Software manual

EGI with PROFINET, EtherCAT or EtherNet/IP™ V2.xx

Superior Clamping and Gripping

SCHUNK

Imprint

Technical changes:

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

Document number: 1396640

Version: 03.00 | 13/07/2020 | en

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

1.1 About this document

This software manual describes the operating and parameterization options of an **I**ntelligent **E**lectric **G**ripper with the following interfaces:

- **P**ROFI**N**ET (EGI PN)
- **E**ther**C**AT (EGI EC)
- **E**therNET/**I**P™ (EGI EI)

EGI PN, EGI EC, or EGI EI are hereinafter referred to as "module".

Trademarks

- PROFINET is a trademark of the PROFIBUS and PROFINET User Organization (PI).
- EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.
- EtherNet/IP™ is a trademark of ODVA, Inc.

Validity

This version of the software manual describes the functions of firmware versions that bear the main version number 2.XX.

The firmware version can be read out. Information on the corresponding parameter can be found in section Parameter list [▶ 42].

Conventions

The following conventions apply to this software manual:

- A user-initiated action that the module is expected to perform is hereafter referred to as a "request".
- Identification of parameters: <parameter>
- Identification of events: WARNING

The following abbreviations are used:

- GKE - Gripping force maintenance

Applicable documents

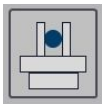
- General terms of business *
- Assembly and Operating Manual of the module *

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

1.2 Definitions and exact limits

1.2.1 Direction of movement and gripper type

The following shows the module's directions of movement, with the aim of allowing the direction of movement of the module to be interpreted clearly. The module can be used for I.D. and/or O.D. gripping.

Direction of movement	Description	Illustration
Outward movement - I.D. gripping	The base jaws move from the inside to the outside.	
	The workpiece is gripped from the inside.	
Inward movement - O.D. gripping	The base jaws move from the outside to the inside.	
	The workpiece is gripped from the outside.	

1.2.2 Hardware and software limits

Hardware limits

The hardware limits correspond to the maximum physical positions that can be approached by the module. Depending on the application, different hardware limits may apply, e.g. when using protruding fingers.

The *lower* hardware limit of the module corresponds to the application-specific physical end stop, which is reached by moving in the negative direction of movement.

The *upper* hardware limit of the module corresponds to the application-specific physical end stop, which is reached by moving in the positive direction of movement.

Software limits

The software limits correspond to the minimum and maximum positions that can be approached by the module.

The *lower* software limit corresponds to the minimum approachable position.

The *upper* software limit corresponds to the maximum approachable position.

The software limits are only monitored in the referenced module state and may be parameterized by the user, Parameter list [▶ 34].

1.2.3 Reference point and zero point

Zero point

The zero point corresponds to the absolute reference point to which the positioning system of the module relates. The zero point corresponds to the lower hardware limit of the module. In order to determine the zero point, a reference point is required.

Reference point

A reference point corresponds to a unique, reproducible position that can be approached by the module.

1.2.4 Overview of important exact limits

The following table contains the module's most important exact limits. For a detailed description of the parameters, see chapter Parameter list [▶ 34].

Value	Minimum value	Maximum value
Ambient temperature	5°C	55°C
Logic voltage supply	21.6 V	26.4 V
Power voltage supply	21.6 V	26.4 V
Positioning speed	15 mm/s	*EGI 40:* 100 mm/s *EGI 80:* 200 mm/s

2 Communication

2.1 Data exchange

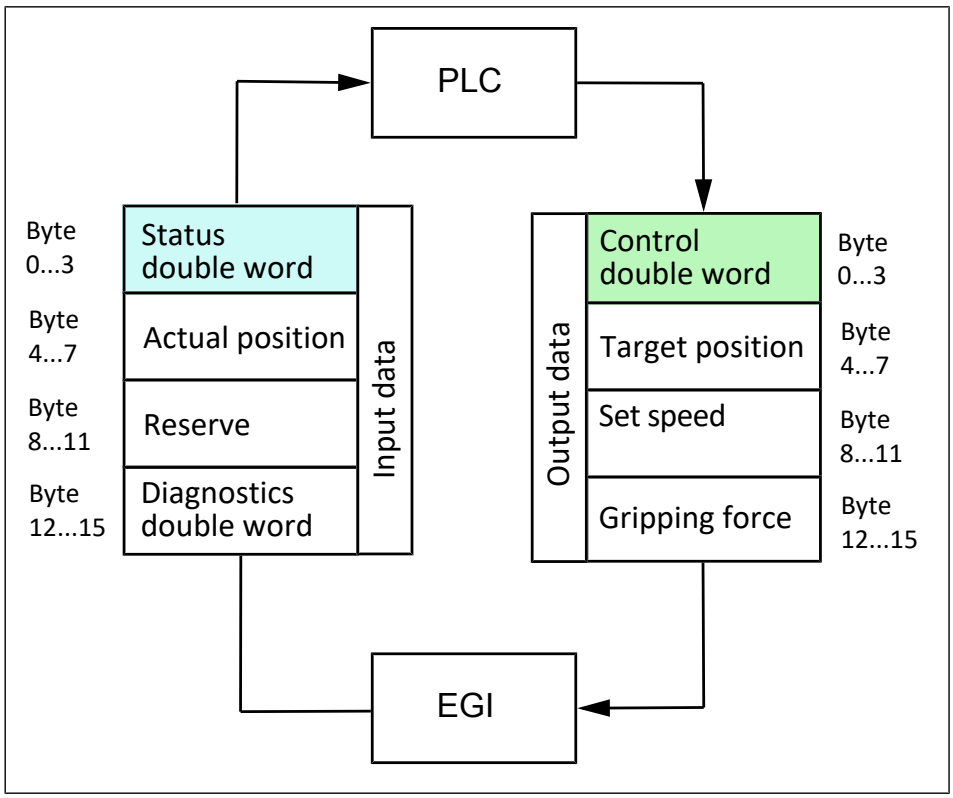
Via integrated fieldbus interfaces, data can be exchanged cyclically and acyclically between the module and the controller.

The following fieldbus interfaces are available:

- PROFINET
- EtherCAT
- EtherNet/IP™

2.1.1 Cyclical data exchange

For cyclical data exchange, a fixed data frame for input and output data is defined. The data frame is based on the use of double word data and is set to a data length of four double words.



Cyclical data exchange

For further information on data transmission and interpretation, see the following sections.

2.1.1.1 Cyclical output data

The cyclical output data is transmitted from the PLC to the module, thereby sending requests to the module. For practical application examples, see chapter Application examples [▶ 52].

Implementing requests

Requests to the module may be permissible or impermissible.

Permissible requests are executed by the module. Impermissible requests are not executed, which is indicated to the PLC by setting the status bit "not feasible", Status double word [▶ 62]-Bit 3.

Impermissible requests

Impermissible requests could be caused by the following:

- The request is temporarily impermissible, e.g. because the module is currently actively executing a movement.
- The transmitted bit combination, in particular the control double word, is impermissible.
- At least one movement parameter that has been transmitted is impermissible.

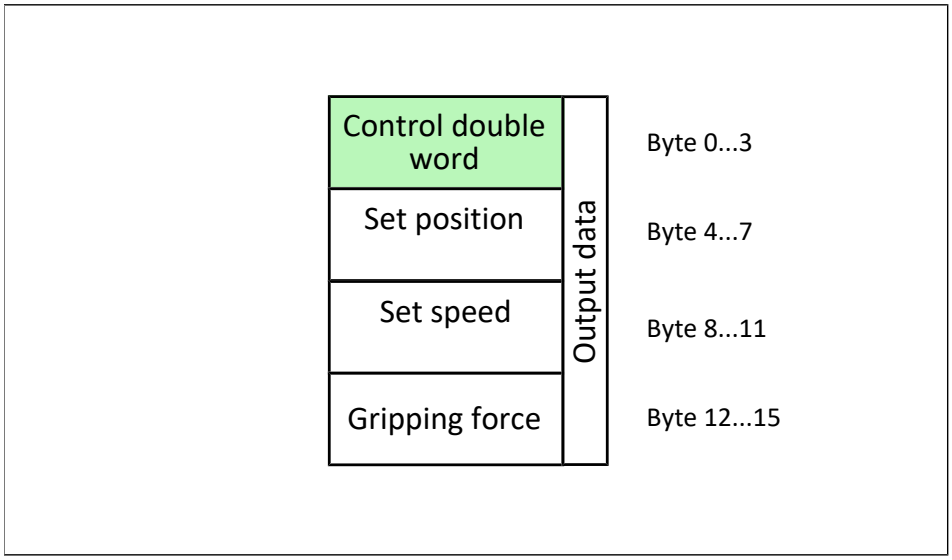
An immediate transition between active movements of the module is not permitted, and results in a controlled termination of the module's current active movement.

Depending on the currently transmitted bit combination, impermissible bit combinations of the control double word will arise. For a detailed description of impermissible bit combinations, see chapter Control double word [▶ 58].

The value of a movement parameter is considered to be impermissible if it is outside the permitted minimum or maximum limits, Definitions and exact limits [▶ 6].

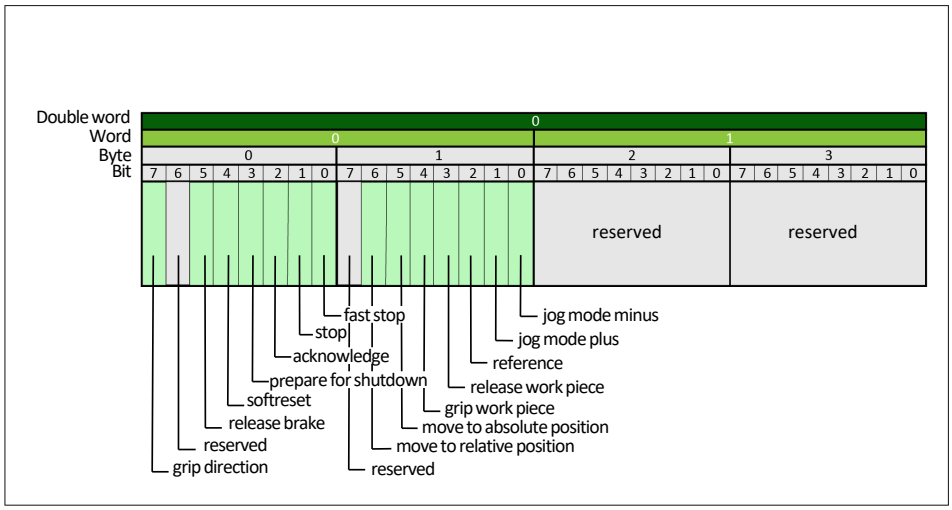
Data frame

The data frame of cyclical output data is composed of the control double word and movement parameters.



Data frame of cyclical output data

Control double word



Bit sequence control double word

In bytes 0 – 3 of the cyclical output data, the control double word is transmitted. The structure of the control double word is shown in the following table. For a detailed description of the control double word, see chapter Control double word [▶ 58].

Word	Byte	Bit	Cyclical output data	For further information, see chapter
0	0	0	fast stop	Control double word [▶ 58], Bit 0
		1	stop	Control double word [▶ 58], Bit 1
		2	acknowledge	Control double word [▶ 58], Bit 2
		3	prepare for shutdown	Control double word [▶ 59], Bit 3
		4	softreset	Control double word [▶ 59], Bit 4
		5	release brake	Control double word [▶ 59], Bit 5
		6	reserved	-
		7	grip direction	Control double word [▶ 59], Bit 7
	1	8	jog mode minus	Control double word [▶ 60], Bit 8
		9	jog mode plus	Control double word [▶ 60], Bit 9
		10	reference	Control double word [▶ 60], Bit 10
		11	release work piece	Control double word [▶ 60], Bit 11
		12	grip work piece	Control double word [▶ 61], Bit 12
		13	move to absolute position	Control double word [▶ 61], Bit 13
		14	move to relative position	Control double word [▶ 61], Bit 14
		15	reserved	-

Word	Byte	Bit	Cyclical output data	For further information, see chapter
1	2	16	reserved	-
		17	reserved	-
		18	reserved	-
		19	reserved	-
		20	reserved	-
		21	reserved	-
		22	reserved	-
		23	reserved	-
	3	24	reserved	-
		25	reserved	-
		26	reserved	-
		27	reserved	-
		28	reserved	-
		29	reserved	-
		30	reserved	-
		31	reserved	-

Position

- In bytes 4 – 7 of the cyclical output data, data is transmitted that is used for positioning purposes, Parameter list [▶ 34].
- The data format of the parameter is *signed 32 bits* and represents a value in micrometers [µm].

Speed

- In bytes 8 – 11 of the cyclical output data, the value of the set speed of a movement is transmitted, Parameter list [▶ 34].
- The data format of the parameter is *signed 32 bits* and represents a value in micrometers per second [µm/s].

Gripping force

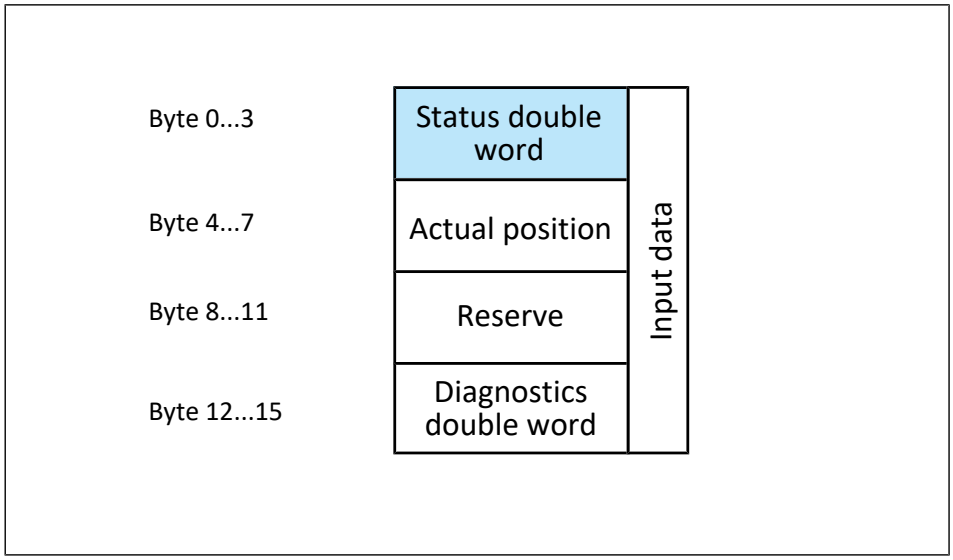
- In bytes 12 – 15 of the cyclical output data, the gripping force with which a workpiece is to be gripped is transmitted, Gripping a workpiece [▶ 28].
- The data format of the parameter is *signed 32 bits* and represents a value in millinewton [mN].

2.1.1.2 Cyclical input data

The cyclical input data is transmitted from the module to the control. This gives the PLC feedback from the module, allowing an appropriate reaction to then take place.

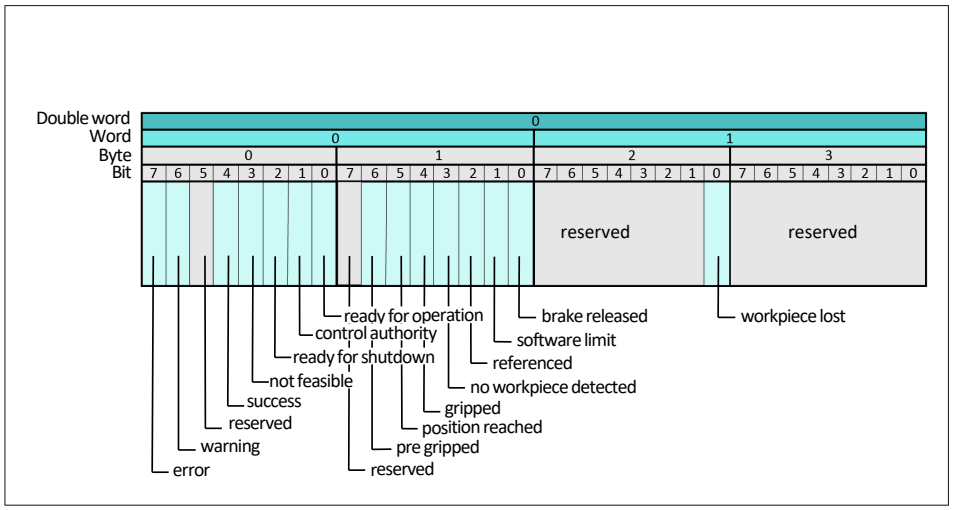
Data frame

The data frame of cyclical input data is composed of the status double word and module feedback signals.



Data frame of cyclical input data

Status double word



Bit sequence status double word

In bytes 0 – 3 of the cyclical input data, the status double word is transmitted. In the following table, the structure of the status double word is shown. For a detailed description of the status double word, see chapter Status double word [▶ 62].

Word	Byte	Bit	Cyclical input data	For further information, see chapter
0	0	0	ready for operation	Status double word [▶ 62], Bit 0
		1	bus control authority	Status double word [▶ 62], Bit 1
		2	ready for shutdown	Status double word [▶ 62], Bit 2
		3	not feasible	Status double word [▶ 62], Bit 3
		4	success	Status double word [▶ 62], Bit 4
		5	reserved	-
		6	warning	Status double word [▶ 62], Bit 6
		7	error	Status double word [▶ 63], Bit 7
	1	8	brake released	Status double word [▶ 63], Bit 8
		9	softwarelimit	Status double word [▶ 63], Bit 9
		10	referenced	Status double word [▶ 63], Bit 10
		11	no part detected	Status double word [▶ 63], Bit 11
		12	gripped	Status double word [▶ 63], Bit 12
		13	position reached	Status double word [▶ 63], Bit 13
		14	pre gripped	Status double word [▶ 63], Bit 14
		15	reserved	-
1	2	16	workpiece lost	Status double word [▶ 64], Bit 16
		17	reserved	-
		18	reserved	-
		19	reserved	-
		20	reserved	-
		21	reserved	-
		22	reserved	-
		23	reserved	-
	3	24	reserved	-
		25	reserved	-
		26	reserved	-
		27	reserved	-
		28	reserved	-
		29	reserved	-
		30	reserved	-
		31	reserved	-

Actual position

- In bytes 4 – 7 of the cyclical input data, the current actual position of the module is transmitted, System parameters [▶ 34].
- The data format of the parameter is *signed 32 bits* and represents a value in micrometers [µm].

Reserve

- In bytes 8 – 11 of the cyclic input data no user data is currently transmitted.

Warning and error codes are unique, confusion is not possible, i.e. an error code cannot be a warning code.
Example: D9 is an error. This code is not used as a warning.

Diagnostic double word

In the diagnostic double word, consisting of a warning and an error word, more detailed information about pending warnings and errors is transmitted.

- PROFINET: In bytes **12 – 13** of the cyclical input data, **warning codes** of the module are transmitted. In bytes **14 – 15** of the cyclical input data, **error codes** of the module are transmitted, Diagnostics [▶ 44].
- EtherNet/IP™: In bytes **12 – 13** of the cyclical input data, **error codes** of the module are transmitted. In bytes **14 – 15** of the cyclical input data, **warning codes** of the module are transmitted, Diagnostics [▶ 44].

Warning and error codes are unique, confusion is not possible, i.e. an error code cannot be a warning code.
Example: D9 is an error. This code is not used as a warning.

NOTE

In the **EtherCAT** variant, the module implements "CANopen over EtherCAT" based on the communication profile DS301. The cyclic data are transmitted as PDOs (Process Data Objects). The PDO mapping is fixed and corresponds to the specifications described above for PROFINET or EtherNET/IP™. The cyclic output data are transmitted in a RPDO (Receive PDO) and the input data in a TPDO (Transmit PDO)

2.1.2 Acyclical data exchange

2.1.2.1 PROFINET

Execution of the acyclic data exchange complies with the specifications of the PNO (Profibus User Organization, www.profibus.com). For all the required information pertaining to acyclic data exchange, see chapter System parameters [▶ 34].

2.1.2.2 EtherCAT

The implementation of the acyclic data exchange corresponds to the EtherCATspecification of the CANopen-specific communication profile DS302. Here, the transmission of the "CANopen over EtherCAT" (CoE) protocol is used.

The acyclic communication is implemented via SDOs (Service Data Object). For SDO communication, an index must be specified. This is calculated from the parameter numbers (Parameter list [▶ 34]) plus an offset of 0x**2**000. The subindex is always **0**.

Example: The parameter "0x0600 – <min_pos>" is read with an SDO on index/subindex 0x**2**600/**0**.

2.1.2.3 EtherNet/IP™

The implementation of the acyclic data exchange corresponds to the specification of the ODVA (Open Devicenet Vendors Association) according to the Common Industrial Protocol (CIP™). The acyclic communication is carried out via a message box.

GetData

Message Type:	CIP Generic
Service Type:	Get Attribute Single
Class:	A2
Instance:	see chapter System parameters [▶ 34]
Attributes:	see the following table "Instance Attributes"
Destination Element:	user-defined day
Communication:	Set path to the desired SCHUNK device

SetData

Message Type:	CIP Generic
Service Type:	Set Attribute Single
Class:	A2
Instance:	see chapter System parameters [▶ 34]
Attributes:	see the following table "Instance Attributes"
Source Element:	user-defined day
Source Length:	Entering the corresponding length of data to be written
Communication:	Set path to the desired SCHUNK device

Instance Attributes

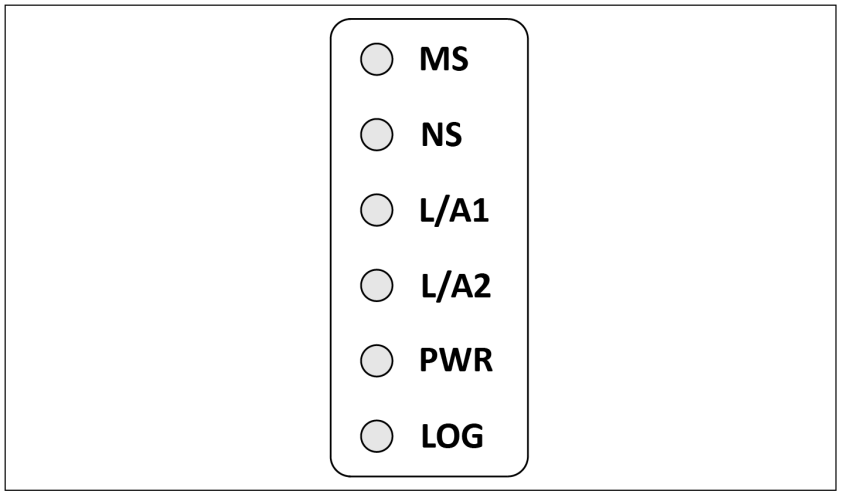
#	Name	Access	Type	Value / description	
1	Name	Get	SHORT_STRING	Parameter name (incl. length)	
2	ABCC data type	Get	Array of UINT	Data type of the instance value	
3	No. of Elements	Get	UINT	Number of elements for the specified data type	
4	Descriptor	Get	Array of UINT	Bit that describes the access rights for this instance	
				Bit:	Meaning:
				0	1 = Get Access
				1	1 = Set Accesss
				2	(reserve set to 0)
				3	1 = Write process data mapping possible
				4	1 = Write process data mapping possible
				5	1 = NVS parameter
				6	1 = Data notification activated
5	Value	Get / Set	Determined by attributes #2, #3 and #9	Instance Value	
6	Max value	Get		Maximum permissible parameter value	
7	Min. value	Get		Minimum permissible parameter value	
8	Default value	Get		Standard parameter value	
9	Number of subelements	Get	Array of UINT	Number of subelements in the parameter value. The standard value is 1, if not implemented in the application. The size of the array depends on attribute #3.	

Attributes #5–8 are converted from/to the CIP™ standard.

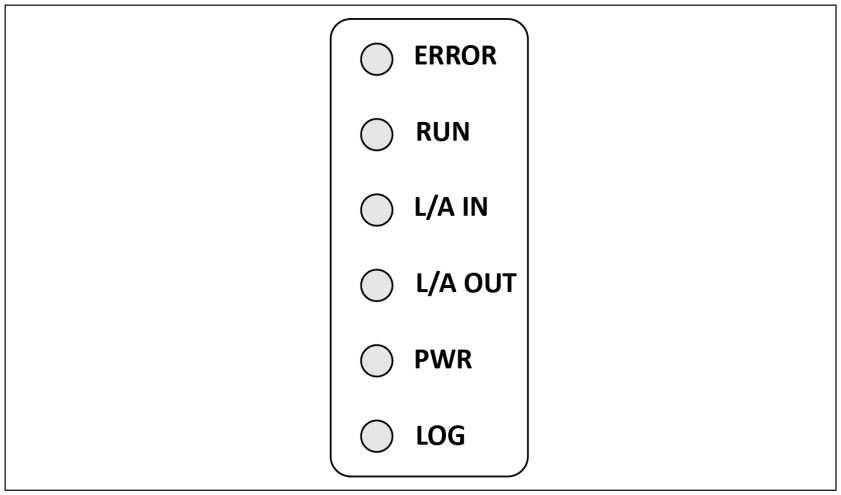
For all the required information pertaining to acyclic data exchange, see chapter System parameters [▶ 34].

2.2 LED status display

The LEDs indicate the user module states.



Arrangement of the LED status display PROFINET, EtherNet/IP™



Arrangement of the LED status display EtherCAT

For further information on the display of module states, see chapter Status display via LED status display [▶ 65].

2.3 Other interfaces

Web server

The module can be equipped with an integrated web server. Depending on the fieldbus interface, the web server can be used for commissioning and parameterization. The web server can be accessed with a browser via "http://IPADRESSE" if the module used contains the web server. "IPADRESSE" corresponds to the IP address of the module in the Ethernet network.

NOTE

The web server is **not** available for EtherCAT.

Service interface

The module is equipped with an additional interface that can be used for service actions exclusively carried out by SCHUNK.

2.4 Management of control logic

The control logic grants write permissions between the individual communication interfaces. The read permissions are not changed by the control logic.

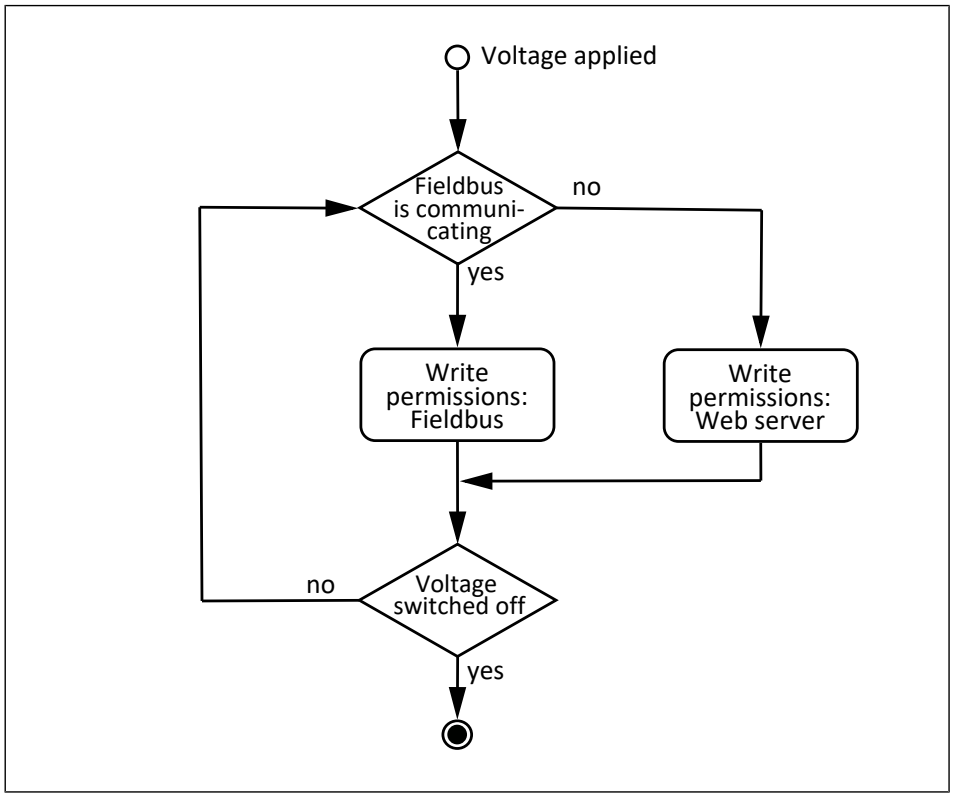
Read permissions

All communication interfaces have read permissions at all times.

Write permissions

Depending on the current communication situation, write permissions are automatically assigned by the module.

If the fieldbus interface has write permissions, this is indicated to the PLC by setting the status bit "bus control authority".



Management of the control logic

3 Module functions

3.1 Booting, shutting down and restarting the module

3.1.1 Booting

Short description

During the booting process, the electronics are started up and a self-test is performed. The internal hardware and connected communication interfaces are checked during the self-test.

Trigger

The booting can be triggered on the hardware side by applying the logic supply voltage, or triggered on the software side by a restart, Restarting [▶ 21].

NOTE

Ensure that all control bits are transferred in state 0. This eliminates any unexpected behavior while booting the module.

Possible diagnostic events

Events leading to warnings and/or errors are detected by the diagnostics. Below is a list of all possible diagnostic events.

Diagnostic event	Diagnostic code *
The connected hardware is not recognized (anymore).	0xF5 - ERR_UNKNOWN_HW
The internal memory is not recognized (anymore).	0xF6 - ERR_NO_BLOCK_DEV
The communication module is not recognized (anymore).	0xF7 - ERR_NO_COMM

* For further information, see chapter Diagnostics [▶ 44].

3.1.2 Shutting down

Short description

Before shutting down the module, data should be persistent, e.g. knowledge about the zero point of the module.

If preparations to shut down *are successful*, this is indicated to the PLC by setting the status bit "ready for shutdown".

If preparations to shut down *are not successful*, this is indicated to the PLC by setting the status bit "error" and the corresponding diagnostic code.

Trigger

Preparations to shut down are only permitted from within a defined system status and are triggered by setting the control bit "prepare for shutdown", Control double word [▶ 59]-Bit3.

System status

Preparations to shut down are only permitted from a stopped status. Stopped means that at the time of impact, the module is not actively moving or gripping a workpiece. Initiating preparations for shut down from the error state is permitted.

Note: If the error state was corrected by resetting the control bit "fast stop" to "0", the control bit "fast stop" must be reset to "1" before shutdown is initiated.

Possible diagnostic events

Events leading to warnings and/or errors are detected by the diagnostics. Below is a list of all possible diagnostic events.

Diagnostic event	Diagnostic code *
Shutting down cannot be prepared.	0xFA - ERR_SD_FAILED

* For further information, see chapter Diagnostics [▶ 44].

3.1.3 Restarting

Short description

A restart of the module can be triggered from within a defined system state on the software side.

Trigger

The restart is triggered by a setting of the control bit "softreset", Control double word [▶ 59]-Bit 4.

This function is enabled via the parameter <enable_softreset>. If the value of this parameter is "0", no restart via cyclic and acyclic data is possible, Parameter list [▶ 43].

System status

Restarting is permitted from a stopped status or after successfully preparing to shut down. Stopped means that at the time of impact, the module is not actively moving or gripping a workpiece. Initiating a restart from the error state is permitted.

Note: If the error state was corrected by resetting the control bit "fast stop" to "0", the control bit "fast stop" must be reset to "1" before a restart is initiated.

Possible diagnostic events

Events leading to warnings and/or errors are detected by the diagnostics. Below is a list of all possible diagnostic events.

Diagnostic event	Diagnostic code *
Shutting down cannot be prepared.	0xFA - ERR_SD_FAILED

* For further information, see chapter Diagnostics [▶ 44].

3.2 Movement functions

3.2.1 Tip mode

Short description

In tip mode, the module performs a movement in the positive or negative direction.

If the module is actively in tip mode, this is indicated to the user by setting the status bit "success".

Trigger

- In the positive direction, tip mode is triggered by setting the control bit "jog mode plus", Control double word [▶ 60]-Bit9.
- In the negative direction, tip mode is triggered by setting the control bit "jog mode minus, Control double word [▶ 60]-Bit8.

Movement parameter

No movement parameters need to be transmitted in order to execute the tip mode.

Tip mode is allowed
– in a non-referenced status over the entire mechanical stroke and
– in a referenced status within the software limits.

Finish

Tip mode is terminated by the following events:

- Resetting the control bit "jog mode plus" or "jog mode minus"
- Setting the control bit "stop"

NOTE

For modules with GKE: The brake is applied with a short delay time after resetting the control bit "jog mode plus" or "jog mode minus". This makes a direct reversal in tip mode possible, without the brake having to switch several times. Transmitting other requests before this delay time expires is not permitted, Cyclical output data [▶ 9].

Possible diagnostic events

Events leading to warnings and/or errors are detected by the diagnostics. Below is a list of all possible diagnostic events.

Diagnostic event	Diagnostic code *
The lower software limit is reached in the referenced state.	0xD5 - ERR_SOFT_LOW
The upper software limit is reached in the referenced state.	0xD6 - ERR_SOFT_HIGH
Drive is already blocked at the start of movement.	0xF4 - ERR_MOVE_BLOCKED
Drive blocked during movement.	0xF4 - ERR_MOVE_BLOCKED
Sending a request that is not permitted.	0x94 - WRN_NOT_FEASIBLE
Movement terminated by user.	0xD9 - ERR_FAST_STOP

* For further information, see chapter Diagnostics [▶ 44].

3.2.2 Referencing

Short description

The module defines its zero point during referencing, Reference point and zero point [▶ 7]. Successful referencing is indicated to the PLC by setting the status bit "referenced". When a reference movement is initiated, any status bit that may be set "referenced" is then reset.

Trigger

Referencing is triggered by setting the "reference" control bit, Control double word [▶ 60]-Bit 10.

Movement parameter

No movement parameters need to be transmitted to perform the referencing.

NOTE

During referencing, the stroke must be free from interfering contours during movement.

Finish

Referencing is terminated by the following events:

- The referencing point has been determined
- Setting the control bit "stop"

Possible diagnostic events

Events leading to warnings and/or errors are detected by the diagnostics. Below is a list of all possible diagnostic events.

Diagnostic event	Diagnostic code *
Referencing is taking too long.	0xF2 - ERR_NO_REF
The referencing point cannot be found.	0xF2 - ERR_NO_REF
Sending a request that is not permitted.	0x94 - WRN_NOT_FEASIBLE
Movement terminated by user.	0xD9 - ERR_FAST_STOP

* For further information, see chapter Diagnostics [▶ 44].

3.2.3 Absolute positioning movement

Short description

During absolute positioning, the module executes a positioning movement relative to the zero point of the module. Once the target position has been reached, it is reported to the user by setting the status bit "position reached".

A practical application example is described in chapter Application examples [▶ 52], example 1.

NOTE

Use positioning movements exclusively for positioning and **not** for gripping.
If the positioning movement is used for gripping workpieces, this represents a misuse, which will result in an error.

Trigger

Absolute positioning is only permitted in the referenced state and is triggered by setting the control bit "move to absolute position", Control double word [▶ 61]-Bit 13.

Movement parameter

In order to perform absolute positioning, the following movement parameters must be transmitted to the module:

- Position parameter
- Speed parameter

Absolute positioning is only permitted within the software limits.

Finish

Absolute positioning is terminated by the following events:

- Target position reached
- Setting the control bit "stop"

Possible diagnostic events

Events leading to warnings and/or errors are detected by the diagnostics. Below is a list of all possible diagnostic events.

Diagnostic event	Diagnostic code *
Positioning is taking too long.	0xF1 - ERR_MOV_ABORT_TO
Lower software limit is reached.	0xD5 - ERR_SOFT_LOW
Upper software limit is reached.	0xD6 - ERR_SOFT_HIGH
Drive is already blocked at the start of movement.	0xF4 - ERR_MOVE_BLOCKED
Drive blocked during movement.	0xF4 - ERR_MOVE_BLOCKED
Sending a request that is not permitted.	0x94 - WRN_NOT_FEASIBLE
Movement terminated by user.	0xD9 - ERR_FAST_STOP

* For further information, see chapter Diagnostics [▶ 44].

3.2.4 Relative positioning movement

Short description

During relative positioning, the module executes a positioning movement relative to the current actual position. Once the target position has been reached, it is reported to the user by setting the status bit "position reached".

A practical application example is described in chapter Application examples [▶ 52], example 2.

NOTE

Use positioning movements exclusively for positioning and **not** for gripping.
If the positioning movement is used for gripping workpieces, this represents a misuse, which will result in an error.

Trigger

Relative positioning is only permitted in a referenced status and is triggered by setting the control bit "move to relative position", Control double word [▶ 61]-Bit 14.

Movement parameter

In order to perform relative positioning, the following movement parameters must be transmitted to the module:

- Position parameter
- Speed parameter

Relative positioning is only permitted within the software limits.

Finish

Relative positioning is terminated by the following events:

- Target position reached
- Setting the control bit "stop"

Possible diagnostic events

Events leading to warnings and/or errors are detected by the diagnostics. Below is a list of all possible diagnostic events.

Diagnostic event	Diagnostic code *
Positioning is taking too long.	0xF1 - ERR_MOV_ABORT_TO
Lower software limit is reached.	0xD5 - ERR_SOFT_LOW
Upper software limit is reached.	0xD6 - ERR_SOFT_HIGH
Drive is already blocked at the start of movement.	0xF4 - ERR_MOVE_BLOCKED
Drive blocked during movement.	0xF4 - ERR_MOVE_BLOCKED
Sending a request that is not permitted.	0x94 - WRN_NOT_FEASIBLE
Movement terminated by user.	0xD9 - ERR_FAST_STOP

* For further information, see chapter Diagnostics [▶ 44].

3.2.5 Controlled stop

Short description

The module is able to stop active movements in a controlled manner.

Trigger

Controlled stops are triggered by setting the control bit "stop", Control double word [▶ 58]-Bit 1.

Movement parameter

No movement parameters need to be transmitted to perform the controlled stop.

Finish

The controlled stop is terminated automatically with the end of the movement.

Possible diagnostic events

Events leading to warnings and/or errors are detected by the diagnostics. Below is a list of all possible diagnostic events.

Diagnostic event	Diagnostic code *
The controlled stop is taking too long	0xF1 - ERR_MOV_ABORT_TO
Sending a request that is not permitted	0x94 - WRN_NOT_FEASIBLE
Movement terminated by user	0xD9 - ERR_FAST_STOP

* For further information, see chapter Diagnostics [▶ 44].

3.2.6 Terminating a movement

Short description

An active movement can be terminated and the module forced into a standstill.

Trigger

Movement termination is triggered by resetting the control bit "fast stop", Control double word [▶ 58]-Bit 0.

NOTE

The control bit "fast stop" is reset because the bit is "low-active" and therefore executed as fail-safe.

3.3 Handling a workpiece

3.3.1 Gripping a workpiece

Short description

The module is able to grip workpieces in a gripping movement. A successful gripping movement is displayed by setting the status bit "gripped". An unsuccessful gripping movement is displayed by setting the status bit "no part detected". A practical application example is described in chapter Application examples [▶ 53], example 3 – 4.

NOTE

Gripping is only permitted within the software limits, Hardware and software limits [▶ 6].

Trigger

Gripping workpieces is only permitted in a referenced state and is triggered by setting the control bit "grip work piece", Control double word [▶ 61]-Bit12.

Movement parameter

In order to grip workpieces, the following movement parameters and information must be issued to the module:

- Gripping force
- Gripping direction
  Note: The gripping direction is set in the control bit "grip direction", Control double word [▶ 59]-Bit 7.

Finish

Gripping workpieces is terminated by the following options:

- Workpiece successfully gripped
- A software limit has been reached
- Setting the control bit "stop"

Possible diagnostic events

Events leading to warnings and/or errors are detected by the diagnostics. Below is a list of all possible diagnostic events.

Diagnostic event	Diagnostic code *
Sending a request that is not permitted.	0x94 - WRN_NOT_FEASIBLE
Movement terminated by user.	0xD9 - ERR_FAST_STOP

* For further information, see chapter Diagnostics [▶ 44].

3.3.2 Re-gripping a workpiece

Short description

In order to prevent the loss of a workpiece during the gripping process, a "re-gripping" step can be parameterized. When re-gripping, after the workpiece stops moving, the gripper continues to be actively pressed in the gripping direction for a configurable period of time. This allows for a secure grip on tilted workpieces that only align themselves to the gripper after a delay. Re-gripping is displayed by setting the status bit "pre gripped". Practical application examples are described in chapter "Application examples [▶ 57]", example 16.

Trigger

If the parameters are set accordingly, re-gripping is always triggered automatically.

Movement parameters and parameterization

No additional movement parameters need to be transmitted to perform the re-gripping of workpieces.

In the parameter <grp_prehold_time> (Parameter list [▶ 36]) the time span of the holding pressure is stored. The maximum time span for re-gripping is 5000 ms.

If a time of 0 ms is stored in this parameter, re-gripping is not executed.

Finish

The re-gripping of workpieces is terminated by the following options:

- Time span of the re-gripping has expired
- Setting the control bit "stop"
- Setting the control bit "release work piece"

NOTE

If re-gripping is stopped by setting the control bit "stop", workpiece loss is to be assumed as the gripping process was not successfully completed. This is displayed by setting the status bit "workpiece lost".

Possible diagnostic events

Events leading to warnings and/or errors are detected by the diagnostics. Below is a list of all possible diagnostic events.

Diagnostic event	Diagnostic code *
Sending a request that is not permitted.	0x94 - WRN_NOT_FEASIBLE
Movement terminated by user.	0xD9 - ERR_FAST_STOP

* For further information, see chapter Diagnostics [▶ 44].

3.3.3 Holding the workpiece

Short description

After a successful gripping movement, the gripped workpiece is held. Modules can be equipped with a gripping force maintenance system (GKE).

- *Modules with GKE:* If a workpiece has been gripped, the gripping force maintenance is activated and the motor is switched off.
- *Modules without GKE:* If a workpiece has been gripped, the motor continues to be energized, thus maintaining the holding force.

Trigger

Workpieces are automatically held by the module. The user does not need to transmit any additional information to the module.

3.3.4 Releasing workpieces

Short description

The module is able to release gripped workpieces. In order to release workpieces, the module moves independently in the opposite direction of the last successful gripping command. Practical application examples are described in chapter Application examples [▶ 53], examples 5 – 6.

Trigger

Releasing workpieces is only permitted after a successful gripping movement and is triggered by setting the control bit "release work piece", Control double word [▶ 60] - Bit11.

NOTE

It is also permissible to release workpieces with absolute or relative positioning movements.

Movement parameter and parameterization

No movement parameters need to be transmitted to perform the release of workpieces.

The factory-set traverse path in which the module moves relatively during the release is recorded in the parameter <grp_prepos_delta> (Parameter list [▶ 37]). This parameter can be changed by the user.

Finish

The release of workpieces is terminated by the following options:

- Target position has been reached
- A software limit has been reached
- Setting the control bit "stop"

NOTE

The module conducts monitoring to ensure that no software limit is exceeded when releasing workpieces. When a software limit is reached, the movement is automatically ended.

Possible diagnostic events

Events leading to warnings and/or errors are detected by the diagnostics. Below is a list of all possible diagnostic events.

Diagnostic event	Diagnostic code *
The release is taking too long.	0xF1 - ERR_MOV_ABORT_TO
Drive is already blocked at the start of movement.	0xF4 - ERR_MOVE_BLOCKED
Drive blocked during movement.	0xF4 - ERR_MOVE_BLOCKED
Sending a request that is not permitted.	0x94 - WRN_NOT_FEASIBLE
Movement terminated by user.	0xD9 - ERR_FAST_STOP

* For further information, see chapter Diagnostics [▶ 44].

3.3.5 Remove workpiece manually (only for modules with GKE)

Short description

The user can manually remove a gripped workpiece from the module. After triggering, the brake of the module is released and the user has five seconds to manually move the fingers of the module or the base jaw in order to manually remove the workpiece.

NOTE

Because the user works directly on the module, manual **removal** of workpieces **is only permitted in an emergency**. To ensure that the module does not perform any unexpected movements, it is only permissible to trigger this function in the error state of the module!

Note: If the error state was corrected by resetting the control bit "fast stop" to "0", the control bit "fast stop" must be reset to "1" before manual removal can be initiated.

Trigger

The manual removal of workpieces is triggered by setting the control bit "release brake", Control double word [▶ 59]-Bit 5.

Movement parameter

No movement parameters need to be transmitted to perform the manual release of workpieces.

Finish

Manually gripping of workpieces is terminated by the following events:

- Response time of five seconds has expired
- Resetting the control bit "fast stop"

3.4 Additional functions

3.4.1 Workpiece loss detection

The module detects the loss of workpieces and indicates this by setting the status bit "workpiece lost".

NOTE

EGI 80: If the module has gripped a workpiece, the loss of the workpiece cannot be detected for technical reasons.

3.4.2 Position maintenance

After completing a movement, the module automatically changes to the position maintenance status.

For modules with GKE: The brake is activated and the motor control is switched off.

For modules without GKE: The position is maintained by active control of the motor.

3.4.3 Firmware update

The software of the module can be updated. The tool can be downloaded from the SCHUNK website.

3.4.4 Factory settings

Short description

The module can be reset to the factory settings by the software from a defined system status. All non-module-specific parameters are reset to the respective default values or to default settings.

Trigger

Resetting to factory settings can be done via the integrated web server.

System status

Resetting to the factory settings is only permitted in the event of an error. If the error state was corrected by resetting the control bit "fast stop" to "0", the control bit "fast stop" must be reset to "1" before initiating a reset to factory settings.

Possible diagnostic events

Events leading to warnings and/or errors are detected by the diagnostics. Below is a list of all possible diagnostic events.

Diagnostic event	Diagnostic code *
Resetting to factory settings is not possible	0x94 - WRN_NOT_FEASIBLE

- The module indicates successful resetting to factory settings by setting the status bit "ready for shutdown", Status double word [▶ 62]. It is mandatory to wait for this bit to be set before restarting or switching off the module.

- The first time the module is restarted after resetting to factory settings, the module reports the error 0x9F - WRN_SD_NOT_PREP even if the module has been properly shut down or restarted. This behavior is inherent to its functional principle, since the reference was also deleted when the system was reset to factory settings.

CAUTION

Material damage due to faulty usage!

- After resetting the module to factory settings, ensure that application-specific parameters are readjusted. Failure to do so may result in damage to the module itself or to adjacent machine parts.

4 System parameters

4.1 Value ranges

Value ranges

The following internal data types are used:

Data type	Threshold	Numerical values
BOOL	MIN_BOOL	0
	MAX_BOOL	1
UINT8	MIN_UINT8	0
	MAX_UINT8	255
UINT16	MIN_UINT16	0
	MAX_UINT16	65535
UINT32	MIN_UINT32	0
	MAX_UINT32	4294968295
FLOAT	MIN_FLOAT	-3.402823E+38
	MAX_FLOAT	3.402823E+38
CHAR	MIN_CHAR	0
	MAX_CHAR	255
ENUM	MIN_ENUM	0
	MAX_ENUM	255

4.2 Parameter list

In the following, all system-relevant parameters are listed according to the diagram "HEX-Code – <Parameter name>".

NOTE

The parameter list refers to parameters that are read out or written acyclically.

0x0110

0x0110 – <ctrl_authority>

Short description:	The current owner of the control logic can be read out with this parameter.
Access rights:	Read
Data type:	ENUM
Enumeration:	0 = Service interface
	1 = Fieldbus control
	2 = Web server

0x0118 **0x0118 – <err_code>**

Short description:	The pending error code can be read out with this parameter.
Access rights:	Read
Data type:	ENUM
Enumeration:	see chapter Error [▶ 47]

0x0120 **0x0120 – <wrn_code>**

Short description:	The pending warning code can be read out with this parameter.
Access rights:	Read
Data type:	ENUM
Enumeration:	see chapter Warnings [▶ 44]

0x0128 **0x0128 – <err_msg_req>**

Short description:	The corresponding error code can be requested by writing an index with this parameter.
Access rights:	Read and write
Data type:	UINT16

0x0130 **0x0130 – <err_msg_buffer>**

Short description:	The requested error code is displayed with this parameter.
Access rights:	Read
Data type:	CHAR[124]
Format:	ASCII-String

0x0200 **0x0200 – <set_pos>**

Short description:	The default value of the absolute target position can be read out or written with this parameter.
Access rights:	Read and write
Data type:	FLOAT
Unit:	Millimeter [mm]

0x0208 **0x0208 – <set_vel>**

Short description:	The default value of the set speed can be read out or written with this parameter.
Access rights:	Read and write
Data type:	FLOAT
Unit:	Millimeter per second [mm/s]

0x0220 **0x0220 – <set_force>**

Short description:	The default value of the set gripping force can be read out or written with this parameter.
Access rights:	Read and write
Data type:	FLOAT
Unit:	Newton [N]

0x0228 **0x0228 – <grp_dir>**

Short description:	The default value of the gripping direction can be read out or written with this parameter.
Access rights:	Read and write
Data type:	BOOL
Values:	0 = O.D. gripping 1 = I.D. gripping

0x0230 **0x0230 – <actual_pos>**

Short description:	This parameter can be used to read out the current actual position.
Access rights:	Read
Data type:	FLOAT
Unit:	Millimeter [mm]

0x0238 **0x0238 – <actual_vel>**

Short description:	This parameter can be used to read out the current actual speed.
Access rights:	Read
Data type:	FLOAT
Unit:	Millimeter per second [mm/s]

0x0380 **0x0380 – <grp_prehold_time>**

Short description:	This parameter can be used to read and write the time span for the re-gripping.
Access rights:	Read and write
Data type:	UINT16
Unit:	Millisecond [ms]

0x0500 **0x0500 – <module_type>**

Short description:	The module type can be read out with this parameter.
Access rights:	Read
Data type:	ENUM
Enumeration:	0 = EGI 40 1 = EGI 80

0x0508 **0x0508 – <use_softlimits>**

Short description:	This parameter can be used to read out whether software limits have been activated.
Access rights:	Read
Data type:	BOOL
Values:	0 = Software limits deactivated 1 = Software limits activated

0x0540 **0x0540 – <grp_prepos_delta>**

Short description:	With this parameter the position delta between gripping position and pre-position can be read out and written.
Access rights:	Read and write
Data type:	FLOAT
Unit:	Millimeter [mm]

0x0600 **0x0600 – <min_pos>**

Short description:	The lower software limit and therefore the smallest position value that can be approached by the module, can be read out and written with this parameter.
Access rights:	Read and write
Data type:	FLOAT
Unit:	Millimeter [mm]

0x0608 **0x0608 – <max_pos>**

Short description:	The upper software limit and therefore the largest position value that can be approached by the module, can be read out and written with this parameter.
Access rights:	Read and write
Data type:	FLOAT
Unit:	Millimeter [mm]

0x0628 **0x0628 – <min_vel>**

Short description:	The minimum movement speed with which the module can be moved can be read out with this parameter.
Access rights:	Read
Data type:	FLOAT
Unit:	Millimeter per second [mm/s]

0x0630 **0x0630 – <max_vel>**

Short description:	The maximum positioning speed with which the module can be moved can be read out with this parameter.
Access rights:	Read
Data type:	FLOAT
Unit:	Millimeter per second [mm/s]

0x0658 **0x0658 – <min_grp_force>**

Short description:	The minimum gripping force can be read out with this parameter.
Access rights:	Read
Data type:	FLOAT
Unit:	Newton [N]

0x0660 **0x0660 – <max_grp_force>**

Short description:	The maximum gripping force can be read out with this parameter.
Access rights:	Read
Data type:	FLOAT
Unit:	Newton [N]

0x0800 **0x0800 – <min_err_mot_volt>**

Short description:	The lower error limit value of the supply voltage of the drive can be read out with this parameter.
Access rights:	Read
Data type:	FLOAT
Unit:	Volt [V]

0x0808 **0x0808 – <max_err_mot_volt>**

Short description:	The upper error limit value of the supply voltage of the drive can be read out with this parameter.
Access rights:	Read
Data type:	FLOAT
Unit:	Volt [V]

0x0810 **0x0810 – <min_err_lgc_volt>**

Short description:	The lower error limit value of the supply voltage of the main board can be read out with this parameter.
Access rights:	Read
Data type:	FLOAT
Unit:	Volt [V]

0x0818 **0x0818 – <max_err_lgc_volt>**

Short description:	The upper error limit value of the supply voltage of the main board can be read out with this parameter.
Access rights:	Read
Data type:	FLOAT
Unit:	Volt [V]

0x0820 **0x0820 – <min_err_lgc_temp>**

Short description:	The lower error limit value of the temperature of the main board can be read out with this parameter.
Access rights:	Read
Data type:	FLOAT
Unit:	Degrees Celsius [°C]

0x0828 **0x0828 – <max_err_lgc_temp>**

Short description:	The upper error limit value of the temperature of the main board can be read out with this parameter.
Access rights:	Read
Data type:	FLOAT
Unit:	Degrees Celsius [°C]

0x0840 **0x0840 – <lgc_temp>**

Short description:	The current measured temperature of the main board can be read out with this parameter.
Access rights:	Read
Data type:	FLOAT
Unit:	Degrees Celsius [°C]

0x0870 **0x0870 – <lgc_volt>**

Short description:	The current measured supply voltage of the main board can be read out with this parameter.
Access rights:	Read
Data type:	FLOAT
Unit:	Volt [V]

0x0878 **0x0878 – <mot_volt>**

Short description:	The current measured supply voltage of the power unit can be read out with this parameter.
Access rights:	Read
Data type:	FLOAT
Unit:	Volt [V]

0x0880 **0x0880 – <min_wrn_mot_volt>**

Short description:	The lower warning limit of the supply voltage of the power unit can be read out and written with this parameter.
Access rights:	Read and write
Data type:	FLOAT
Unit:	Volt [V]

0x0888 **0x0888 – <max_wrn_mot_volt>**

Short description:	The upper warning limit of the supply voltage of the power unit can be read out and written with this parameter.
Access rights:	Read and write
Data type:	FLOAT
Unit:	Volt [V]

0x0890 **0x0890 - <min_wrn_lgc_volt>**

Short description:	The lower warning limit of the supply voltage of the logic part can be read out and written with this parameter.
Access rights:	Read and write
Data type:	FLOAT
Unit:	Volt [V]

0x0898 **0x0898 – <max_wrn_lgc_volt>**

Short description:	The upper warning limit of the supply voltage of the logic part can be read out and written with this parameter.
Access rights:	Read and write
Data type:	FLOAT
Unit:	Volt [V]

0x08A0 **0x08A0 – <min_wrn_lgc_temp>**

Short description:	The lower warning limit of the temperature of the logic part can be read out and written with this parameter.
Access rights:	Read
Data type:	FLOAT
Unit:	Degrees Celsius [°C]

0x08A8 **0x08A8 – <max_wrn_lgc_temp>**

Short description:	The upper warning limit of the temperature of the logic part can be read out and written with this parameter.
Access rights:	Read
Data type:	FLOAT
Unit:	Degrees Celsius [°C]

0x1000 **0x1000 – <serial_no_txt>**

Short description:	The serial number of the module can be read out with this parameter.
Access rights:	Read
Data type:	CHAR[16]
Format:	ASCII-String

0x1008 **0x1008 – <order_no_txt>**

Short description:	The order number of the module can be read out with this parameter.
Access rights:	Read
Data type:	CHAR[16]
Format:	ASCII-String

0x1020 **0x1020 – <serial_no_num>**

Short description:	The number-coded serial number of the module can be read out with this parameter.
Access rights:	Read
Data type:	UINT32
Format:	ASCII-String

0x1100 **0x1100 – <sw_build_date>**

Short description:	The creation date of the firmware version can be read out with this parameter.
Access rights:	Read
Data type:	CHAR[12]
Format:	ASCII-String

0x1108 **0x1108 – <sw_build_time>**

Short description:	The creation time of the firmware version can be read out with this parameter.
Access rights:	Read
Data type:	CHAR[9]
Format:	ASCII-String

0x1118 **0x1118 – <sw_version_txt>**

Short description:	The version of the software can be read out as a text with this parameter.
Access rights:	Read
Data type:	CHAR[22]
Format:	ASCII-String

0x1120 **0x1120 – <comm_version_txt>**

Short description:	The firmware version of the communication block can be read out with this parameter.
Access rights:	Read
Data type:	CHAR[12]
Format:	ASCII-String

0x1330 **0x1330 – <enable_softreset>**

Short description:	The "Restart" function can be enabled with this parameter.
Access rights:	Read and write
Data type:	BOOL
Values:	0 = function switched off 1 = function switched on

0x1400 **0x1400 – <system_uptime>**

Short description:	This parameter can be used to read out the operating time that has elapsed since the last module (re)start.
Access rights:	Read
Data type:	UINT32
Unit:	Seconds [s]

4.3 Parameter configuration

All system parameters for which the user has write permissions can be parameterized via acyclic data exchange Acyclical data exchange [▶ 14].

For further information on parameterization, see chapter Parameter list [▶ 34].

5 Diagnostics

The diagnostics are used to monitor the system and respond to detected diagnostic events by generating the appropriate diagnostic codes. The diagnostics of the module run permanently in the background and is not visible to the user.

Diagnostic events

Diagnostic events are subdivided into warning and error events. Information about diagnostic events that have occurred is transmitted in the cyclical input data.

5.1 Warnings

If the diagnostics detect that a warning event has occurred, the module enters a warning state. A warning code is generated and transmitted. The issue related to a warning is displayed by setting the status bit "warning".

NOTE

If more than one warning is present, the last occurring warning code is transmitted.

Warning state

In a warning state, the module remains ready for operation but may be operated at the limit of the error state.

Warning code

Each detectable warning event includes a unique warning code that is transmitted in the cyclical input data.

Acknowledging

Warnings are both acknowledgeable and self-acknowledging.

By setting the control bit "acknowledge" the acknowledgment of an existing warning is triggered, Control double word [▶ 58]-Bit 2.

If the cause of the warning event no longer exists at that time, the warning is acknowledged. If the cause of the warning event still exists, the warning cannot be acknowledged at that time and remains active. If the module detects that the cause of an existing warning event no longer exists, this warning is automatically acknowledged.

Recognizable warning events

Listed below are all warning events and their associated warning codes that can be detected by the module.

0x90 - WRN_LGC_TEMP_LO

Diagnostic event:	The measured logic temperature is too low.
Ability to acknowledge:	self-acknowledging

0x91 - WRN_LGC_TEMP_HI

Diagnostic event:	The measured logic temperature is too high.
Ability to acknowledge:	self-acknowledging

0x92 - WRN_MOT_TEMP_LO

Diagnostic event:	The measured motor temperature is too low.
Ability to acknowledge:	self-acknowledging

0x93 - WRN_MOT_TEMP_HI

Diagnostic event:	The measured motor temperature is too high.
Ability to acknowledge:	self-acknowledging

0x94 - WRN_NOT_FEASIBLE

Diagnostic event:	The request cannot be executed. When triggering 0x94 - WRN_NOT_FEASIBLE a message is stored in the error memory which describes the cause of the error in more detail. The error memory can be read out via the parameters 0x0128 – <err_msg_req> and 0x0130 – <err_msg_buffer>, Parameter list [▶ 34].
Ability to acknowledge:	acknowledgeable/self-acknowledging

0x96 - WRN_LGC_VOLT_LO

Diagnostic event:	The measured logic supply voltage is too low.
Ability to acknowledge:	self-acknowledging

0x97 - WRN_LGC_VOLT_HI

Diagnostic event:	The measured logic supply voltage is too high.
Ability to acknowledge:	self-acknowledging

0x98 - WRN_MOT_VOLT_LO

Diagnostic event:	The measured power supply voltage is too low.
Ability to acknowledge:	self-acknowledging

0x99 - WRN_MOT_VOLT_HI

Diagnostic event:	The measured power supply voltage is too high.
Ability to acknowledge:	self-acknowledging

0x9B - WRN_FLASH_FAILED

Diagnostic event:	The firmware file used for the software update has not passed the preliminary check and may not be compatible with this module.
Ability to acknowledge:	acknowledgeable

0x9D - WRN_FACT_FAILED

Diagnostic event:	Error when resetting to factory settings.
Ability to acknowledge:	acknowledgeable/self-acknowledging

0x9E - WRN_AUTH_FAILED

Diagnostic event:	When attempting to transfer control to the service interface, the module was not in transfer state. The transfer of the control logic has failed.
Ability to acknowledge:	acknowledgeable/self-acknowledging

0x9F - WRN_SD_NOT_PREP

Diagnostic event:	The shutdown was not requested before the module was disconnected from the voltage supply.
Ability to acknowledge:	self-acknowledging

5.2 Error

If the diagnostics detect that a warning event has occurred, the module enters an error state. An error code is generated and transmitted. The issue related to an error is displayed by setting the status bit "error".

NOTE

If more than one error is present, the last occurring error code is transmitted.

Error state

In an error state, the module is not longer ready for operation. By changing to the error state, the module is forced into a standstill.

For modules with GKE: The brake is applied.

Error code

Each detectable error event includes a unique error code that is transmitted in the cyclical input data.

Acknowledging

Errors can be separated into those requiring acknowledgment and errors that are non-acknowledgeable.

Errors requiring acknowledgment: By setting the control bit "acknowledge", the acknowledgment of an error requiring acknowledgment is triggered.

If the cause of the error event no longer exists at that time, the error is acknowledged. If the cause of the error event still exists, the error cannot be acknowledged at that time and remains active.

Non-acknowledgeable errors: If a serious error occurs, the module may become damaged or destroyed if restarted. In these cases, the error state cannot be left. The module must be inspected by SCHUNK Service or sent in directly.

Recognizable error events

Listed below are all error events and their associated error codes that can be detected by the module.

0x6C - ERR_MOT_TEMP_LO

Diagnostic event:	The measured motor temperature is too low.
Ability to acknowledge:	requiring acknowledgment

0x6D - ERR_MOT_TEMP_HI

Diagnostic event:	The measured motor temperature is too high.
Ability to acknowledge:	requiring acknowledgment

0x70 - ERR_LGC_TEMP_LO

Diagnostic event:	The measured logic temperature is too low.
Ability to acknowledge:	requiring acknowledgment

0x71 - ERR_LGC_TEMP_HI

Diagnostic event:	The measured logic temperature is too high.
Ability to acknowledge:	requiring acknowledgment

0x72 - ERR_LGC_VOLT_LO

Diagnostic event:	The measured logic supply voltage is too low.
Ability to acknowledge:	requiring acknowledgment

0x73 - ERR_LGC_VOLT_HI

Diagnostic event:	The measured logic supply voltage is too high.
Ability to acknowledge:	requiring acknowledgment

0x74 - ERR_MOT_VOLT_LO

Diagnostic event:	The measured power supply voltage is too low.
Ability to acknowledge:	requiring acknowledgment

0x75 - ERR_MOT_VOLT_HI

Diagnostic event:	The measured power supply voltage is too high.
Ability to acknowledge:	requiring acknowledgment

0x8A - ERR_ENC_PHASE

Diagnostic event:	The phase shift of the encoder signals is deviating.
Ability to acknowledge:	not acknowledgeable

0x8B - ERR_ENC_SIN_LO

Diagnostic event:	The measured sine wave signal is too low.
Ability to acknowledge:	not acknowledgeable

0x8C - ERR_ENC_SIN_HI

Diagnostic event:	The measured sine wave signal is too high.
Ability to acknowledge:	not acknowledgeable

0x8D - ERR_ENC_COS_LO

Diagnostic event:	The measured cosine wave signal is too low.
Ability to acknowledge:	not acknowledgeable

0x8E - ERR_ENC_COS_HI

Diagnostic event:	The measured cosine wave signal is too high.
Ability to acknowledge:	not acknowledgeable

0x8F - ERR_ENC_SHORTCUT

Diagnostic event:	The encoder signals are identical.
Ability to acknowledge:	not acknowledgeable

0xD5 - ERR_SOFT_LOW

Diagnostic event:	The lower software limit has been reached.
Ability to acknowledge:	requiring acknowledgment

0xD6 - ERR_SOFT_HIGH

Diagnostic event:	The upper software limit has been reached.
Ability to acknowledge:	requiring acknowledgment

0xD9 - ERR_FAST_STOP

Diagnostic event:	A fast stop was triggered.
Ability to acknowledge:	requiring acknowledgment

0xDE - ERR_CURRENT

Diagnostic event:	The maximum current has been exceeded.
Ability to acknowledge:	requiring acknowledgment

0xE4 - ERR_TOO_FAST

Diagnostic event:	The maximum permissible speed was exceeded by a factor of 1.2.
Ability to acknowledge:	requiring acknowledgment

0xF0 - ERR_REF_ABORT_TO

Diagnostic event:	Referencing could not be performed within a defined period of time.
Ability to acknowledge:	requiring acknowledgment

0xF1 - ERR_MOV_ABORT_TO

Diagnostic event:	Positioning could not be performed within a defined period of time.
Ability to acknowledge:	requiring acknowledgment

0xF2 - ERR_NO_REF

Diagnostic event:	The module could not find a reference point.
Ability to acknowledge:	requiring acknowledgment

0xF4 - ERR_MOVE_BLOCKED

Diagnostic event:	The drive was blocked.
Ability to acknowledge:	requiring acknowledgment

0xF5 - ERR_UNKNOWN_HW

Diagnostic event:	The hardware is not recognized.
Ability to acknowledge:	not acknowledgeable

0xF6 - ERR_NO_BLOCK_DEV

Diagnostic event:	The internal memory is not recognized.
Ability to acknowledge:	not acknowledgeable

0xF7 - ERR_NO_COMM

Diagnostic event:	The communication block is not recognized.
Ability to acknowledge:	not acknowledgeable

0xF8 - ERR_WRONG_HW

Diagnostic event:	Hardware and firmware are incompatible.
Ability to acknowledge:	not acknowledgeable

0xFA - ERR_SD_FAILED

Diagnostic event:	Preparing to shutdown is not feasible.
Ability to acknowledge:	requiring acknowledgment

6 Appendix

6.1 Application examples

The following application examples describe the operation and behavior of the module.

EXAMPLE 1 **Absolute positioning movement**

The current actual position of the module is 30 mm. The controller sends the following request to the module:

- Position parameter = 80000 µm (≙ 80 mm)
- Speed parameter = 200000 µm/s (≙ 200 mm/s)
- Control bit "move to absolute position" is set

The module then moves to the absolute position 80 mm with respect to the zero point. The maximum speed that the module can reach during this positioning process is 200 mm/s.

After completing the movement, the current actual position of the module is 80 mm ± tolerance of the module-specific positioning accuracy.

For further information, see chapter Absolute positioning movement [▶ 25].

EXAMPLE 2 **Relative positioning movement**

The current actual position of the module is 66 mm. The controller sends the following request to the module:

- Position parameter = -20000 µm (≙ -20 mm)
- Speed parameter = 135000 µm/s (≙ 135 mm/s)
- Control bit "move to relative position" is set

The module then moves -20 mm from the current actual position. The maximum speed that the module can reach during this positioning movement is 135 mm/s.

→ After completing the movement, the current actual position of the module is 46 mm ± tolerance of the module-specific positioning accuracy.

For further information, see chapter Relative positioning movement [▶ 26].

EXAMPLE 3 **Gripping the workpiece (1)**

A workpiece is available and should be gripped. In order to grip the workpiece, the controller sends the following request to the module:

- Control bit state "grip direction" = 1 (≙ I.D. gripping)
- Gripping force = 60000 mN (≙ 60 N)
- Control bit "grip work piece" is set

→ The module then performs a gripping movement as I.D. gripping, the base jaws move apart. The workpiece is gripped with 60 N and the status bit "gripped" is set.

For further information, see chapter Gripping a workpiece [▶ 28].

EXAMPLE 4 **Gripping the workpiece (2)**

No workpiece is available, however gripping is supposed to take place. In order to grip the intended workpiece, the controller sends the following request to the module:

- Control bit state "grip direction" = 0 (≙ O.D. gripping)
- Gripping force = 25000 mN (≙ 25 N)
- Control bit "grip work piece" is set

→ The module then performs a gripping movement as O.D. gripping, the base jaws move towards each other. When a software limit is reached, it is recognized that no workpiece has been gripped. This is displayed by setting the status bit "no part detected".

For further information, see chapter Gripping a workpiece [▶ 28].

EXAMPLE 5 **Releasing workpieces (1)**

A workpiece was first gripped from the **inside**. The software limits are at 0 and 100 mm. The current gripping position is 60 mm. The value of the parameter "grp_prepos_delta" is 5 (≙ 5 mm). The controller sends the following request to the module:

- Control bit "release work piece" is set

The module knows that the workpiece was first gripped from the inside and then, taking into account the software limits, moves -5 mm from the current actual position.

→ The process ends when the target position is reached. The current actual position of the module is 55 mm ± tolerance of the module-specific positioning accuracy. Status bit "position reached" is set.

For further information, see chapter Releasing workpieces [▶ 30].

EXAMPLE 6 **Releasing workpieces (2)**

A workpiece was first gripped from the **outside**. The software limits are at 0 and 100 mm. The current gripping position is 95 mm. The value of the parameter "grp_prepos_delta" is 10 (≙ 10 mm). The controller sends the following request to the module:

- Control bit "release work piece" is set

The module knows that the workpiece was first gripped from the outside and then, taking into account the software limits, moves -10 mm from the current actual position.

→ The process ends when a software limit is reached. The current actual position of the module is 100 mm ± tolerance of the module-specific positioning accuracy. Status bit "position reached" is set.

For further information, see chapter Releasing workpieces [▶ 30].

EXAMPLE 7 **Request to the module (1)**

The module is ready for operation, referenced, is not actively moving and has not gripped a workpiece. The control bit "move to absolute position" is set and the transmitted movement parameters are ok. These requirements reflect those of a permissible request.

→ Absolute positioning is triggered and ends when the target position is reached.

For further information, see chapter Cyclical output data [▶ 9].

EXAMPLE 8 **Request to the module (2)**

The module is ready for operation, referenced, is not actively moving and has not gripped a workpiece. The control bit "move to absolute position" is set and the transmitted movement parameters are ok. These requirements reflect those of a permissible request.

→ Absolute positioning is triggered.

Before the target position is reached, the control bit "move to absolute position" is reset. These requirements reflect those of a permissible request.

→ The movement ends when the target position is reached.

For further information, see chapter Cyclical output data [▶ 9].

EXAMPLE 9 **Request to the module (3)**

The module is ready for operation, referenced, is not actively moving and has not gripped a workpiece. The control bit "move to absolute position" is set and the transmitted movement parameters are ok. These requirements reflect those of a permissible request.

→ Absolute positioning is triggered.

Before the target position is reached, the control bit "move to absolute position" is reset and the control bit "stop" is set. These requirements reflect those of a permissible request.

→ Absolute positioning is ended with a controlled stop. The actual position reached at the end of the movement depends on when the control bit "stop" was set.

For further information, see chapter Cyclical output data [▶ 9].

EXAMPLE 10 **Request to the module (4)**

The module is ready for operation, referenced, is not actively moving and has not gripped a workpiece. The control bit "move to absolute position" is set and the transmitted movement parameters are ok. These requirements reflect those of a permissible request.

→ Absolute positioning is triggered.

Before the target position is reached, in addition to the set control bit that is still set "move to absolute position", control bit stop is set. These requirements reflect those of an impermissible request.

→ The absolute positioning is ended in a controlled manner and the status bit "not feasible" is set. The actual position reached at the end of the movement depends on when the control bit "stop" was set.

→ Error: Impermissible bit combination during active movement.

For further information, see chapter Cyclical output data [▶ 9].

EXAMPLE 11 **Request to the module (5)**

The module is ready for operation, referenced, is not actively moving and has not gripped a workpiece. The control bit "move to absolute position" is set and the transmitted movement parameters are ok. These requirements reflect those of a permissible request.

→ Absolute positioning is triggered and ends when the target position is reached.

In addition to the control bit that is still set "move to absolute position", the control bit "stop" is set. These requirements reflect those of a impermissible request.

→ The absolute positioning is ended in a controlled manner and the status bit "not feasible" is set.

→ Error: Impermissible bit combination

For further information, see chapter Cyclical output data [▶ 9].

EXAMPLE 12 **Request to the module (6)**

The module is ready for operation, not referenced, is not actively moving and has not gripped a workpiece. The control bit "move to absolute position" is set and the transmitted movement parameters are ok. These requirements reflect those of a permissible request.

→ The status bit "not feasible" is set.

→ Error: The module is not referenced.

For further information, see chapter Cyclical output data [▶ 9].

EXAMPLE 13 **Request to the module (7)**

The module is **not** ready for operation, referenced, is not actively moving and has not gripped a workpiece. The control bit "move to absolute position" is set and the transmitted movement parameters are ok. These requirements reflect those of a permissible request.

→ The status bit "not feasible" is set.

→ Error: The module is not ready for operation, because there is an error.

For further information, see chapter Cyclical output data [▶ 9].

EXAMPLE 14 **Request to the module (8)**

The module is ready for operation, referenced, performs **one** active movement and has not gripped a workpiece. The control bit "move to absolute position" is set and the transmitted movement parameters are ok. These requirements reflect those of a permissible request.

→ The movement is ended with a controlled stop and the status bit "not feasible" is set.

→ Error: Impermissible bit combination during active movement.

For further information, see chapter Cyclical output data [▶ 9].

EXAMPLE 15 **Request to the module (9)**

The module is ready for operation, referenced, is not actively moving and has not gripped a workpiece. The control bit "move to absolute position" is set and the transmitted movement parameters are **faulty**. These requirements reflect those of a permissible request.

→ The status bit "not feasible" is set.

→ Error: At least one movement parameter is outside the exact limits.

For further information, see chapter Cyclical output data [▶ 9].

EXAMPLE 16

Re-gripping a workpiece (1)

The module is prepared to grip a workpiece. The workpiece is present and the application requires re-gripping to prevent workpiece loss. In order to grip the workpiece, the controller sends the following request to the module:

- Parameter value <grp_prehold_time> = 2300 ms
- Status control bit grip direction = 0 (≙ gripped from outside)
- Gripping force = 50000 mN (≙ 50 N)

→ The control bit "grip work piece" is set.

→ The module then performs a gripping movement as an O.D. gripper. The base jaws move towards each other. The workpiece contact is detected and indicated by setting the status bit "pre gripped". Gripping is maintained for a further 2.3 seconds, i.e. the gripper automatically compensates for any change in the workpiece diameter that is visible to the gripper. A change such as this can occur, for example, with a workpiece that is initially gripped in a slightly tilted position, and which can only align itself in the gripper after a delay.

If a workpiece has been gripped after the time period has expired, this is indicated by setting the status bit "gripped". The status bit "pre gripped" remains set.

If **no** workpiece has been gripped once the time period has expired (loss of workpiece while re-gripping) this is indicated by setting the status bit "no part detected". In this case the status bit "pre gripped" is reset again.

For further information, see chapter Re-gripping a workpiece [▶ 29].

6.2 Control double word

The control bits of the control double word are described in detail below. For a clear illustration of the control double word, see chapter Cyclical output data [▶ 10].

Bit 0 - fast stop

Edge change or status	Module reaction
0 -> 1 or 1	no reaction
1 -> 0 or 0	The module performs a quick stop, Terminating a movement [▶ 27].

If this bit has the state 0 and one of the following bits is set at this time, there is an invalid bit combination.

- all bits

Note: As long as the control bit "fast stop" is set to "0", no other operations can be performed until the bit is set to "1" again. This also applies to operations without movement, such as
– Switch off (Shutting down [▶ 20]),
- Restart (Restarting [▶ 21]),
– Manual removal (Remove workpiece manually (only for modules with GKE) [▶ 31]) and
– Reset to factory settings (Factory settings [▶ 32]).

Bit 1 - stop

Edge change	Module reaction
0 -> 1	The module performs a controlled stop, Controlled stop [▶ 27].
1 -> 0	no reaction

If this bit changes to the status 1 while one of the following bits is set, there is an invalid bit combination.

- Bit: 2, 3, 4, 5, 8, 9, 10, 11, 12, 13, 14

Bit 2 - acknowledge

Edge change	Module reaction
0 -> 1	The module tries to acknowledge all existing warnings and errors, Warnings [▶ 44], Error [▶ 47].
1 -> 0	no reaction

If this bit changes to the status 1 while one of the following bits is set, there is an invalid bit combination.

- Bit: 1, 3, 4, 5, 8, 9, 10, 11, 12, 13, 14

Bit 3 - prepare for shutdown

Edge change	Module reaction
0 -> 1	The module is preparing for shutdown, Shutting down [▶ 20].
1 -> 0	no reaction

If this bit changes to the status 1 while one of the following bits is set, there is an invalid bit combination.

- Bit: 1, 2, 4, 5, 8, 9, 10, 11, 12, 13, 14

Bit 4 - softreset

Edge change	Module reaction
0 -> 1	The module is restarted on the software side, Restarting [▶ 21].
1 -> 0	no reaction

If this bit changes to the status 1 while one of the following bits is set, there is an invalid bit combination.

- Bit: 1, 2, 3, 5, 8, 9, 10, 11, 12, 13, 14

Bit 5 - release brake

Edge change	Module reaction
0 -> 1	The brake is released in order to manually remove a workpiece, Remove workpiece manually (only for modules with GKE) [▶ 31].
1 -> 0	no reaction

If this bit changes to the status 1 while one of the following bits is set, there is an invalid bit combination.

- Bit: 1, 2, 3, 4, 8, 9, 10, 11, 12, 13, 14

Bit 6 - reserved

Edge change	Module reaction
0 -> 1	no reaction
1 -> 0	no reaction

Bit 7 - grip direction

Status	Module reaction
0	During a gripping process, the gripping is done from the outside.
1	During a gripping process, the gripping is done from the inside.

Bit 8 - jog mode minus

Edge change	Module reaction
0 -> 1	The module executes a movement in the negative direction of movement, Tip mode [▶ 22].
1 -> 0	no reaction

If this bit changes to the status 1 while one of the following bits is set, there is an invalid bit combination.

- Bit: 1, 2, 3, 4, 5, 9, 10, 11, 12, 13, 14

Bit 9 - jog mode plus

Edge change	Module reaction
0 -> 1	The module executes a movement in the positive direction of movement, Tip mode [▶ 22].
1 -> 0	no reaction

If this bit changes to the status 1 while one of the following bits is set, there is an invalid bit combination.

- Bit: 1, 2, 3, 4, 5, 8, 10, 11, 12, 13, 14

Bit 10 - reference

Edge change	Module reaction
0 -> 1	The module executes a reference run, Referencing [▶ 24].
1 -> 0	no reaction

If this bit changes to the status 1 while one of the following bits is set, there is an invalid bit combination.

- Bit: 1, 2, 3, 4, 5, 8, 9, 11, 12, 13, 14

Bit 11 - release work piece

Edge change	Module reaction
0 -> 1	The module releases a workpiece, Releasing workpieces [▶ 30].
1 -> 0	no reaction

If this bit changes to the status 1 while one of the following bits is set, there is an invalid bit combination.

- Bit: 1, 2, 3, 4, 5, 8, 9, 10, 12, 13, 14

Bit 12 - grip work piece

Edge change	Module reaction
0 -> 1	The module performs a gripping movement, Gripping a workpiece [▶ 28]
1 -> 0	no reaction

If this bit changes to the status 1 while one of the following bits is set, there is an invalid bit combination.

- Bit: 1, 2, 3, 4, 5, 8, 9, 10, 11, 13, 14

Bit 13 - move to absolute position

Edge change	Module reaction
0 -> 1	The module performs a positioning movement to an absolute position, Absolute positioning movement [▶ 25].
1 -> 0	no reaction

If this bit changes to the status 1 while one of the following bits is set, there is an invalid bit combination.

- Bit: 1, 2, 3, 4, 5, 8, 9, 10, 11, 12, 14

Bit 14 - move to relative position

Edge change	Module reaction
0 -> 1	The module performs a positioning movement to a relative position, Referencing [▶ 24].
1 -> 0	no reaction

If this bit changes to the status 1 while one of the following bits is set, there is an invalid bit combination.

- Bit: 1, 2, 3, 4, 5, 8, 9, 10, 11, 12, 13

Bit 15 – 31 - reserved

Edge change	Module reaction
0 -> 1	no reaction
1 -> 0	no reaction

6.3 Status double word

The status bits of the status double word are described in detail below. For a clear illustration of the status double word, see chapter Cyclical input data [▶ 12].

Bit 0 - ready for operation

Status	Module feedback
0	The module is not ready for operation.
1	The module is ready for operation.

Bit 1 - bus control authority

Status	Module feedback
0	The fieldbus does not have a control logic.
1	The fieldbus has a control logic.

Bit 2 - ready for shutdown

Status	Module feedback
0	The module is not ready to be shut down.
1	The module is ready to be shut down.

Bit 3 - not feasible

Status	Module feedback
0	The request sent to the module is feasible.
1	The request sent to the module is not feasible.

Bit 4 - success

Status	Module feedback
0	No information is reported.
1	The last request sent to the module was successfully processed.

Bit 5 - reserved

Status	Module feedback
0	No information is reported.
1	No information is reported.

Bit 6 - warning

Status	Module feedback
0	There is no warning.
1	The issue related to a warning is displayed.

Bit 7 - error

Status	Module feedback
0	There is no error.
1	The issue related to an error is displayed.

Bit 8 - brake released

Status	Module feedback
0	The brake is activated.
1	The brake is released.

Bit 9 - softwarelimit

Status	Module feedback
0	No information is reported.
1	A software limit has been exceeded.

Bit 10 - referenced

Status	Module feedback
0	The module is not referenced.
1	The module is referenced.

Bit 11 - no part detected

Status	Module feedback
0	No information is reported.
1	The gripping process was not successful.

Bit 12 - gripped

Status	Module feedback
0	No information is reported.
1	The gripping process was successful.

Bit 13 - position reached

Status	Module feedback
0	No information is reported.
1	The module has reached the target position.

Bit 14 - pre gripped

Status	Module feedback
0	No information is reported.
1	A re-gripping action is/was carried out.

Bit 15 - reserved

Status	Module feedback
0	No information is reported.
1	No information is reported.

Bit 16 - workpiece lost

Status	Module feedback
0	No information is reported.
1	The gripped workpiece was lost.

Bit 17 – 31 - reserved

Status	Module feedback
0	No information is reported.
1	No information is reported.

6.4 Status display via LED status display

Below is a list of the information that can be viewed via the LED status display. The function of the LEDs, especially the NS and MS LEDs, is fieldbus-specific.

6.4.1 PROFINET

Below is a list of the information that can be viewed via the LED status display PROFINET.

LED	Designation	Color	Function
LOG	Supply logic	Green	**LED off:** No supply voltage logic present, or supply voltage logic is outside the operating range for the module.
			LED lights up green: Supply voltage logic is present.
PWR	Supply power	Green	**LED off:** No supply voltage power present, or supply voltage power is outside the operating range for the module.
			LED lights up green: Supply voltage power is present.
L/A2	Link/Activity 2: Network connection and network activity of port P2	Green	**LED off:** Connection inactive, communication inactive
			LED lights up green: Connection active, communication inactive
			LED flashes quickly: Connection active, communication active
L/A1	Link/Activity 1: Network connection and network activity of port P1	Green	**LED off:** Connection inactive, communication inactive
			LED lights up green: Connection active, communication inactive
			LED flashes quickly: Connection active, communication active
NS	Network status	Red/ Green	**LED off:** No connection to the control system available.
			LED lights up green: Connection to the control system present and control system is in "Run" mode.
			LED flashes green x1: Connection to the control system present and control system is in "Stop" mode. The IRT synchronization is not yet finished.
			LED flashes green continuously: The network participant is in identification mode.
			LED lights up red: Serious network error present.
			LED lights up red x1: The station name is not known.
			LED lights up red x2: The IP address is not known.
			LED lights up red x3: A configuration error is present.

LED	Designation	Color	Function
MS	Module status	Red/ Green	**LED off:** The product is in setup or NW_Init status.
			LED lights up green: The product is in normal operating mode.
			LED flashes green x1: The product is currently processing diagnostics processes.
			LED lights up red: Serious error. The product is not ready for operation.
			LED flashes alternately red and green: A firmware update is in progress. **WARNING Do not switch off the supply voltage otherwise the product may become damaged permanently.**

6.4.2 EtherNet/IP™

Below is a list of the information that can be viewed via the LED status display EtherNet/IP™.

LED	Designation	Color	Function
LOG	Supply logic	Green	**LED off:** No supply voltage logic present, or supply voltage logic is outside the operating range for the module.
			LED lights up green: Supply voltage logic is present.
PWR	Supply power	Green	**LED off:** No supply voltage power present, or supply voltage power is outside the operating range for the module.
			LED lights up green: Supply voltage power is present.
L/A2	Link/Activity 2: Network connection and network activity of port P2	Green	**LED off:** Connection inactive, communication inactive
			LED lights up green: Connection active, communication inactive
			LED flashes quickly: Connection active, communication active
L/A1	Link/Activity 1: Network connection and network activity of port P1	Green	**LED off:** Connection inactive, communication inactive
			LED lights up green: Connection active, communication inactive
			LED flashes quickly: Connection active, communication active
NS	Network status	Red/ Green	**LED off:** No supply voltage present and/or no IP address.
			LED lights up green: Product is online. One or more connections are/have been established (CIP™ Class 1 or 3)
			LED flashes green: Product is online, but has not yet established a connection.
			LED lights up red: Duplicate network address present. Serious network error present.
			LED flashes red: Timeout for one or more connections.

LED	Designation	Color	Function
MS	Module status	Red/ Green	**LED off:** No supply voltage present.
			LED lights up green: Controlled by a scanner in operating mode.
			LED flashes green: The product is not configured, scanner in sleep mode.
			LED lights up red: Serious error. The product is not ready for operation.
			LED flashes red: Removable malfunction/errors. The product is configured, but the stored parameters differ from the parameters currently in use.

6.4.3 EtherCAT

Below is a list of the information that can be viewed via the LED status display EtherCAT.

LED	Designation	Color	Function
LOG	Supply logic	Green	**LED off:** No supply voltage logic present, or supply voltage logic is outside the operating range for the module.
			LED lights up green: Supply voltage logic is present.
PWR	Supply power	Green	**LED off:** No supply voltage power present, or supply voltage power is outside the operating range for the module.
			LED lights up green: Supply voltage power is present.
L/A OUT	Link/Activity 2: Network connection and network activity of port P2	Green	**LED off:** Connection inactive, communication inactive
			LED lights up green: Connection active, communication inactive
			LED flashes quickly: Connection active, communication active
L/A IN	Link/Activity 1: Network connection and network activity of port P1	Green	**LED off:** Connection inactive, communication inactive
			LED lights up green: Connection active, communication inactive
			LED flashes quickly: Connection active, communication active

LED	Designation	Color	Function
RUN	RUN LED	Red/ Green	**LED off:** No supply voltage on and/or EtherCAT device in 'INIT' state.
			LED lights up green: EtherCAT-Device is in 'OPERATIONAL' state.
			LED flashes green: EtherCAT-Device is in 'PRE-OPERATIONAL' state.
			LED flashes single green: EtherCAT-Device is in 'OPERATIONAL' state.
			LED flickers: EtherCAT-Device is in 'BOOT' state.
			LED lights up red: If RUN LED (ST) and ERR LED (MS) light up red, this indicates a serious error. The bus interface has been put into a physically passive state. Contact SCHUNK Service.
ERROR	ERR LED	Red	**LED off:** No supply voltage present and/or no error.
			LED flashes red: Invalid configuration. The status change requested by the master is not possible due to invalid register or object settings.
			LED flashes single red: Unrequested status change. Device has changed the EtherCAT-status independently.
			LED flashes double red: Timeout of the Sync Manager Watchdog
			LED lights up red: If RUN LED (ST) and ERR LED (MS) light up red, this indicates a serious error. The bus interface has been put into a physically passive state. Contact SCHUNK Service.
			LED flickers: Boot error, e.g. due to a failed firmware download