, read - write:	USER - DISABLED
Data type:	UINT32
Parameter value, min. - max.:	0 - MAX_UINT32

NOTE

Important information for service purposes.

4.4.13.11 Time firmware on the main board

The parameter shows the compile time of the main board firmware.

Code:	0x7DC3
Access rights, read - write:	USER - DISABLED
Data type:	UINT32
Parameter value, min. - max.:	0 - MAX_UINT32

NOTE

Important information for service purposes.

4.4.13.12 Hardware version on the main board

The parameter shows the hardware version of the main board.

Code: **0x7DC4**
 Access rights, read - write: USER - DISABLED
 Data type: ENUM
 Parameter value, min. - max.: see the following table

Parameter value	Designation
530	PTA 5.3
531	PTA 5.3 with FRAM
540	PTA 5.4
541	PTA 5.4 with FRAM
611.	PTA 6.1
613.	PTA 6.1 with IO Expander
615.	PTA 6.1 with ARM Co-processor
621	PTA 6.2
623	PTA 6.2 with IO Expander
625	PTA 6.2 with ARM Co-processor
631	PTA 6.3
633	PTA 6.3 with IO Expander
635	PTA 6.3 with ARM Co-processor
801	ERB-ECU-1 for ERB 130
802	ERB-ECU-1 for ERB 150
803	ERB-ECU-1 for ERB 170
991	Unknown PTA
993	Unknown PTA with IO Expander
995	Unknown PTA with ARM Co-processor

NOTE

Important information for service purposes.

4.4.13.13 Firmware version on the main board

The parameter shows the firmware version of the main board.

Code:	0x7DC5
Access rights, read - write:	USER - DISABLED
Data type:	UINT16
Parameter value, min. - max.:	0 - MAX_UINT16

NOTE

Important information for service purposes.

4.4.13.14 Date OPT firmware. Comm.

The parameter shows the compile date of the communication board firmware.

Code:	0x7DC6
Access rights, read - write:	USER - SCHUNK
Data type:	UINT32
Parameter value, min. - max.:	0 - MAX_UINT32

NOTE

Important information for service purposes.

4.4.13.15 Time OPT firmware. Comm.

The parameter shows the compile time of the communication board firmware.

Code:	0x7DC7
Access rights, read - write:	USER - DISABLED
Data type:	UINT32
Parameter value, min. - max.:	0 - MAX_UINT32

NOTE

Important information for service purposes.

4.4.13.16 Hardware OPT version Comm.

The parameter shows the hardware version of the communication board.

Code: **0x7DC8**
 Access rights, read - write: USER - DISABLED
 Data type: ENUM
 Parameter value, min. - max.: see the following table

Parameter value	Designation	Parameter value	Designation
100	V6R PROFIBUS 1.1	130	ECM CAN-Bus 1.1
101	V6R PROFIBUS 1.2	131	ECM CAN-Bus 1.2
102	V6R PROFIBUS 1.3	132	ECM CAN-Bus 1.3
103	V6R PROFIBUS 1.4	132	ECM CAN-Bus 1.4
104	V6R PROFIBUS 1.5	133	ECM CAN-Bus 1.5
105	V6R PROFIBUS 1.6	134	ECM CAN-Bus 1.6
110	V6R CAN-Bus 1.1	135	V6R PROFINET 1.1
111	V6R CAN-Bus 1.2	140	V6R PROFINET 1.2
112	V6R CAN-Bus 1.3	142	V6R PROFINET 1.3
113	V6R CAN-Bus 1.4	143	V6R PROFINET 1.4
114	V6R CAN-Bus 1.5	144	V6R PROFINET 1.5
115	V6R CAN-Bus 1.6	145	V6R PROFINET 1.6
120	ECM PROFIBUS 1.1	150	ECM PROFINET 1.1
121	ECM PROFIBUS 1.2	151	ECM PROFINET 1.2
122	ECM PROFIBUS 1.3	152	ECM PROFINET 1.3
123	ECM PROFIBUS 1.4	153	ECM PROFINET 1.4
124	ECM PROFIBUS 1.5	154	ECM PROFINET 1.5
125	ECM PROFIBUS 1.6	155	ECM PROFINET 1.6
		255	Unknown

NOTE

Important information for service purposes.

4.4.13.17 Firmware OPT version Comm.

Access rights, read - write:

The parameter shows the firmware version of the communication board.

Code:	0x7DC9
Access rights, read - write:	USER - DISABLED
Data type:	UINT16
Parameter value, min. - max.:	0 - MAX_UINT16

NOTE

Important information for service purposes.

4.4.13.18 OPT serial number Comm.

The parameter shows the serial number of the communication board.

Code:	0x7DCA
Access rights, read - write:	USER - SCHUNK
Data type:	UINT32
Parameter value, min. - max.:	0 - MAX_UINT32

NOTE

Important information for service purposes.

4.4.13.19 Date Firmware OPT 1

The parameter shows the compile date of the expansion board 1 firmware.

Code:	0x7DCB
Access rights, read - write:	USER - DISABLED
Data type:	UINT32
Parameter value, min. - max.:	0 - MAX_UINT32

NOTE

Important information for service purposes.

4.4.13.20 Time firmware OPT 1

The parameter shows the compile time of the expansion board 1 firmware.

Code:	0x7DCC
Access rights, read - write:	USER - DISABLED
Data type:	UINT32
Parameter value, min. - max.:	0 - MAX_UINT32

NOTE

Important information for service purposes.

4.4.13.21 Hardware OPT 1 version

The parameter shows the hardware version of the expansion board 1.

Code:	0x7DCD
Access rights, read - write:	USER - DISABLED
Data type:	UINT16
Parameter value, min. - max.:	0 - MAX_UINT16

NOTE

Important information for service purposes.

4.4.13.22 Firmware OPT 1 version

The parameter shows the firmware version of the expansion board 1.

Code:	0x7DCE
Access rights, read - write:	USER - DISABLED
Data type:	UINT16
Parameter value, min. - max.:	0 - MAX_UINT16

NOTE

Important information for service purposes.

4.4.13.23 OPT 1 serial number

The parameter shows the serial number of the expansion board 1.

Code:	0x7DCF
Access rights, read - write:	USER - SCHUNK
Data type:	UINT32
Parameter value, min. - max.:	0 - MAX_UINT32

NOTE

Important information for service purposes.

4.4.13.24 Date firmware OPT 2

The parameter shows the compile date of the expansion board 2 firmware.

Code:	0x7DD0
Access rights, read - write:	USER - DISABLED
Data type:	UINT32
Parameter value, min. - max.:	0 - MAX_UINT32

NOTE

Important information for service purposes.

4.4.13.25 Time firmware OPT 2

The parameter shows the compile time of the expansion board 2 firmware.

Code:	0x7DD1
Access rights, read - write:	USER - DISABLED
Data type:	UINT32
Parameter value, min. - max.:	0 - MAX_UINT32

NOTE

Important information for service purposes.

4.4.13.26 Hardware OPT 2 version

The parameter shows the hardware version of the expansion board 2.

Code:	0x7DD2
Access rights, read - write:	USER - DISABLED
Data type:	UINT16
Parameter value, min. - max.:	0 - MAX_UINT16

NOTE

Important information for service purposes.

4.4.13.27 Firmware OPT 2 version

The parameter shows the firmware version of the expansion board 2.

Code:	0x7DD3
Access rights, read - write:	USER - DISABLED
Data type:	UINT16
Parameter value, min. - max.:	0 - MAX_UINT16

NOTE

Important information for service purposes.

4.4.13.28 OPT 2 serial number

The parameter shows the serial number of the expansion board 2.

Code:	0x7DD4
Access rights, read - write:	USER - SCHUNK
Data type:	UINT32
Parameter value, min. - max.:	0 - MAX_UINT32

NOTE

Important information for service purposes.

4.4.13.29 Protocol version

The parameter shows the version of the protocol used.

Code:	0x7DD5
Access rights, read - write:	USER - DISABLED
Data type:	UINT16
Parameter value, min. - max.:	0 - MAX_UINT16

4.4.13.30 Motor voltage

The parameter shows the current motor voltage [V].

Code:	0x7DD6
Access rights, read - write:	USER - DISABLED
Data type:	FLOAT
Parameter value, min. - max.:	0 - MAX_FLOAT

4.4.13.31 Logic voltage

The parameter shows the current logic voltage [V].

Code:	0x7DD7
Access rights, read - write:	USER - DISABLED
Data type:	FLOAT
Parameter value, min. - max.:	0 - MAX_FLOAT

4.4.13.32 Max. software end stop (temporary)

The parameter shows the maximum value of the software stop.

Code:	0x7DD8
Access rights, read - write:	PROFI - PROFi
Data type:	FLOAT/INT32
Parameter value, min. - max.:	MIN_FLOAT - MAX_FLOAT

If the value is changed, this is only done temporarily without storing the change in the EEPROM.

4.4.13.33 Min. software end stop (temporary)

The parameter shows the minimum value of the software stop.

Code:	0x7DD9
Access rights, read - write:	PROFI - PROFi
Data type:	FLOAT/INT32
Parameter value, min. - max.:	MIN_FLOAT - MAX_FLOAT

If the value is changed, this is only done temporarily without storing the change in the EEPROM.

4.4.13.34 User

The parameter shows the current user, [User administration](#) [► 7].

Code: **0x7DDA**

Access rights, read - write: USER -

Data type: ENUM while reading
(CHAR_ARRAY) - while writing

Parameter value, min. - max.: see the following table

Parameter value	Designation
0	User
1	Diag
2	Profi
3	Advanced
4	Root
5	SCHUNK

The current user is reported in the case of read-only access.

5 Info and error messages

5.1 Detailed error information

Information about the last error that occurred can be queried using the following parameters:

- Error Detail
- [Error value](#) [► 74]
- Error Line
- [Error line](#) [► 74]
- Error File
- [Error file](#) [► 74]

5.2 Info codes

5.2.1 INFO NO RIGHTS

The appropriate rights to execute the command are missing.

Code: **0x03**

5.2.2 INFO UNKNOWN COMMAND

The sent command is not recognized.

Code: **0x04**

5.2.3 INFO FAILED

The command has failed.

Code: **0x05**

All of the parameters are correct, but the execution of the command is not possible at this time due to other reasons, e. g. the module is in emergency stop mode.

5.2.4 NOT REFERENCED

The module is not referenced and can therefore not execute the command.

Code: **0x06**

A referencing process is necessary in order to carry out a positioning movement.

5.2.5 INFO COMMUNICATION ERROR

An error has occurred in communication.

Code: **0x09**

The connection of the communication cable and external influences on the communication cable must be checked.

5.2.6 INFO TIMEOUT

A timeout occurred during communication.

Code: **0x10**

The data could not be sent, or more data was anticipated and was not received on time.

5.2.7 INFO WRONG DATA TYPE

The datatype does not match the parameter.

Code: **0x12**

5.2.8 INFO RESTART

The module or control unit was restarted.

The module is not ready for operation. The message must be acknowledged; for acknowledgment the bits "Stop" and "Fast stop" must be set to "1", [Sample travel commands](#) [► 95].

Code: **0x13**

5.2.9 INFO CHECKSUM

The checksum is incorrect, the data is invalid.

Code: **0x19**

5.2.10 INFO VALUE LIMIT MAX

The specified value exceeds the maximum permitted set value.

Code: **0x1B**

5.2.11 INFO VALUE LIMIT MIN

The specified value falls below the minimum permitted set value.

Code: **0x1C**

5.2.12 INFO MESSAGE LENGTH

The length command does not match the data received.

Code: **0x1D**

5.2.13 INFO WRONG PARAMETER

One of the specified parameter values is outside of the permitted range.

Code: **0x1E**

If a parameter value is found to be not permitted, all old parameter values will be retained, even if the other parameter values should lie in the valid range.

5.2.14 INFO UNKNOWN PARAMETER

The parameter requested is unknown.

Code: **0x23**

5.3 Error codes

5.3.1 ERROR FILE NOT FOUND

The file to be edited is not on the USB stick or the USB stick is defective.

Code: **0x60**

5.3.2 ERROR FILE IS CORRUPT

The file to be edited on the USB stick is corrupted.

Code: **0x61**

5.3.3 ERROR FILE TYPE WRONG

The data type is not correct.

Code: **0x62**

- Create a new file.

5.3.4 ERROR FILE SYSTEM WRONG

The file system from the USB stick has an error.

Code: **0x64**

- Check whether the USB drive is formatted with FAT16 or FAT32.

5.3.5 ERROR FILE READ

A reading error has occurred during the reading of the file.

Code: **0x65**

5.3.6 ERROR FILE IS NOT CREATED

No file could be created.

Code: **0x66**

- Check whether the USB stick is defective or write-protected.

5.3.7 ERROR FILE WRITE

A writing error has occurred during the writing of the file.

Code: **0x67**

5.3.8 ERROR REBOOT

A parameter was written which is needed by a reboot.

Code: **0x7C**

- Switch the module off and on.

5.3.9 ERROR MOTOR PHASE

One motor phase is not properly connected.

Code: **0x7D**

5.3.10 ERROR WRONG RAMP TYPE

No valid motion profile has been selected for the positioning movement.

Code: **0xC8**

5.3.11 ERROR WRONG DIRECTION

The module moves in the wrong direction for the check of the pseudo-absolute encoder

Code: **0xD1**

- Check the sine pointer.

5.3.12 ERROR CONFIG MEMORY

The configuration range is incorrect. Data could not be written to EEPROM or EEPROM is defective.

Code: **0xD2**

5.3.13 ERROR SOFT LOW

The module has exceeded the lower software limit range.

Code: **0xD5**

- If necessary, acknowledge the error and move the module out of the software end stop with a movement command.

5.3.14 ERROR SOFT HIGH

The module has exceeded the upper software limit range.

Code: **0xD6**

- If necessary, acknowledge the error and move the module out of the software end stop with a movement command.

5.3.15 ERROR SERVICE

An error has occurred that can only be remedied by SCHUNK.

Code: **0xD8**

The detailed error information can be used by SCHUNK Service to localize the error precisely, [Detailed error information](#) [► 85].

Contact SCHUNK's service and provide the following data:

- Module type
- Serial number of the module
- Description of how the error occurred

5.3.16 ERROR FAST STOP

A fast stop was triggered, [Incoming process data](#) [► 12].

The module is not ready for operation. No error condition was triggered. The message must be acknowledged.

Code: **0xD9**

5.3.17 ERROR TOW

A towing error has occurred.

Code: **0xDA**

- Reduce the load.
- Check the "towing error" parameter.

5.3.18 ERROR VPC3

The controller works incorrectly or is faulty.

Code: **0xDB**

5.3.19 ERROR FRAGMENTATION

An error has occurred in the fragmentation protocol. Data packets have been lost.

Code: **0xDC**

5.3.20 ERROR COMMUTATION

The motor cannot commute.

Code: **0xDD**

If this error occurs, the commutation mode is selected incorrectly. In case of block commutation the hall sensors are defective or not connected. With regard to sine commutation, there is an error in the position measuring system.

5.3.21 ERROR I2T

An I²T error has occurred.

Code: **0xDF**

- Reduce load of motor.

5.3.22 ERROR CURRENT

The maximum current was exceeded.

Code: **0xDE**

- Reduce load of motor.

5.3.23 ERROR TOO FAST

The maximum speed was exceeded.

Code: **0xE4**

5.3.24 ERROR POS SYSTEM

The position measuring system is not working correctly.

Code: **0xE5**

- Check configuration of the module.

5.3.25 ERROR RESOLVER CHECK FAILED

A parameter for the resolver setting is incorrect.

Code: **0xEB**

5.3.26 ERROR MATH

A mathematical error has occurred, e. g. division by zero.

Code: **0xEC**

Usually, a configuration parameter is incorrect, resulting in the value range being exceeded. In most cases, a controller parameter is set incorrectly.

The detailed error information can be used by SCHUNK Service to localize the error precisely, [Detailed error information](#) [► 85].

5.3.27 ERROR CALIB CURRENT

The measured values of the current sensors are beyond the tolerance limits.

Code: **0xEE**

- Calibrate the module.
 - If the error occurs again, the current measuring is defective.

5.3.28 ERROR INITIALIZE

The module could not be properly initialized.

Code: **0xE0**

- Check configuration parameters.

The detailed error information can be used by SCHUNK Service to localize the error precisely, [Detailed error information](#) [► 85].

5.3.29 ERROR INTERNAL

An internal error has occurred.

Code: **0xE1**

The firmware is in an undefined status.

Contact SCHUNK's service and provide the following data:

- Module type
- Serial number of the module
- Description of how the error occurred

5.3.30 ERROR CONNECTION TEMP LOW

The temperature of the communication board dropped below the minimal permissible level.

Code: **0x6A**

- Warm up the module.

5.3.31 ERROR CONNECTION TEMP HIGH

The maximum permissible temperature of the communication board was exceeded.

Code: **0x6B**

- Allow the module to cool down.
- Reduce the load.

5.3.32 ERROR MOTOR TEMP LOW

The temperature of the motor dropped below the minimal permissible level.

Code: **0x6C**

- Warm up the module.

5.3.33 ERROR MOTOR TEMP HIGH

Code:: 0x6D

The temperature of the motor has exceeded the maximum permissible level.

Code: **0x6D**

- Allow the module to cool down.
- Reduce the load.

5.3.34 ERROR TEMP LOW OPTION

The temperature of the option board has fallen below the permissible temperature range.

Code: **0x6E**

- Warm up the module.

5.3.35 ERROR TEMP HIGH OPTION

The temperature of the option board has exceeded the permissible temperature range.

Code: **0x6F**

- Allow the module to cool down.
- Reduce the load.

5.3.36 ERROR TEMP LOW

The temperature value for the main board dropped below the minimal permissible temperature range.

Code: **0x70**

- Warm up the module.

5.3.37 ERROR TEMP HIGH

The temperature value for the main board exceeded the maximum permissible temperature range.

Code: **0x71**

- Allow the module to cool down.
- Reduce the load.

5.3.38 ERROR LOGIC LOW

The logic voltage is below the limit values.

Code: **0x72**

- Check the logic voltage.

5.3.39 ERROR LOGIC HIGH

The logic voltage is above the limit values.

Code: **0x73**

- Check the logic voltage.

5.3.40 ERROR MOTOR VOLTAGE LOW

The motor voltage is under the limit values.

Code: **0x74**

- Check the motor voltage.
 - If necessary, the power supply unit for the motor voltage might be underdimensioned or the voltage supply cables to the module are not dimensioned correctly.

NOTE

MotorVoltageLow is a fatal error when the module is moved.

5.3.41 ERROR MOTOR VOLTAGE HIGH

The motor voltage is above the limit values.

Code: **0x75**

NOTE

If this error occurs repeatedly, the module is disabled and can only be put into operation again by SCHUNK.

- Check the motor voltage.
 - If necessary, an external brake chopper might have to be used.

5.3.42 ERROR CABLE BREAK

Communication to the control was interrupted.

Code: **0x76**

NOTE

The error is only displayed once communication has been re-established.

- Check communication cables
 - The communication cable is defective.

5.3.43 ERROR LIFE SIGN

Timeout of the internal module communication due to an internal error.

Code: **0x7A**

- Module must be restarted.

5.3.44 ERROR CUSTOM DEFINED

An error has occurred in a customer-defined function.

Code: **0x7B**

5.3.45 ERROR OVERSHOOT

The module has overshoot the target position.

Code: **0x82**

- Increase the specified current.
 - The current needed for deceleration is too low.
- Check the controller parameters.

5.3.46 ERROR HARDWARE VERSION

The hardware of the different components do not match. One of the files saved on the USB stick can not be edited with the available hardware.

Code: **0x83**

5.3.47 ERROR SOFTWARE VERSION

The software of the different components does not match. One of the files saved on the USB stick can not be edited with the available software version.

Code: **0x84**

6 Appendix

6.1 Sample travel commands

In the following, the sequence from system start to the first gripping cycle is shown as an example.

The gripping task refers to a workpiece with a dimension of 35 mm and flat, parallel gripping surfaces. The surface should be protected when gripping. It is stored in a container with limited space.

Sequence

1. Ensure that the machine is in a ready-to-operate condition
 - Acknowledging
 - Referencing
2. Gripping task
 - Approach the pre-position (optimizes cycle time)
 - gripping without damaging the surface (can be omitted)
 - Build up gripping force
 - placement in confined space

6.1.1 Acknowledge error

































Incoming process data Control word

Byte	Bit	Designation	PROFINET command		PROFIBUS command	
0	0	Command release A		1		1
0	1	Command release B		0		0
0	2	Jerk activated		0		0
0	3	Acceleration active		0		0
0	4	Current active		0		0
0	5	Speed active		0		0
0	6	Position active		0		0
0	7	Relative position active		0		0
1	8	Reserved		0		0
1	9	Referencing		0		0
1	10	Res_A_01_2		0		0
1	11	Res_A_01_3		0		0
1	12	Restart module		0		0
1	13	Acknowledge error		1		1
1	14	Stop		1		1
1	15	Quick stop		1		1

Target values

Byte	Designation	Value
4-7	Target position	-
8-11	Target speed	-
12-15	Target current	-
16-19	Target acceleration	-
20-23	Target jerk	-

Outgoing process data Status word

































Byte	Bit	Designation	previous		Response	
0	0	Command release A		0		1
0	1	Command release B		0		0
0	2	Reserved		0		0
0	3	Reserved		0		0
0	4	Reserved		0		0
0	5	Reserved		0		0
0	6	Module ready for operation		0		1
0	7	Reserved		0		0
1	8	Module moves		0		0
1	9	Target position has been reached		0		0
1	10	Movement has been blocked		0		0
1	11	Break applied		1		1
1	12	Module is referenced		0		0
1	13	Info message		0		0
1	14	Warning		1		0
1	15	Fatal error		1		0

Actual values

Byte	Designation	Value
4-7	Actual position	-
8-11	Actual speed	-
12-15	Actual current	-

6.1.2 Referencing

Incoming process data Control word

































Byte	Bit	Designation	PROFINET command		PROFIBUS command	
0	0	Command release A		0		0
0	1	Command release B		1		1
0	2	Jerk activated		0		0
0	3	Acceleration active		0		0
0	4	Current active		0		0
0	5	Speed active		0		0
0	6	Position active		0		0
0	7	Relative position active		0		0
1	8	Reserved		0		0
1	9	Referencing *		1		1
1	10	Res_A_01_2		0		0
1	11	Res_A_01_3		0		0
1	12	Restart module		0		0
1	13	Acknowledge error		0		0
1	14	Stop		1		1
1	15	Quick stop		1		1

* The bit must remain set until the referencing run is complete.

Target values

Byte	Designation	Value
4-7	Target position	-
8-11	Target speed	100
12-15	Target current	1
16-19	Target acceleration	2500
20-23	Target jerk	50000

Outgoing process data Status word

































Byte	Bit	Designation	Response Start		Response completed	
0	0	Command release A		0		0
0	1	Command release B		1		1
0	2	Reserved		0		0
0	3	Reserved		0		0
0	4	Reserved		0		0
0	5	Reserved		0		0
0	6	Module ready for operation		1		1
0	7	Reserved		0		0
1	8	Module moves		1		0
1	9	Target position has been reached		0		0
1	10	Movement has been blocked		0		0
1	11	Break applied		0		1
1	12	Module is referenced		0		0
1	13	Info message		0		0
1	14	Warning		0		0
1	15	Fatal error		0		0

Actual values

Byte	Designation	Value on Start reply	Value on Complete reply
4-7	Actual position	~	0
8-11	Actual speed	100	0
12-15	Actual current	~	0

6.1.3 Prepositioning

































































Incoming process data Control word

Byte	Bit	Designation	PROFINET command		PROFIBUS command	
0	0	Command release A		1		1
0	1	Command release B		0		1
0	2	Jerk activated		0		1
0	3	Acceleration active		0		1
0	4	Current active		0		1
0	5	Speed active		0		1
0	6	Position active		1		1
0	7	Relative position active		0		0
1	8	Reserved		0		0
1	9	Referencing		0		0
1	10	Res_A_01_2		0		0
1	11	Res_A_01_3		0		0
1	12	Restart module		0		0
1	13	Acknowledge error		0		0
1	14	Stop		1		1
1	15	Quick stop		1		1

Target values

Byte	Designation	Value
4-7	Target position	37
8-11	Target speed	150
12-15	Target current	2
16-19	Target acceleration	2500
20-23	Target jerk	50000

Outgoing process data Status word

































Byte	Bit	Designation	Response PROFINET Start		Response PROFINET completed		Response PROFIBUS Start		Response PROFIBUS completed	
0	0	Command release A		1		1		1		1
0	1	Command release B		0		0		0		0
0	2	Reserved		0		0		0		0
0	3	Reserved		0		0		0		0
0	4	Reserved		0		0		0		0
0	5	Reserved		0		0		0		0
0	6	Module ready for operation		1		1		1		1
0	7	Reserved		0		0		0		0
1	8	Module moves		1		0		1		0
1	9	Target position has been reached		0		1		0		1
1	10	Movement has been blocked		0		0		0		0
1	11	Break applied		0		1		0		1
1	12	Module is referenced		1		1		1		1
1	13	Info message		0		0		0		0
1	14	Warning		0		0		0		0
1	15	Fatal error		0		0		0		0

Actual values

Byte	Designation	Response PROFINET Start	Response PROFINET completed	Response PROFIBUS Start	Response PROFIBUS completed
4-7	Actual position	~	37	~	37
8-11	Actual speed	150	0	150	0
12-15	Actual current	~	0	~	0

6.1.4 Grip

































































Incoming process data Control word

Byte	Bit	Designation	PROFINET command		PROFIBUS command	
0	0	Command release A		0		0
0	1	Command release B		1		1
0	2	Jerk activated		0		1
0	3	Acceleration active		0		1
0	4	Current active		0		1
0	5	Speed active		1		1
0	6	Position active		0		0
0	7	Relative position active		0		0
1	8	Reserved		0		0
1	9	Referencing		0		0
1	10	Res_A_01_2		0		0
1	11	Res_A_01_3		0		0
1	12	Restart module		0		0
1	13	Acknowledge error		0		0
1	14	Stop		1		1
1	15	Quick stop		1		1

Target values

Byte	Designation	Value
4-7	Target position	-
8-11	Target speed	-15
12-15	Target current	2
16-19	Target acceleration	2500
20-23	Target jerk	50000

Outgoing process data Status word

Byte	Bit	Designation	Response PROFINET		Response PROFINET		Response PROFIBUS		Response PROFIBUS	
			Start		completed		Start		completed	
0	0	Command release A		0		0		0		0
0	1	Command release B		1		1		1		1
0	2	Reserved		0		0		0		0
0	3	Reserved		0		0		0		0
0	4	Reserved		0		0		0		0
0	5	Reserved		0		0		0		0
0	6	Module ready for operation		1		1		1		1
0	7	Reserved		0		0		0		0
1	8	Module moves		1		0		1		0
1	9	Target position has been reached		0		0		0		0
1	10	Movement has been blocked		0		1		0		1
1	11	Break applied		0		0		0		0
1	12	Module is referenced		1		1		1		1
1	13	Info message		0		0		0		0
1	14	Warning		0		0		0		0
1	15	Fatal error		0		0		0		0

Actual values

Byte	Designation	Response PROFINET	Response PROFINET	Response PROFIBUS	Response PROFIBUS
		Start	completed	Start	completed
4-7	Actual position	~	35	~	35
8-11	Actual speed	-15	0	-15	0
12-15	Actual current	~	2	~	2

6.1.5 Gripping force

































































Incoming process data Control word

Byte	Bit	Designation	PROFINET command		PROFIBUS command	
0	0	Command release A		1		1
0	1	Command release B		0		0
0	2	Jerk activated		0		1
0	3	Acceleration active		0		1
0	4	Current active		0		1
0	5	Speed active		1		1
0	6	Position active		0		0
0	7	Relative position active		0		0
1	8	Reserved		0		0
1	9	Referencing		0		0
1	10	Res_A_01_2		0		0
1	11	Res_A_01_3		0		0
1	12	Restart module		0		0
1	13	Acknowledge error		0		0
1	14	Stop		1		1
1	15	Quick stop		1		1

Target values

Byte	Designation	Value
4-7	Target position	-
8-11	Target speed	-15
12-15	Target current	4
16-19	Target acceleration	2500
20-23	Target jerk	50000

Outgoing process data Status word

































Byte	Bit	Designation	Response PROFINET Start		Response PROFINET completed		Response PROFIBUS Start		Response PROFIBUS completed	
0	0	Command release A		1		1		1		1
0	1	Command release B		0		0		0		0
0	2	Reserved		0		0		0		0
0	3	Reserved		0		0		0		0
0	4	Reserved		0		0		0		0
0	5	Reserved		0		0		0		0
0	6	Module ready for operation		1		1		1		1
0	7	Reserved		0		0		0		0
1	8	Module moves		1		0		1		0
1	9	Target position has been reached		0		0		0		0
1	10	Movement has been blocked		0		1		0		1
1	11	Break applied		0		0		0		0
1	12	Module is referenced		1		1		1		1
1	13	Info message		0		0		0		0
1	14	Warning		0		0		0		0
1	15	Fatal error		0		0		0		0

Actual values

Byte	Designation	Response PROFINET Start	Response PROFINET completed	Response PROFIBUS Start	Response PROFIBUS completed
4-7	Actual position	~	35	~	35
8-11	Actual speed	~	0	~	0
12-15	Actual current	2	4	2	4

6.1.6 Release

































































Incoming process data Control word

Byte	Bit	Designation	PROFINET command		PROFIBUS command	
0	0	Command release A		0		0
0	1	Command release B		1		1
0	2	Jerk activated		0		1
0	3	Acceleration active		0		1
0	4	Current active		0		1
0	5	Speed active		0		1
0	6	Position active		0		0
0	7	Relative position active		1		1
1	8	Reserved		0		0
1	9	Referencing		0		0
1	10	Res_A_01_2		0		0
1	11	Res_A_01_3		0		0
1	12	Restart module		0		0
1	13	Acknowledge error		0		0
1	14	Stop		1		1
1	15	Quick stop		1		1

Target values

Byte	Designation	Value
4-7	Target position	1.4
8-11	Target speed	150
12-15	Target current	1
16-19	Target acceleration	2500
20-23	Target jerk	50000

Outgoing process data Status word

Byte	Bit	Designation	Response PROFINET		Response PROFINET		Response PROFIBUS		Response PROFIBUS	
			Start		completed		Start		completed	
0	0	Command release A		0		0		0		0
0	1	Command release B		1		1		1		1
0	2	Reserved		0		0		0		0
0	3	Reserved		0		0		0		0
0	4	Reserved		0		0		0		0
0	5	Reserved		0		0		0		0
0	6	Module ready for operation		1		1		1		1
0	7	Reserved		0		0		0		0
1	8	Module moves		1		0		1		0
1	9	Target position has been reached		0		1		0		1
1	10	Movement has been blocked		0		0		0		0
1	11	Break applied		0		1		0		1
1	12	Module is referenced		1		1		1		1
1	13	Info message		0		0		0		0
1	14	Warning		0		0		0		0
1	15	Fatal error		0		0		0		0

Actual values

Byte	Designation	Response PROFINET	Response PROFINET	Response PROFIBUS	Response PROFIBUS
		Start	completed	Start	completed
4-7	Actual position	35	36.4	35	36.4
8-11	Actual speed	150	0	150	0
12-15	Actual current	~	0	~	0