Software Manual, Firmware 4
EGI with PROFINET, EtherCAT or EtherNet/IP™

Superior Clamping and Gripping

SCHUNK®

Imprint

Technical changes:

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

Document number: 1491302

Version: 02.00 | 25/04/2022 | 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

Customer Management

Tel. +49-7133-103-2503
Fax +49-7133-103-2189

cmg@de.schunk.com

Please read the operating manual in full and keep it close to the product.

Table of Contents

1 Changes from software release V2 to V4.

The form and structure of the software manual have essentially remained the same. The terminology has been adapted and the following corrections and additions have been made compared to software release V2.

- The logic and power voltage supply limits have been adjusted.
- Designations of the control and status bits have been revised.
- "Outward Referencing" has been added.
- A new gripping movement has been added with "Workpiece gripping at expected position".
- "Workpiece release" has been adapted due to the new gripping movement.
- Consecutive identical requests can be carried out again in a time-optimized manner.
- A handshake bit is used to report the receipt of a request.
- An active executed absolute positioning movement can be replaced by an absolute positioning movement with other movement parameters.
- Error detection of the position measuring system has been extended.
- Behavior when restoring factory settings has been improved.
- For modules with GKE, error monitoring of error events ERROR_MOTOR_VOLTAGE_LOW and ERROR_MOTOR_VOLTAGE_HIGH has been adapted.
- With ERROR_POWER_STAGE and ERROR_COMMUNICATION_LOST two new error events have been added.
- PROFINET GSDML breaks down the individual bits in the cyclic protocol for the control and status double word.
- PROFINET GSDML version has been increased to PNIO_Version V2.42.

- Error correction (bug fixing):
  - When re-gripping is terminated by "Workpiece release", "Absolute positioning movement" or "Relative positioning movement", the status bit "workpiece pre-grip started" is reset correctly.
  - Acyclically modified parameters are stored persistently each second.
  - In the event of a voltage drop, the warning message is displayed before the error message.
  - After the *EGI 40* has been reset to factory settings, the status bit "brake released" is no longer set.
  - The reason why restoring the factory settings can fail in some cases has been eliminated.
  - The maximum motor current is used for referencing
  - With the *EGI 40*, relative positioning can be performed correctly in both directions of movement.
  - Initializing the persistent memory has been improved in terms of failure probability.
  - Compliance check of PROFINET identification has been optimized.
  - Duplicate diagnostic memory entries in case of errors related to the encoder system have been fixed.
  - Status bit 4 "command successfully processed" is set and reset correctly when the "Remove workpiece manually" function is executed.
  - Incorrect display of error code after successful acknowledgment has been fixed.

2 General

2.1 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)

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 have the main version number 4.XX.

The firmware version can be read out. Information on the corresponding parameter can be found in section ▶ 5.2 [🗋 57].

Conventions

The following conventions apply to this software manual:

- The Intelligent Electric Gripper is referred to as the "module" in the following.
- 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
- Page number in references: [► 4]

The following abbreviations are used:

- GKE - maintenance of gripping force

Applicable documents

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

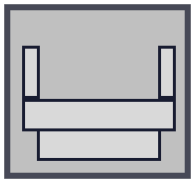
The documents labeled with an asterisk (*) can be downloaded from **schunk.com**.

2.2 Definitions and exact limits

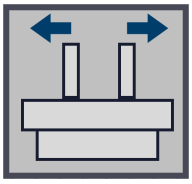
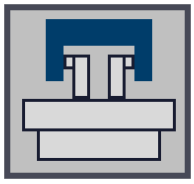
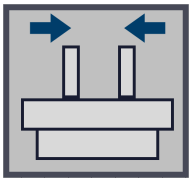
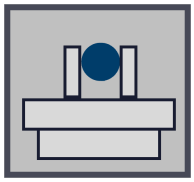
2.2.1 Mechanical stops, directions of movement and gripping modes

The definitions of the mechanical stops, directions of movement and gripping modes are shown below. These definitions refer to the positions and movement of the base jaws of the module.

Mechanical stops

Internal mechanical stop of the base jaws	External mechanical stop of the base jaws
The base jaws are opened on top of each other. 	The base jaws are at the maximum distance from each other. 

Direction of movement and gripping modes

Direction of movement	Gripping modes
The movement of the base jaws from the inner to the outer mechanical stop corresponds to the outward movement. 	If a workpiece is gripped from the inside during an *outward* movement, this corresponds to the *I.D. gripping* mode. 
The movement of the base jaws from the outer to the inner mechanical stop corresponds to the inward movement. 	If a workpiece is gripped from the outside during an *inward* movement, this corresponds to the *O.D. gripping* mode. 

2.2.2 Hardware and software limits

Hardware limits

The *inner* hardware limit corresponds to the application-specific mechanical end stop, which is achieved by a complete inward movement.

The *outer* hardware limit corresponds to the application-specific mechanical end stop, which is achieved by a complete outward movement.

NOTE

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

The module's own hardware limits correspond to the inner and outer mechanical stops of the base jaws.

Software limits

The software limits correspond to the limits of a "virtual" range within which movements are permitted.

The *inner* software limit corresponds to the smallest position value that can be approached in an application-specific manner.

The *outer* software limit corresponds to the largest position value that can be approached in an application-specific manner.

The software limits are only monitored in the referenced module state and may be parameterized by the user, ▶ 5.2 [🗋 47].

CAUTION

Material damage possible due to incorrect parameterization!

If the module is used in an application in which the base jaws *cannot* be moved between the inner and/or outer mechanical stops, the software limits *must* be adapted to the application. Failure to adjust the software limits may result in damage or destruction of the module.

- Adjust software limits.

2.2.3 Reference point and zero point

Reference point

A reference point corresponds to a unique, reproducible position that can be approached by the base jaws. The inner reference point corresponds to the inner hardware limit.

Zero point

The zero point corresponds to the absolute reference point to which the position of the base jaws of the module relates. Without changing the factory settings of the module, the zero point corresponds to the inner reference point.

2.2.4 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 ▶ 5.2 [🗋 47].

Value	Minimum value	Maximum value
Ambient temperature	5°C	55°C
Logic voltage supply	19.2 V	30 V
Power voltage supply	19.2 V	30 V
Factory setting software limits	0 mm	*EGI 40:* 80 mm *EGI 80:* 115 mm
Positioning speed	15 mm/s	*EGI 40:* 100 mm/s *EGI 80:* 200 mm/s
Gripping force	25 N	*EGI 40:* 70 N *EGI 80:* 100 N

3 Communication

3.1 Data exchange

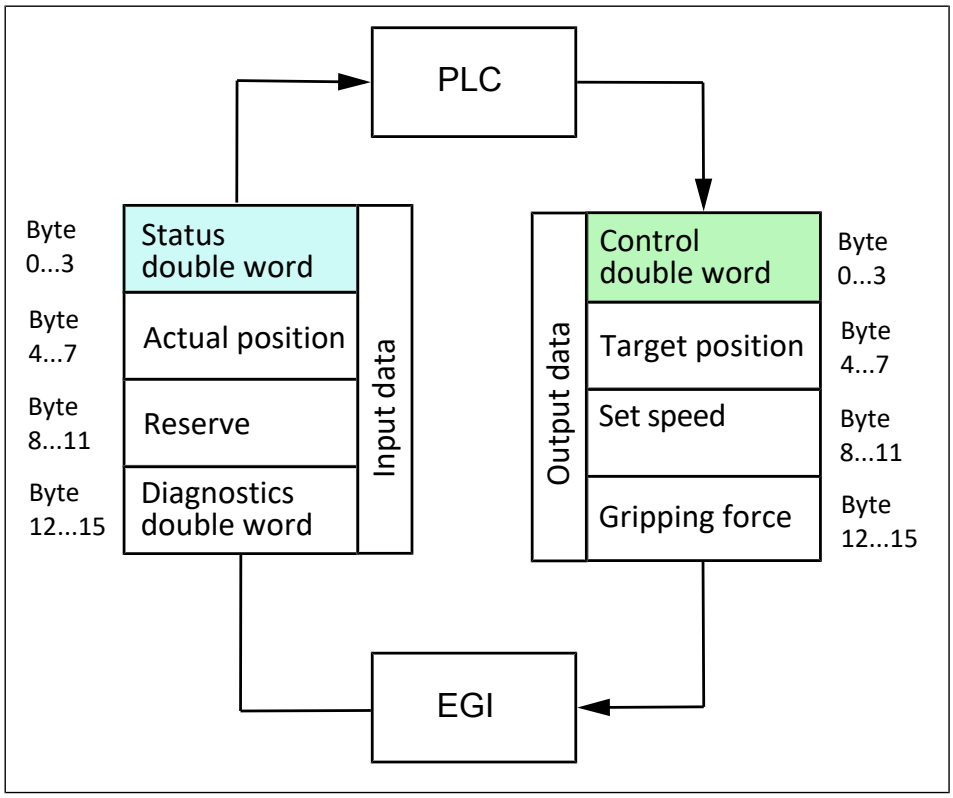
Via integrated fieldbus interfaces, data can be exchanged cyclically and acyclically between the module and the controller.

The following fieldbus interfaces are available:

- PROFINET
- EtherCAT
- EtherNet/IP™

3.1.1 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.

3.1.1.1 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 ▶ 7.1 [🗋 68].

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", ▶ 7.4 [🗋 89]-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.

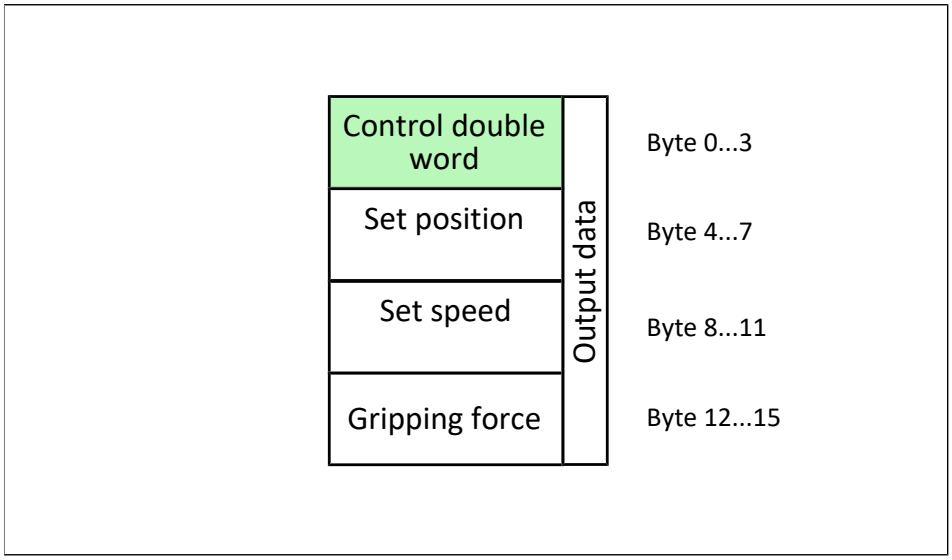
  An immediate transition between active movements of the module is only permitted with absolute position movement, and otherwise leads to the controlled termination of the current active movement.
- The transmitted bit combination, in particular the control double word, is impermissible.

  For a detailed description of impermissible bit combinations, see chapter ▶ 7.3 [🗋 84].
- At least one movement parameter that has been transmitted is impermissible.

  The value of a movement parameter is considered to be impermissible if it is outside the permitted minimum or maximum limits. See Chapter ▶ 2.2 [🗋 8]

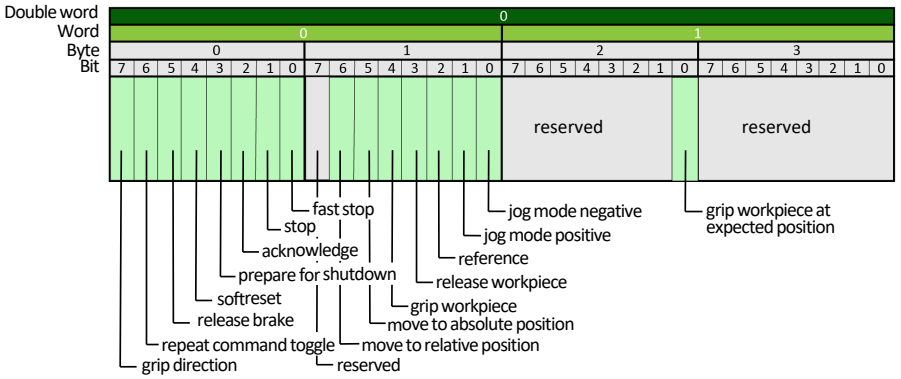
Data frame

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 ▶ 7.3 [🗋 84].

Word	Byte	Bit	Cyclical output data
0	1	0	fast stop [🗋 84]
		1	stop [🗋 84]
		2	acknowledge [🗋 85]
		3	prepare for shutdown [🗋 85]
		4	softreset [🗋 85]
		5	release brake [🗋 86]
		6	repeat command toggle [🗋 86]
		7	grip direction [🗋 86]
	2	8	jog mode negative [🗋 86]
		9	jog mode positive [🗋 87]
		10	reference [🗋 87]
		11	release workpiece [🗋 87]
		12	grip workpiece [🗋 87]
		13	move to absolute position [🗋 88]
		14	move to relative position [🗋 88]
		15	reserved

Word	Byte	Bit	Cyclical output data
1	3	16	grip workpiece at expected position [🗋 88]
		17	reserved
		18	reserved
		19	reserved
		20	reserved
		21	reserved
		22	reserved
		23	reserved
	4	24	reserved
		25	reserved
		26	reserved
		27	reserved
		28	reserved
		29	reserved
		30	reserved
		31	reserved

Target position

- In bytes 4 – 7 of the cyclical output data, data is transmitted that is used for positioning purposes, ▶ 5.2 [🗋 47].
- The data format of the parameter is a signed integer with 32 bits and represents a value in microns [µm] (1000 µm ≙ 1 mm).

Target speed

- In bytes 8 – 11 of the cyclical output data, the value of the set speed of a movement is transmitted, ▶ 5.2 [🗋 47].
- The data format of the parameter is a signed integer with 32 bits and represents a value in microns per second [µm/s] (1000 µm/s ≙ 1 mm/s).

Gripping force

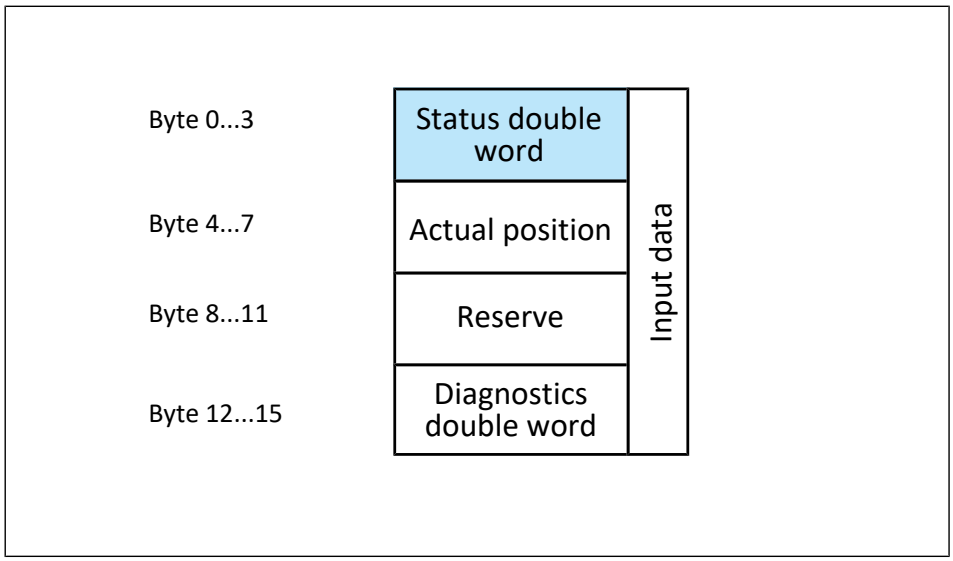
- In bytes 12 – 15 of the cyclical output data, the gripping force with which a workpiece is to be gripped is transmitted, ▶ 4.3.1 [🗋 34].
- The data format of the parameter is a signed integer with 32 bits and represents a value in millinewtons [mN] (1000 mN ≙ 1 N).

3.1.1.2 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.

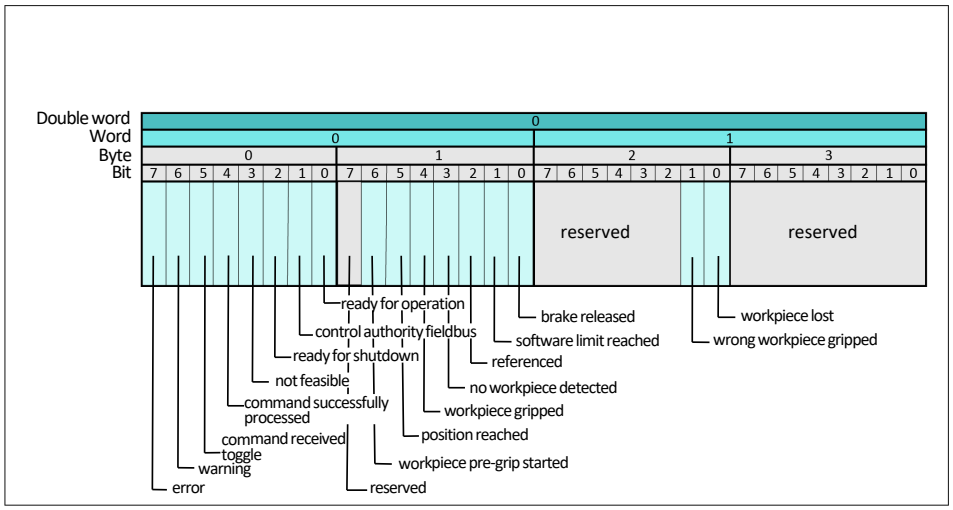
Data frame

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 ▶ 7.4 [🗋 89].

NOTE

In the "Cyclical input data" column of the following table, the designations of the status bits are shown as follows:

- Long English designation
- Short English designation
- Short German designation

✓ The long designation increases comprehensibility when reading this manual. The short designations (EN - short, DE - short) are used for the internal web application.

Word	Byte	Bit	Cyclical input data
0	0	0	ready for operation [🗋 89] EN - short: ready for op DE - short: Betriebsbereit
		1	control authority fieldbus [🗋 89] EN - short: ctrl authority fb DE - short: Feldbus
		2	ready for shutdown [🗋 89] EN - short: ready for sd DE - short: Abschaltbereit
		3	not feasible [🗋 89] EN - short: not feasible DE - short: Nicht durchführb.
		4	command successfully processed [🗋 89] EN - short: cmd success DE - short: Kdo. erfolgreich
		5	command received toggle [🗋 90] EN - short: cmd rcvd tgl DE - short: Kommandowechsel
		6	warning [🗋 90] EN - short: warning DE - short: Warnung
		7	error [🗋 90] EN - short: error DE - short: Fehler

Word	Byte	Bit	Cyclical input data
0	1	8	brake released [🗋 90] EN - short: brake released DE - short: Bremse offen
		9	software limit reached [🗋 90] EN - short: softlimit reached DE - short: Softlimit
		10	referenced [🗋 90] EN - short: referenced DE - short: Referenziert
		11	no workpiece detected [🗋 90] EN - short: no wp detected DE - short: Kein Werkstück
		12	workpiece gripped [🗋 90] EN - short: wp gripped DE - short: Gegriffen
		13	position reached [🗋 91] EN - short: pos reached DE - short: Positioniert
		14	workpiece pre-grip started [🗋 91] EN - short: wp pre-grip started DE - short: Nachgreifen
		15	reserved
1	2	16	workpiece lost [🗋 91] EN - short: wp lost DE - short: Werkst. verloren
		17	wrong workpiece gripped [🗋 91] EN - short: wrong wp gripped DE - short: Falsches Werkst.
		18	reserved
		19	reserved
		20	reserved
		21	reserved
		22	reserved
		23	reserved

Word	Byte	Bit	Cyclical input data
1	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, ▸ 5 [🗋 47].
- 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.

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.

Mix-ups are impossible with diagnostic codes (warning and error codes), since each of these codes is assigned only once.

- 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, ▸ 6 [🗋 59].
- 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, ▸ 6 [🗋 59].

NOTE

Based on the DS301 communication profile, the module implements "CANopen over EtherCAT" in the **EtherCAT** variant. 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 cyclical output data are transmitted in a RPDO (Receive PDO) and the input data in a TPDO (Transmit PDO).

3.1.2 Acyclical data exchange

3.1.2.1 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 ▶ 5 [🗋 47].

3.1.2.2 EtherCAT

The implementation of the acyclic data exchange corresponds to the EtherCAT specification of the CANopen-specific communication profile DS301. 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 (▶ 5.2 [🗋 47]) 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**.

3.1.2.3 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 ▶ 5 [🗋 47]
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 ▶ 5 [🗋 47]
Attributes:	see the following table "Instance Attributes"
Source Element:	user-defined day
Source Length:	Length of the data to be written
Communication:	Set path to the desired SCHUNK device

Instance Attributes

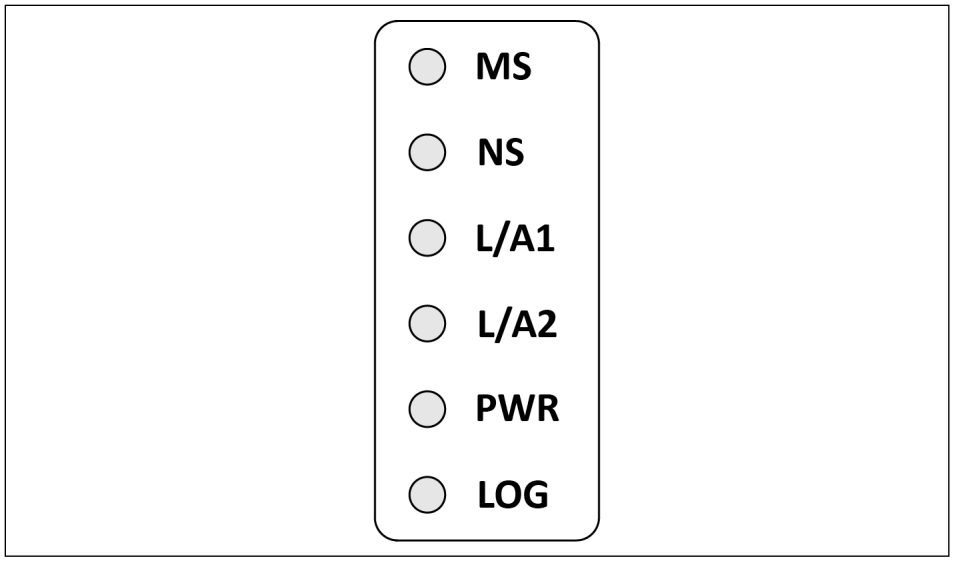
#	Name	Access	Type	Value / description	
1	Name	Get	SHORT_STRING	Parameter name (incl. length)	
2	ABCC data type	Get	Array of UINT	Data type of the instance value	
3	No. of Elements	Get	UINT	Number of elements for the specified data type	
4	Descriptor	Get	Array of UINT	Bit that describes the access rights 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.

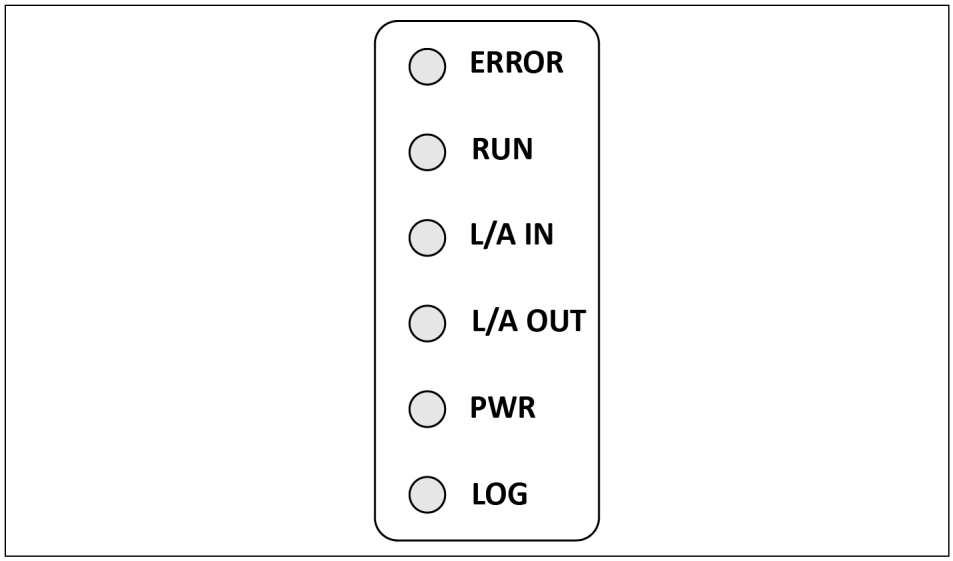
For all the required information pertaining to acyclic data exchange, see chapter ▶ 5 [🗋 47].

3.2 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 ▶ 7.5 [🗋 92].

3.3 Web application

The web application can be used for the commissioning, parameterization and diagnostics of modules.

NOTE

Only modules with the fieldbus interfaces PROFINET and EtherNet/IP™ are delivered with an integrated web application. The web application is **not** available for modules with a EtherCAT interface.

3.3.1 Access

The following prerequisites must be met before a mobile device can access the web application:

- The web application must be installed on the module.
- The module has a valid IP address.
- The IP addresses of the module and the end device must be in the same network.

If the prerequisites are met, the web application can be accessed with a browser via "http://IPADRESSE". The "IPADDRESS" in this case is the IP address of the module in the Ethernet network.

NOTE

The "Anybus-IP Config Tool" (found in zipped file: HMS IPconfig - Utility for module TCP/IP configuration) can be used to assign an IP address to a module or to determine the IP address of a module. This tool can be downloaded from "https://www.anybus.com". For a detailed description of downloading the tool see chapter ▶ 7.6 [🗋 96].

3.3.2 Range of functions

The web application provides the following interactive application options:

- Movements can be triggered.
- Parameters can be displayed and/or changed.
- Parameter lists can be downloaded.
- The entries of the diagnostic memory can be displayed.
- The contents of the diagnostic memory can be downloaded as a list.
- The factory settings of the module can be restored.
- The Assembly and Operating Manual as well as the software manual can be downloaded.
- The device description file can be downloaded.

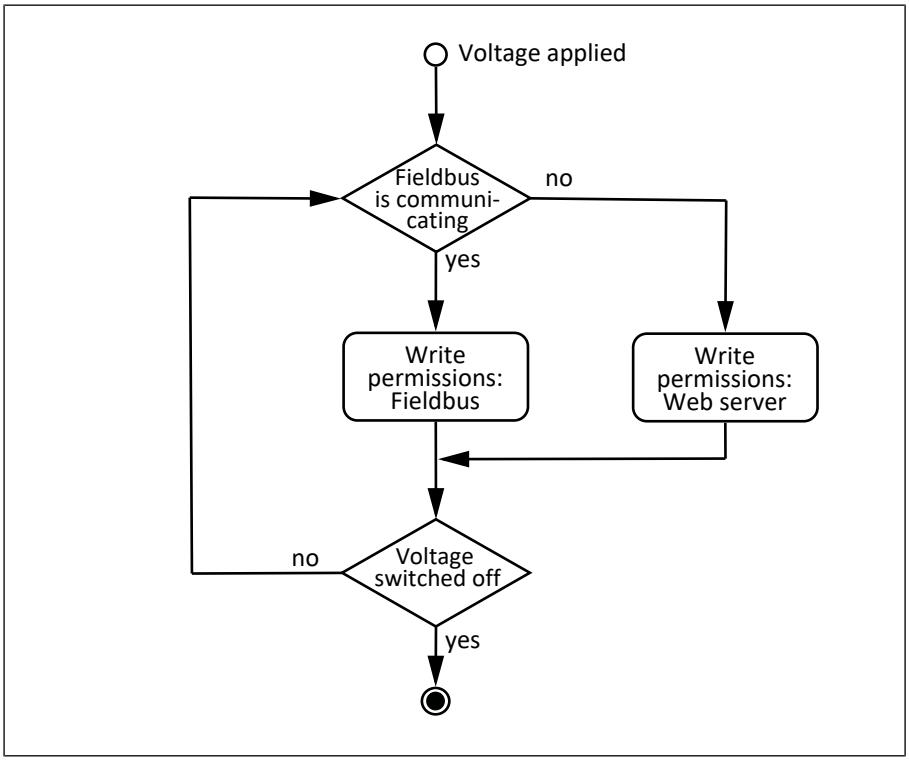
3.4 Management of control logic

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 "control authority fieldbus".



Management of the control logic

4 Module functions

4.1 Booting, shutting down and restarting the module

4.1.1 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, ▶ 4.1.3 [🗋 26].

NOTE

In order to exclude unexpected behavior after booting, the module sets all control bits to status 1 during booting. This prevents unwanted requests from being triggered by the external controller during the ongoing cyclical transmission of the control word. Only the status bit "command received toggel" reflects the inverted value of the initially received control bit "fast stop".

Background: If the controller transmits an 0 in "fast stop" during or after booting, this 1 -> 0 transition is interpreted as a request for a "fast stop" and accordingly "command received toggel" is set from the start value 0 to 1.

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).	ERR_UNKNOWN_HW
The internal memory is not recognized (anymore).	ERR_NO_BLOCK_DEV
The communication module is not recognized (anymore).	ERR_NO_COMM

* For further information, see chapter ▶ 6 [🗋 59].

4.1.2 Shutting down

Short description

When the module is switched off in a controlled manner, data is stored persistently (permanently), e.g. the "reference" of the module zero point.

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", ▶ 7.3 [🗋 85]-Bit3.

Module feedback

- The receipt of the request is immediately displayed by a change of the status bit "command received toggel". This confirmation is made irrespective of whether the request is subsequently completed successfully or can be processed at all.
- If preparations to shut down the module are *successful*, this is displayed by setting the status bit "ready for shutdown".
- If preparations to shut down the module are *not successful*, this is displayed by setting the status bit "error" and the corresponding diagnostic code.

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.

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.	ERR_SD_FAILED

* For further information, see chapter ▶ 6 [🗋 59].

4.1.3 Restarting

Short description

A restart of the module can be triggered from within a defined system state on the software side. During a restart, data is stored persistently (permanently), analogous to the preparation for shutdown, see chapter ▶ 4.1.2 [🗋 25].

Trigger

The restart is triggered by a setting of the control bit "softreset", ▶ 7.3 [🗋 85]-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, ▶ 5.2 [🗋 57].

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.

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.	ERR_SD_FAILED

* For further information, see chapter ▶ 6 [🗋 59].

4.2 Movement functions

4.2.1 Tip mode

Short description

In jog mode, the module executes a movement to the outside or inside.

Trigger

- Jog mode toward the outside is triggered by setting the control bit "jog mode positive", ▶ 7.3 [🗋 87]-Bit9.
- Jog mode toward the inside is triggered by setting the control bit "jog mode negative", ▶ 7.3 [🗋 86]-Bit8.

NOTE

The origin of the designation of the control bits lies in the fact that the outward movement is considered to be a movement in the positive direction and the inward movement a movement in the negative direction.

Movement parameter

No movement parameters need to be transmitted cyclically in order to carry out jog mode.

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

Finish

Jog mode is terminated by the following events:

- Resetting the control bit "jog mode positive" or "jog mode negative"
- Setting the control bit "stop"

NOTE

For modules with GKE: The brake is only applied following a short delay time after resetting the control bit "jog mode positive" or "jog mode negative". This makes direct reversal in jog mode possible, without the brake having to switch several times. Other requests before this delay time expires are not permitted, ▶ 3.1.1.1 [🗋 12].

Module feedback

- The receipt of the request is immediately displayed by a change of the status bit "command received toggel". This confirmation is made irrespective of whether the request is subsequently completed successfully or can be processed at all.
- If the module is actively in jog mode, this is displayed by setting the status bit "command successfully processed".

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.	ERR_SOFT_LOW
The upper software limit is reached in the referenced state.	ERR_SOFT_HIGH
Drive is already blocked at the start of movement.	ERR_MOVE_BLOCKED
Drive blocked during movement.	ERR_MOVE_BLOCKED
Sending a request that is not permitted.	WRN_NOT_FEASIBLE
Movement terminated by user.	ERR_FAST_STOP

* For further information, see chapter ▶ 6 [🗋 59].

4.2.2 Referencing

Short description

The module defines its zero point during referencing, ▶ 2.2.3 [🗋 10].

Trigger

Referencing is triggered by setting the "reference" control bit, ▶ 7.3 [🗋 87]-Bit 10.

If the control bit "reference" is set, new referencing can be triggered by changing the control bit "repeat command toggle", ▶ 7.3 [🗋 86]-Bit 6.

Movement parameter

No movement parameters need to be transmitted cyclically to carry out referencing.

NOTE

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

Parameterization

The <ref_type> (▶ 5.2 [🗋 51]) parameter can be used to parameterize the direction of movement in which referencing is to be executed.

Finish

Referencing is terminated by the following events:

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

Module feedback

- The receipt of the request is immediately displayed by a change of the status bit "command received toggel". This confirmation is made irrespective of whether the request is subsequently completed successfully or can be processed at all.
- Successful referencing is displayed by setting the status bit "referenced".
- Unsuccessful referencing is displayed by an error code

NOTE

When a new reference movement is initiated, any status bit that may be set "referenced" is then reset.

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.	ERR_NO_REF
The referencing point cannot be found.	ERR_NO_REF
Sending a request that is not permitted.	WRN_NOT_FEASIBLE
Movement terminated by user.	ERR_FAST_STOP

* For further information, see chapter ▶ 6 [🗋 59].

4.2.3 Absolute positioning movement

Short description

During absolute positioning, the module executes a positioning movement relative to the zero point of the module. A practical application example is described in chapter ▶ 7.1 [🗋 70], example 1.

NOTE

Using a positioning movement for gripping workpieces represents a misuse, which will result in a module error. Absolute positioning is only permitted within the software limits, ▶ 2.2.2 [🗋 9].

Trigger

Absolute positioning is only permitted in the referenced status and is triggered by setting the control bit "move to absolute position", ▶ 7.3 [🗋 88]-Bit 13.

If the control bit "move to absolute position" is set, new absolute positioning can be triggered by changing the control bit "repeat command toggle", ▶ 7.3 [🗋 86]-Bit 6.

Movement parameter

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

- Position parameter
- Speed parameter

Finish

Absolute positioning is terminated by the following events:

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

Module feedback

- The receipt of the request is immediately displayed by a change of the status bit "command received toggel". This confirmation is made irrespective of whether the request is subsequently completed successfully or can be processed at all.
- Once the target position has been reached, it is displayed by setting the status bit "position reached".

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.	ERR_MOV_ABORT_TO
Lower software limit is reached.	ERR_SOFT_LOW
Upper software limit is reached.	ERR_SOFT_HIGH
Drive is already blocked at the start of movement.	ERR_MOVE_BLOCKED
Drive blocked during movement.	ERR_MOVE_BLOCKED

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

* For further information, see chapter ▶ 6 [🗋 59].

4.2.4 Relative positioning movement

Short description

During relative positioning, the module executes a positioning movement relative to the current actual position. A practical application example of this is described in chapter ▶ 7.1 [🗋 70], Example 2.

NOTE

Using a positioning movement for gripping workpieces represents a misuse, which will result in a module error.

Relative positioning is only permitted within the software limits, ▶ 2.2.2 [🗋 9].

Trigger

Relative positioning is only permitted in a referenced status and is triggered by setting the control bit "move to relative position", ▶ 7.3 [🗋 88]-Bit 14.

If the control bit "move to relative position" is set, new relative positioning can be triggered by changing the control bit "repeat command toggle", ▶ 7.3 [🗋 86]-Bit 6.

Movement parameter

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

- Position parameter
- Speed parameter

Finish

Relative positioning is terminated by the following events:

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

Module feedback

- The receipt of the request is immediately displayed by a change of the status bit "command received toggel". This confirmation is made irrespective of whether the request is subsequently completed successfully or can be processed at all.
- Once the target position has been reached, it is displayed by setting the status bit "position reached".

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.	ERR_MOV_ABORT_TO
Lower software limit is reached.	ERR_SOFT_LOW
Upper software limit is reached.	ERR_SOFT_HIGH

Diagnostic event	Diagnostic code *
Drive is already blocked at the start of movement.	ERR_MOVE_BLOCKED
Drive blocked during movement.	ERR_MOVE_BLOCKED
Sending a request that is not permitted.	WRN_NOT_FEASIBLE
Movement terminated by user.	ERR_FAST_STOP

* For further information, see chapter ▶ 6 [🗋 59].

4.2.5 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", ▶ 7.3 [🗋 84]-Bit 1.

If the control bit "stop" is set, a new controlled stop can be triggered by a change of the control bit "repeat command toggle", ▶ 7.3 [🗋 86]-Bit 6. In this case, however, no active movement is triggered, since no active movement is carried out at this time.

Movement parameter

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

Finish

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

Module feedback

- The receipt of the request is immediately displayed by a change of the status bit "command received toggel". This confirmation is made irrespective of whether the request is subsequently completed successfully or can be processed at all.
- The controlled stop of an active movement is displayed by setting the status bit "position reached".

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	ERR_MOV_ABORT_TO
Sending a request that is not permitted	WRN_NOT_FEASIBLE
Movement terminated by user	ERR_FAST_STOP

* For further information, see chapter ▶ 6 [🗋 59].

4.2.6 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", ▶ 7.3 [🗋 84]-Bit 0.

Module feedback

- The receipt of the request is immediately displayed by a change of the status bit "command received toggel". This confirmation is made irrespective of whether the request is subsequently completed successfully or can be processed at all.
- Since the control bit "fast stop" is executed in a fail-safe and thus "low-active" manner, the cancellation of an active movement is displayed by the reset (1 -> 0).

4.3 Handling a workpiece

During workpiece handling, the module executes the following sub-actions one after the other after a gripping command has been issued:

1. Gripping
2. Re-gripping workpiece (optional)
3. Holding workpiece incl. workpiece loss detection
4. Releasing workpiece

4.3.1 Workpiece gripping (simple gripping movement)

Short description

The module is able to grip workpieces with a simple gripping movement. A practical application example is described in chapter ▶ 7.1 [🗋 71], example 3 – 4.

NOTE

Depending on the parameterization, the module changes after a successful simple gripping process to the re-gripping workpiece [🗋 39] or holding workpiece [🗋 41] status.

The simple gripping movement is only permitted within the software limits, ▶ 2.2.2 [🗋 9].

Trigger

The simple gripping movement is only permitted in a referenced status and is triggered by setting the control bit "grip workpiece", ▶ 7.3 [🗋 87]Bit 12.

Movement parameter

In order to carry out a simple gripping movement, the following movement parameters and information must be transmitted cyclically to the module:

- Gripping force
- Gripping direction
  The gripping direction is to be defined by the control bit "grip direction", ▶ 7.3 [🗋 86]-Bit 7.

Finish

The simple gripping movement is terminated by the following events:

- Workpiece successfully gripped
- Automatic switchover to re-gripping
- Software limit has been reached
- Setting the control bit "stop"

Module feedback

- The receipt of the request is immediately displayed by a change of the status bit "command received toggel". This confirmation is made irrespective of whether the request is subsequently completed successfully or can be processed at all.
- The automated change to re-gripping is displayed by setting the status bit "workpiece pre-grip started".
- A successful gripping movement is displayed by setting the status bit "workpiece gripped".
- An unsuccessful gripping movement is displayed by setting the status bit "no workpiece detected".

Possible diagnostic events

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

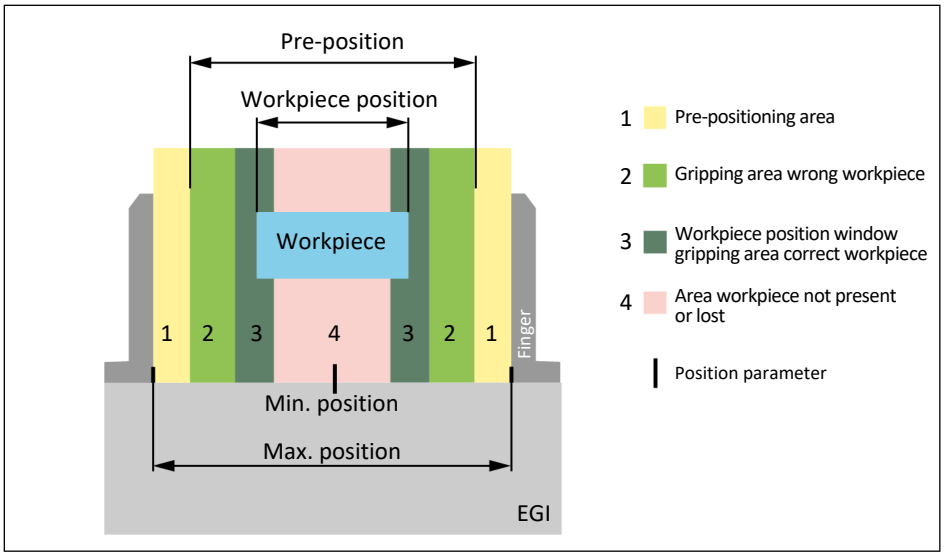
Diagnostic event	Diagnostic code *
Sending a request that is not permitted.	WRN_NOT_FEASIBLE
Movement terminated by user.	ERR_FAST_STOP

* For further information, see chapter ▶ 6 [🗋 59].

4.3.2 Workpiece gripping at expected position (combined gripping movement)

Short description

The module can grip workpieces at an expected position with a combined gripping movement.



Gripping workpiece at expected position, example of O.D. gripping

NOTE

The example illustrated shows the O.D. gripping mode, but the statements made also apply to the I.D. gripping mode.

The combination consists of an optional pre-positioning (Fig.: yellow area 1) and the gripping movement (Fig.: light and dark green areas 2 and 3).

The decision as to whether the correct or wrong workpiece has been gripped is made on the basis of the gripping position detected:

- The *correct* workpiece is gripped within the workpiece position window (Fig.: dark green area 3).
- The *wrong* workpiece is gripped between the pre-position and the workpiece position window (Fig.: light green area 2).

The workpiece position window is a "virtual window" that is clamped around the expected workpiece position. If the workpiece position window is exceeded (Fig. light red area 4), no workpiece has been

detected or the workpiece was lost during re-gripping. Practical application examples are described in chapter Application examples [🗋 71], examples 5 – 9.

NOTE

The pre-position and the workpiece position window are calculated from the cyclically transferred workpiece position and the parameterization of the module.

If the start position lies between the pre-position and the gripping position when the gripping movement is triggered, the pre-positioning is dispensed with and the gripping movement is carried out immediately.

Depending on the parameterization, the module changes after a successful combined gripping process to the re-gripping workpiece [🗋 39] or holding workpiece [🗋 41] status. The combined gripping movement is only permitted within the software limits, ▸ 2.2.2 [🗋 9].

Trigger

The combined gripping movement is only permitted in a referenced status and is triggered by setting the control bit "grip workpiece at expected position" ,▸ 7.3 [🗋 88]-Bit 16.

Movement parameter

In order to carry out the combined gripping movement, the following movement parameters and information must be transmitted cyclically to the module

- Position parameter
  In this case, the position parameter corresponds to the workpiece position at which the workpiece is expected.
- Speed parameter
  In this case, the speed parameter corresponds to the maximum speed for approaching the pre-position.
- Gripping force
- Gripping direction
  The gripping direction is to be defined by the control bit "grip direction", ▸ 7.3 [🗋 86]-Bit 7.

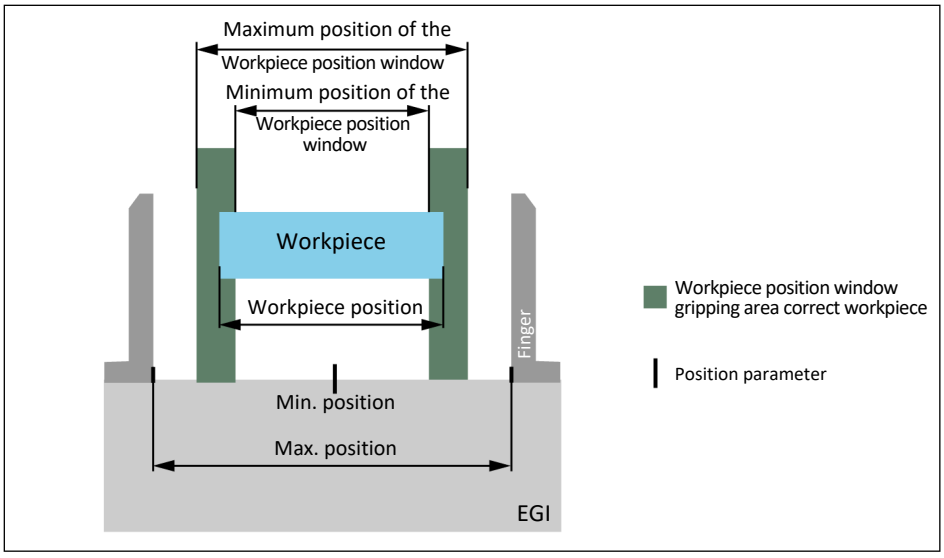
Parameterization

Minimum and maximum position

The parameter <grp_pos_margin> (▶ 5.2 [🗋 51]) can be used to parameterize the value from which the minimum and maximum positions of the workpiece position window are calculated.

NOTE

- The minimum position of the workpiece position window is calculated according to: workpiece position **-** <grp_pos_margin>.
- The maximum position of the workpiece position window is calculated according to: workpiece position **-** <grp_pos_margin>.



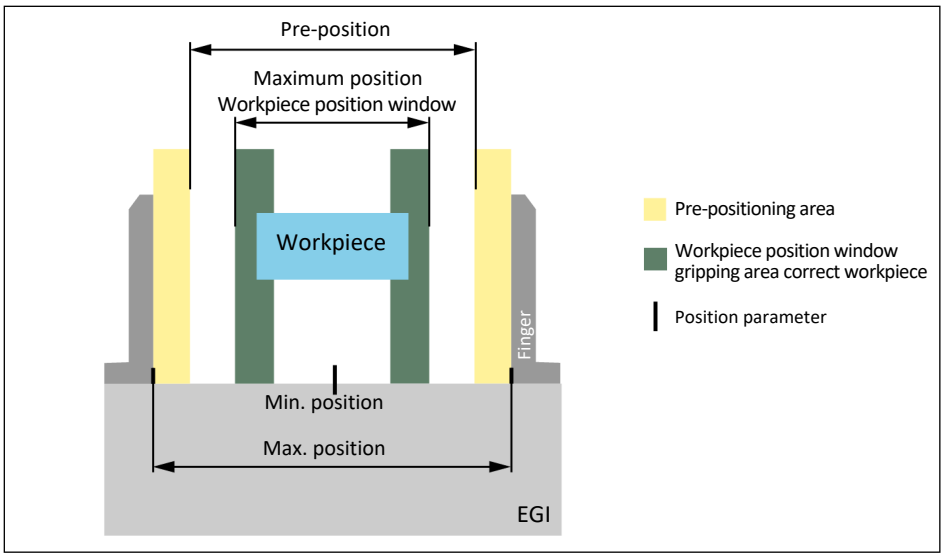
Workpiece position window for O.D. gripping, minimum and maximum position

Pre-position

The parameter <grp_prepos_delta> (▶ 5.2 [🗋 51]) can be used to parameterize the difference in position between the workpiece position window and the pre-position.

NOTE

- The pre-position is calculated from the minimum or maximum position of the workpiece position window depending on the direction from which a workpiece is gripped.
- The pre-position during I.D. gripping is calculated according to: minimum position workpiece position window **-** <grp_prepos_delta>.
- The pre-position during O.D. gripping is calculated according to: maximum position workpiece position window **+** <grp_prepos_delta>



Pre-positioning area for O.D. gripping

Finish

The gripping of workpieces is terminated by the following options:

- Expected workpiece was gripped
- Unexpected workpiece was gripped
- Automatic switchover to re-gripping
- Gripping position was exceeded
- Software limit has been reached
- Setting the control bit "stop"

Module feedback

- The receipt of the request is immediately displayed by a change of the status bit "command received toggel". This confirmation is made irrespective of whether the request is subsequently completed successfully or can be processed at all.
- The automated change to re-gripping is displayed by setting the status bit "workpiece pre-grip started".
- Gripping of the expected workpiece is displayed by setting the status bit "workpiece gripped".
- Gripping of an unexpected workpiece is displayed by setting the status bit "wrong workpiece gripped".
- Exceeding the workpiece position window is displayed by setting the status bit "no workpiece detected".

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.	WRN_NOT_FEASIBLE
Movement terminated by user.	ERR_FAST_STOP
The drive is blocked during pre-positioning.	ERR_MOVE_BLOCKED

* For further information, see chapter ▶ 6 [🗋 59].

4.3.3 Workpiece re-gripping

Short description

In order to prevent the loss of a workpiece during the gripping movement, "re-gripping" 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. Practical application examples are described in chapter ▶ 7.1 [🗋 79], examples 21 – 23.

Trigger

With the appropriate parameterization, re-gripping is carried out automatically after a gripping movement. No further information has to be transmitted cyclically to the module.

Parameterization

The parameter <grp_prehold_time> (▶ 5.2 [🗋 50]) can be used to parameterize the time span of the re-gripping. The maximum time span for re-gripping is 5000 ms.

NOTE

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

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 workpiece"
- Setting the control bit "move to absolute position"
- Setting the control bit "move to relative position"

NOTE

If re-gripping is interrupted by setting the control bit "stop", workpiece loss is to be assumed as the re-gripping was not successfully completed. This is displayed by setting the status bit "workpiece lost". Furthermore, the status bit "workpiece pre-grip started" is reset.

Module feedback

- The start of re-gripping is displayed by setting the status bit "workpiece pre-grip started".

Feedback after a previous simple gripping movement:

- Gripping of a workpiece is displayed by setting the status bit "workpiece gripped".
- Unsuccessful re-gripping is displayed by setting the status bit "workpiece lost".

Feedback after a previous combined gripping movement:

- Gripping of the expected workpiece is displayed by setting the status bit "workpiece gripped".
- Gripping of an unexpected workpiece is displayed by setting the status bit "wrong workpiece gripped".
- Exceeding the workpiece position window 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.	WRN_NOT_FEASIBLE
Movement terminated by user.	ERR_FAST_STOP

* For further information, see chapter ▶ 6 [🗋 59].

4.3.4 Holding workpiece incl. workpiece loss detection

Short description

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

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

Trigger

Holding the workpiece is performed automatically after gripping a workpiece. No additional information needs to be transferred to the module.

Module feedback

- *For modules with GKE:* A workpiece loss can only be detected before the gripping force retention is activated. This is indicated by setting the status bit "workpiece lost".
- *Modules with GKE:* A workpiece loss is displayed by setting the status bit "workpiece lost".

NOTE

For modules with GKE: For technical reasons, a workpiece loss cannot be detected after gripping force retention has been activated and therefore cannot be displayed.

The status bits relating to workpiece handling are not reset when an error occurs. These include the status bits "workpiece gripped", "wrong workpiece gripped" and "workpiece pre-grip started".

4.3.5 Workpiece release

Short description

When releasing workpieces, the module executes a relative positioning movement to the release position. The release position is calculated by the module and depends on the parameterization of the module and the previous gripping movement. Practical application examples are described in chapter ▶ 7.1 [🗋 74], examples 10 – 12.

NOTE

- After a simple gripping movement, the parameter <wp_release_delta> (▶ 5.2 [🗋 51]) is used to calculate the release position.
- After a combined gripping movement, the parameters <wp_release_delta> (▶ 5.2 [🗋 51]) and <grp_pos_margin> (▶ 5.2 [🗋 51]) are used to calculate the release position. This makes it possible to initiate a combined gripping movement again directly from the release position.

Trigger

Releasing workpieces is only permitted after a successful gripping movement and is triggered by setting the control bit "release workpiece", ▶ 7.3 [🗋 87] - Bit11.

NOTE

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

Movement parameter

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

Parameterization

The parameter <wp_release_delta> (▶ 5.2 [🗋 51]), can be used to parameterize the distance which the module moves relatively during release.

The parameter <grp_pos_margin> (▶ 5.2 [🗋 51]) can be used to parameterize the tolerance value from which the workpiece position window is calculated.

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.

Module feedback

- The receipt of the request is immediately displayed by a change of the status bit "command received toggel". This confirmation is made irrespective of whether the request is subsequently completed successfully or can be processed at all.
- The release of workpieces is displayed by setting the status bit "position reached".

NOTE

All corresponding status bits that displayed that a workpiece was gripped are reset.

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.	ERR_MOV_ABORT_TO
Drive is already blocked at the start of movement.	ERR_MOVE_BLOCKED
Drive blocked during movement.	ERR_MOVE_BLOCKED
Sending a request that is not permitted.	WRN_NOT_FEASIBLE
Movement terminated by user.	ERR_FAST_STOP

* For further information, see chapter ▶ 6 [🗋 59].

4.3.6 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!

Trigger

The manual removal of workpieces is triggered by setting the control bit "release brake", ▶ 7.3 [🗋 86]-Bit 5.

If the control bit "release brake" is set, a new manual removal of workpieces can be triggered by changing the control bit "repeat command toggle", ▶ 7.3 [🗋 86]-Bit 6.

No additional information needs to be transferred to the module.

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"

Module feedback

The receipt of the request is immediately displayed by a change of the status bit "command received toggel". This confirmation is made irrespective of whether the request is subsequently completed successfully or can be processed at all.

4.4 Additional functions

4.4.1 Handshake

Short description If a request is detected by the module, this is reported back to the outside.

Module feedback The receipt of a request is displayed by toggling the status bit "command received toggel".

4.4.2 Repeat request with optimized time

Short description This function makes it possible to send consecutive identical requests to the module in a time-optimized manner. Practical application examples are described in chapter ▶ 7.1 [🗋 82], Example 24.

NOTE

The standard procedure is to send requests to the module by 0 -> 1 edges of individual control bits. If the same function is to be carried out again, the corresponding control bit must first be reset and then set again.

Trigger If the control bit remains set, the time-optimized transmission of identical requests is triggered by toggling the control bit "repeat command toggle".

4.4.3 Position maintenance

After completing a movement, the module automatically changes to the position maintenance status. Movements that end in workpiece gripping are an exception.

Modules with GKE: The brake is activated and the motor control is switched off with a short delay.

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

4.4.4 Factory settings

Short description

The module can be reset from the error state to the factory settings using the software. All non-module-specific parameters are reset to the respective default values or to default settings. In addition, the diagnostic memory is deleted.

NOTE

When the first movement is started after resetting to factory settings, the module must restore part of the internal parameterization. As a result, there may be minimal time delays between the triggering and start of the first movement.

Trigger

Resetting to factory settings is initiated via the integrated web application.

NOTE

For technical reasons, a module with EtherCAT interface **cannot** be reset to factory settings via the internal web application. In this special case, contact SCHUNK Service.

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	WRN_NOT_FEASIBLE

- The module indicates successful resetting to factory settings by setting the status bit "ready for shutdown", ▶ 7.4 [🗋 89]. 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 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.

5 System parameters

5.1 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

5.2 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 can be read out or written acyclically.

All parameters that do not appear in this list are internal or reserved parameters. In the device description files, more parameters may be displayed than are listed here.

0x0040

0x0040 – <plc_sync_output>

Short description:	This parameter corresponds to the image of the cyclical input data (PLC frame input data)
Access rights:	Read
Data type:	4 x UINT32(16 Byte)

0x0048 **0x0048 – <plc_sync_output>**

Short description:	Short description: This parameter corresponds to the image of the cyclical output data (PLC frame output data).
Access rights:	Read and write
Data type:	4 x UINT32(16 Byte)

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 application

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 ▶ 6.2 [🗋 62]

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 ▶ 6.1 [🗋 59]

0x0128 **0x0128 – <sys_msg_req>**

Short description:	With this parameter the output of a corresponding entry from the diagnostic memory can be requested by writing an index.
Access rights:	Read and write
Data type:	UINT16

NOTE

The 32 most recent diagnostic events are stored in the diagnostic memory.

0x0130 **0x0130 – <sys_msg_buffer>**

Short description:	The requested diagnostic memory entry can be read out via this parameter.
Access rights:	Read
Data type:	CHAR[214]
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 – <wp_release_delta>**

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

0x0580 **0x0580 – <grp_pos_margin>**

Short description:	With this parameter the tolerance value of the workpiece position window can be read and written.
Access rights:	Read and write
Data type:	FLOAT
Unit:	Millimeter [mm]

0x0590 **0x0590 – <ref_type>**

Short description:	This parameter can be used to set the direction of movement during referencing.
Access rights:	Read and write
Data type:	ENUM
Unit:	0 = inward
	1 = outward
	2 = outward, with measuring travel - internal function (not intended for use in customer systems)
	3 = inward, with measuring travel - internal function (not intended for use in customer systems)

0x05A8 **0x05A8 – <grp_prepos_delta>**

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

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:	With this parameter the lower exact error limit of the supply voltage of the motor can be read out.
Access rights:	Read
Data type:	FLOAT
Unit:	Volt [V]

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

Short description:	With this paramete the upper exact error limit of the supply voltage of the motor can be read out.
Access rights:	Read
Data type:	FLOAT
Unit:	Volt [V]

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

Short description:	With this parameter the lower exact error limit of the supply voltage of the logic part can be read out.
Access rights:	Read
Data type:	FLOAT
Unit:	Volt [V]

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

Short description:	With this parameter the upper exact error limit of the supply voltage of the logic part can be read out.
Access rights:	Read
Data type:	FLOAT
Unit:	Volt [V]

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

Short description:	With htis parameter the lower exact error limit of the temperature of the logic part can be read out.
Access rights:	Read
Data type:	FLOAT
Unit:	Degrees Celsius [°C]

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

Short description:	With this parameter the upper exact error limit of the temperature of the logic part can be read out.
Access rights:	Read
Data type:	FLOAT
Unit:	Degrees Celsius [°C]

0x0840 **0x0840 – <meas_lgc_temp>**

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

0x0870 **0x0870 – <meas_lgc_volt>**

Short description:	With this parameter the current measured supply voltage of the logic part can be read out.
Access rights:	Read
Data type:	FLOAT
Unit:	Volt [V]

0x0878 **0x0878 – <meas_mot_volt>**

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

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

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

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

Short description:	With this parameter the upper exact warning limit of the supply voltage of the motor can be read out and written.
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]

0x1730 **0x1730 – <control_dword>**

Short description:	This parameter corresponds to the image of the control double word of the cyclical output data.
Access rights:	Read and write
Data type:	UINT32

0x1738 **0x1738 – <status_dword>**

Short description:	This parameter corresponds to the image of the status double word of the cyclical input data.
Access rights:	Read
Data type:	UINT32

5.3 Parameter configuration

All system parameters for which the user has write permissions can be parameterized via acyclic data exchange ▶ 3.1.2 [🗋 19].

For further information on parameterization, see chapter ▶ 5.2 [🗋 47].

6 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.

6.1 Warnings

If the diagnostics detect that a warning event has occurred, the module enters a warning state. A warning code is generated and transmitted cyclically. 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 cyclically.

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, ▶ 7.3 [🗋 85]-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.

HEX 0x90 / DEC 144

WRN_LGC_TEMP_LO

WARNING_LOGIC_TEMP_LOW

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

HEX 0x91 / DEC 145 **WRN_LGC_TEMP_HI**

WARNING_LOGIC_TEMP_HIGH

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

HEX 0x96 / DEC 146 **WRN_MOT_TEMP_LO**

WARNING_MOTOR_TEMP_LOW

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

HEX 0x93 / DEC 147 **WRN_MOT_TEMP_HI**

WARNING_MOTOR_TEMP_HIGH

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

HEX 0x94 / DEC 148 **WRN_NOT_FEASIBLE**

WARNING_CMD_NOT_FEASIBLE

Diagnostic event:	The request cannot be executed.
Ability to acknowledge:	acknowledgeable/self-acknowledging

NOTE

Information describing the cause of this warning in more detail is stored in the diagnostic memory. Read out diagnostic memory, see parameters <sys_msg_req>, ▶ 5.2 [🗋 48].

HEX 0x96 / DEC 150 **WRN_LGC_VOLT_LO**

WARNING_LOGIC_VOLTAGE_LOW

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

HEX 0x97 / DEC 151 **WRN_LGC_VOLT_HI**

WARNING_LOGIC_VOLTAGE_HIGH

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

HEX 0x98 / DEC 152 **WRN_MOT_VOLT_LO**

WARNING_MOTOR_VOLTAGE_LOW

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

HEX 0x99 / DEC 153 **WRN_MOT_VOLT_HI**

WARNING_MOTOR_VOLTAGE_HIGH

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

HEX 0x9B / DEC 155 **WRN_FLASH_FAILED**

WARNING_FLASHING_NOT_FEASIBLE

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

HEX 0x9D / DEC 157 **WRN_FACT_FAILED**

WARNING_FACTORY_RESET_UNSUCCESSFUL

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

HEX 0x9F / DEC 159 **WRN_SD_NOT_PREP**

WARNING_SHUTDOWN_NOT_PREPARED

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

6.2 Error

If the diagnostics detect that a warning event has occurred, the module enters an error state. An error code is generated and transmitted cyclically. 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.

HEX 0x6C / DEC 108

ERR_MOT_TEMP_LO

ERROR_MOTOR_TEMP_LOW

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

HEX 0x6D / DEC 109

ERR_MOT_TEMP_HI

ERROR_MOTOR_TEMP_HIGH

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

HEX 0x70 / DEC 112 **ERR_LGC_TEMP_LO**

ERROR_LOGIC_TEMP_LOW

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

HEX 0x71 / DEC 113 **ERR_LGC_TEMP_HI**

ERROR_LOGIC_TEMP_HIGH

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

HEX 0x72 / DEC 114 **ERR_LGC_VOLT_LO**

ERROR_LOGIC_VOLTAGE_LOW

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

HEX 0x73 / DEC 115 **ERR_LGC_VOLT_HI**

ERROR_LOGIC_VOLTAGE_HIGH

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

HEX 0x74 / DEC 116 **ERR_MOT_VOLT_LO**

ERROR_MOTOR_VOLTAGE_LOW

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

NOTE

For modules with GKE: Monitoring of this error is active when the drive is in control. If the drive is not being controlled, this error occurs with a time delay.

For modules without GKE: Monitoring of this error is permanently active.

HEX 0x75 / DEC 117 **ERR_MOT_VOLT_HI**

ERROR_MOTOR_VOLTAGE_HIGH

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

NOTE

For modules with GKE: Monitoring of this error is active when the drive is in control. Furthermore, the error is checked if a movement is to be started from position maintenance.

For modules without GKE: Monitoring of this error is permanently active.

HEX 0x7E / DEC 126 **ERR_POWER_STAGE**

ERROR_POWER_STAGE

Diagnostic event:	During operation of the power output stage, a voltage, overcurrent or temperature error has occurred, which has caused the power output stage to switch off.
Ability to acknowledge:	requiring acknowledgment

HEX 0x8A / DEC 138 **ERR_ENC_PHASE**

ERROR_ENCODER_PHASESHIFT

Diagnostic event:	The phase shift of the encoder signals is outside the tolerance range, i.e. the internal position measuring system is faulty.
Ability to acknowledge:	not acknowledgeable

HEX 0x8B / DEC 139 **ERR_ENC_SIN_LO**

ERROR_ENCODER_SINE_LOW

Diagnostic event:	The signal measured for the sine track is too low, i.e. the internal position measuring system is faulty.
Ability to acknowledge:	not acknowledgeable

HEX 0x8C / DEC 140 **ERR_ENC_SIN_HI**

ERROR_ENCODER_SINE_HIGH

Diagnostic event:	The signal measured for the sine track is too high, i.e. the internal position measuring system is faulty.
Ability to acknowledge:	not acknowledgeable

HEX 0x8D / DEC 141 **ERR_ENC_COS_LO**

ERROR_ENCODER_COSINE_LOW

Diagnostic event:	The signal measured for the cosine track is too low, i.e. the internal position measuring system is faulty.
Ability to acknowledge:	not acknowledgeable

HEX 0x8E / DEC 142 **ERR_ENC_COS_HI**

ERROR_ENCODER_COSINE_HIGH

Diagnostic event:	The signal measured for the cosine track is too high, i.e. the internal position measuring system is faulty.
Ability to acknowledge:	not acknowledgeable

HEX 0x8F / DEC 143 **ERR_ENC_SHORTCUT**

ERROR_ENCODER_SHORTCUT

Diagnostic event:	The signals of the sine and cosine tracks are unexpectedly identical, i.e. the internal position measuring system is faulty.
Ability to acknowledge:	not acknowledgeable

HEX 0xD5 / DEC 213 **ERR_SOFT_LOW**

ERROR_SOFT_LOW

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

HEX 0xD6 / DEC 214 **ERR_SOFT_HIGH**

ERROR_SOFT_HIGH

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

HEX 0xD9 / DEC 217 **ERR_FAST_STOP**

ERROR_FAST_STOP

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

HEX 0xDE / DEC 222 **ERR_CURRENT**

ERROR_CURRENT

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

HEX 0xE4 / DEC 228 **ERR_TOO_FAST**

ERROR_TOO_FAST

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

HEX 0xEF / DEC 239 **ERR_COMM_LOST**

ERROR_COMMUNICATION_LOST

Diagnostic event:	The communication link between the module and a controller has been interrupted.
Ability to acknowledge:	requiring acknowledgment

HEX 0xF0 / DEC 240 **ERR_REF_ABORT_TO**

ERROR_REFERENCE_ABORT_TIMEOUT

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

HEX 0xF1 / DEC 241 **ERR_MOV_ABORT_TO**

ERROR_MOVE_ABORT_TIMEOUT

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

HEX 0xF2 / DEC 242 **ERR_NO_REF**

ERROR_NO_REFERENCE

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

HEX 0xF4 / DEC 244 **ERR_MOVE_BLOCKED**

ERROR_MOVE_BLOCKED

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

HEX 0xF6 / DEC 246 **ERR_NO_BLOCK_DEV**

ERROR_BOOT_BLOCK_DEVICE_NOT_DETECTED

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

HEX 0xF7 / DEC 247 **ERR_NO_COMM**

ERROR_BOOT_ANYBUS_NOT_DETECTED

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

HEX 0xF8 / DEC 248 **ERR_WRONG_HW**

ERROR_BOOT_HARDWARE_NOT_SUPPORTED_BY_FIRMWARE

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

HEX 0xFA / DEC 250 **ERR_SD_FAILED**

ERROR_PREPARE_FOR_SHUTDOWN_FAILED

Diagnostic event:	Preparing to shutdown failed.
Ability to acknowledge:	requiring acknowledgment

7 Appendix

7.1 Application examples

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

NOTE

In principle, the applications described apply to both sizes of the EGI. However, the following examples are described from the point of view of the *EGI 80*, which means some parameter values are accepted as valid which would be invalid with *EGI 40*.

Scenario description	Example
An absolute position movement is carried out.	Example 1 [🗋 70]
A relative position movement is carried out.	Example 2 [🗋 70]
Workpiece gripping is carried out. A workpiece is available.	Example 3 [🗋 71]
Workpiece gripping is carried out. A workpiece is *not* available.	Example 4 [🗋 71]
Workpiece gripping at expected position is carried out. Pre-positioning is carried out. The correct workpiece is gripped during the subsequent gripping process.	Example 5 [🗋 71]
Workpiece gripping at expected position is carried out. Pre-positioning is *not* carried out. The correct workpiece is gripped during the gripping process.	Example 6 [🗋 72]
Workpiece gripping at expected position is carried out. Pre-positioning is carried out. A wrong workpiece is gripped during the gripping process.	Example 7 [🗋 72]
Workpiece gripping at expected position is carried out. Pre-positioning is carried out. *No* workpiece is gripped during the gripping process.	Example 8 [🗋 73]
Workpiece gripping at expected position is carried out. During pre-positioning, a hard impact occurs.	Example 9 [🗋 74]
A workpiece release is carried out subsequently to workpiece gripping. The target position is within the software limits.	Example 10 [🗋 74]
A workpiece release is carried out subsequently to workpiece gripping. The target position is outside the software limits.	Example 11 [🗋 75]

Scenario description	Example
A workpiece release is carried out subsequently to workpiece gripping at the expected position. The target position is within the software limits.	Example 12 [🗋 76]
Permitted request to the module.	Example 13 [🗋 76]
Permitted request to the module.	Example 14 [🗋 77]
Permitted request to the module.	Example 15 [🗋 77]
Permitted request to the module.	Example 16 [🗋 77]
Permitted request to the module.	Example 17 [🗋 78]
Impermissible request to the module.	Example 18 [🗋 78]
Impermissible request to the module.	Example 19 [🗋 78]
Impermissible request to the module.	Example 20 [🗋 79]
Workpiece re-gripping is carried out subsequently to a simple gripping movement. A workpiece is available and is *not* lost during re-gripping.	Example 21 [🗋 79]
Workpiece re-gripping is carried out subsequently to a simple gripping movement. A workpiece is present and is lost during re-gripping.	Example 22 [🗋 80]
Workpiece re-gripping is carried out subsequently to a combined gripping movement at the expected position. A workpiece is present and is lost during re-gripping.	Example 23 [🗋 81]
Two relative positioning movements are carried out one after the other in a time-optimized manner.	Example 24 [🗋 82]

EXAMPLE 1 **Absolute positioning movement**

The current actual position of the module is 30 mm. The controller cyclically transmits 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 ▶ 4.2.3 [🗋 30].

EXAMPLE 2 **Relative positioning movement**

The current actual position of the module is 66 mm. The controller cyclically transmits 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 process 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 ▶ 4.2.4 [🗋 31].

EXAMPLE 3

Workpiece gripping (1)

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

- Control bit status "grip direction" = 1 (≙ I.D. gripping)
- Gripping force = 60000 mN (≙ 60 N)
- Control bit "grip workpiece" 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 "workpiece gripped" is set.

For further information, see chapter ▸ 4.3.1 [🗋 34].

EXAMPLE 4

Workpiece gripping (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 cyclically:

- Control bit status "grip direction" = 0 (≙ O.D. gripping)
- Gripping force = 25000 mN (≙ 25 N)
- Control bit "grip workpiece" 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 workpiece detected".

For further information, see chapter ▸ 4.3.1 [🗋 34].

EXAMPLE 5

Workpiece gripping at expected position (1)

A workpiece should be gripped from the inside at 80 mm and is also available there. The current actual position of the module is 10 mm.

The parameterization of the module is as follows:

- Parameter value <grp_prepos_delta> = 9 (≙ 9 mm)
- Parameter value <grp_pos_margin> = 1 (≙ 1 mm)

The controller cyclically transmits the following request to the module:

- Position parameter = 80000 µm (≙ 80 mm)
- Speed parameter = 175000 µm/s (≙ 175 mm/s)
- Gripping force = 80000 mN (≙ 80 N)
- Control bit status "grip direction" = 1 (≙ I.D. gripping)
- Control bit "grip workpiece at expected position" is set

→ The module calculates the pre-position (calculation: gripping position 80 mm - [<grp_pos_margin> 1 mm + <grp_prepos_delta> 9 mm] = pre-position 70 mm). Since the current position is not between the pre-position and the gripping position, the module

then moves to the calculated absolute pre-position 70 mm relative to the zero point. The maximum speed that the module can reach during this positioning process is 175 mm/s.

→ When the pre-position is reached, the module automatically switches to the gripping movement with the gripping speed <grp_vel> = 50 mm/s. The workpiece is gripped with 80 N at the actual gripping position 80.5 mm. Since the actual gripping position is within the workpiece position window, the status bit "workpiece gripped" is set.

For further information, see chapter ▸ 4.3.2 [🗋 35].

EXAMPLE 6 **Workpiece gripping at expected position (2)**

A workpiece should be gripped from the outside at 50 mm and is also available there. The current actual position of the module is 60 mm.

The parameterization of the module is as follows:

- Parameter value <grp_prepos_delta> = 18 (≙ 18 mm)
- Parameter value <grp_pos_margin> = 2 (≙ 2 mm)

The controller cyclically transmits the following request to the module:

- Position parameter = 50000 µm (≙ 50 mm)
- Speed parameter = 200000 µm/s (≙ 200 mm/s)
- Gripping force = 65000 mN (≙ 65 N)
- Control bit status "grip direction" = 0 (≙ O.D. gripping)
- Control bit "grip workpiece at expected position" is set

→ The module calculates the pre-position (calculation: gripping position 50 mm + [<grp_pos_margin> 2 mm + <grp_prepos_delta> 18 mm] = pre-position 70 mm). Since the current position is between the pre-position and the gripping position, the module immediately carries out the gripping movement with the gripping speed <grp_vel> = 50 mm/s.

→ The workpiece is gripped with 65 N at the actual gripping position 49.1 mm. Since the actual gripping position is within the workpiece position window, the status bit "workpiece gripped" is set.

For further information, see chapter ▸ 4.3.2 [🗋 35].

EXAMPLE 7 **Workpiece gripping at expected position (3)**

A workpiece should be gripped from the outside at 20 mm, but there is a workpiece at 25 mm. The current actual position of the module is 55 mm.

The parameterization of the module is as follows:

- Parameter value <grp_prepos_delta> = 9 (≙ 9 mm)
- Parameter value <grp_pos_margin> = 1 (≙ 1 mm)

The controller cyclically transmits the following request to the module:

- Position parameter = 20000 µm (≙ 20 mm)
- Speed parameter = 90000 µm/s (≙ 90 mm/s)
- Gripping force = 100000 mN (≙ 100 N)
- Control bit status "grip direction" = 0 (≙ O.D. gripping)
- Control bit "grip workpiece at expected position" is set

→ The module calculates the pre-position (calculation: gripping position 20 mm + [<grp_pos_margin> 1 mm + <grp_prepos_delta> 9 mm] = pre-position 30 mm). Since the current position is not between the pre-position and the gripping position, the module then moves to the calculated absolute pre-position 30 mm relative to the zero point. The maximum speed that the module can reach during this positioning process is 90 mm/s.

→ When the pre-position is reached, the module automatically switches to the gripping movement with the gripping speed <grp_vel> = 50 mm/s. The workpiece is gripped with 100 N at the actual gripping position 25.0 mm. Since the actual gripping position is outside the workpiece position window, the status bit "wrong workpiece gripped" is set.

For further information, see chapter ▶ 4.3.2 [🗋 35].

EXAMPLE 8

Workpiece gripping at expected position (4)

A workpiece should be gripped from the outside at 55 mm, but there is no workpiece available. The current actual position of the module is 20 mm.

The parameterization of the module is as follows:

- Parameter value <grp_prepos_delta> = 4 (≙ 4 mm)
- Parameter value <grp_pos_margin> = 1 (≙ 1 mm)

The controller cyclically transmits the following request to the module:

- Position parameter = 55000 µm (≙ 55 mm)
- Speed parameter = 150000 µm/s (≙ 150 mm/s)
- Gripping force = 45000 mN (≙ 45 N)
- Control bit status "grip direction" = 1 (≙ I.D. gripping)
- Control bit "grip workpiece at expected position" is set

→ The module calculates the pre-position (calculation: gripping position 55 mm - [<grp_pos_margin> 1 mm + <grp_prepos_delta> 4 mm] = pre-position 50 mm). Since the current position is not between the pre-position and the gripping position, the module then moves to the calculated absolute pre-position 50 mm relative to the zero point. The maximum speed that the module can reach during this positioning process is 150 mm/s.

→ When the pre-position is reached, the module automatically switches to the gripping movement with the gripping speed <grp_vel> = 50 mm/s.

→ Since no workpiece is present, the gripping movement is terminated in a controlled manner when the workpiece position window is exceeded (gripping position 55 mm + <grp_pos_margin> 1 mm = 56 mm). This is displayed by setting the status bit "no workpiece detected".

For further information, see chapter ▶ 4.3.2 [🗋 35].

EXAMPLE 9 **Workpiece gripping at expected position (5)**

A workpiece should be gripped from the inside at 90 mm, but there is a workpiece at 60 mm. The current actual position of the module is 5 mm.

The parameterization of the module is as follows:

- Parameter value <grp_prepos_delta> = 4 (≙ 4 mm)
- Parameter value <grp_pos_margin> = 1 (≙ 1 mm)

The controller cyclically transmits the following request to the module:

- Position parameter = 90000 µm (≙ 90 mm)
- Speed parameter = 200000 µm/s (≙ 200 mm/s)
- Gripping force = 75000 mN (≙ 75 N)
- Control bit status "grip direction" = 1 (≙ I.D. gripping)
- Control bit "grip workpiece at expected position" is set

→ The module calculates the pre-position (calculation: gripping position 90 mm - [<grp_pos_margin> 1 mm + <grp_prepos_delta> 4 mm] = pre-position 85 mm). Since the current position is not between the pre-position and the gripping position, the module then moves to the calculated absolute pre-position 75 mm relative to the zero point. The maximum speed that the module can reach during this positioning process is 200 mm/s.

→ The module detects an impact at 60 mm during pre-positioning and switches to the error state. The status bit "error" is displayed and the error code "ERR_MOVE_BLOCKED" is transmitted in the diagnostic double word.

For further information, see chapter ▶ 4.3.2 [🗋 35].

EXAMPLE 10 **Workpiece release (1)**

A workpiece was previously gripped from the **inside** with a simple gripping movement. The software limits are at 0 and 100 mm. The current gripping position is 60 mm.

The parameterization of the module is as follows:

- <wp_release_delta> = 5 (≙ 5 mm)

The controller cyclically transmits the following request to the module:

- Control bit "release workpiece" is set

The module knows that the workpiece was previously gripped from the inside and then, taking into account the software limits, moves -5 mm opposite to the last gripping direction 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 ▶ 4.3.5 [🗋 42].

NOTE

Since the gripped status results from "workpiece gripping", only the value of the parameter <wp_release_delta> is used as the relative release position.

EXAMPLE 11

Workpiece release (2)

A workpiece was previously gripped from the **outside** with a simple gripping movement. The software limits are at 0 and 100 mm. The current gripping position is 95 mm.

The parameterization of the module is as follows:

- <wp_release_delta> 10 (≙ 10 mm)

The controller cyclically transmits the following request to the module:

- Control bit "release workpiece" is set

→ The module knows that the workpiece was previously gripped from the outside and then, taking into account the software limits, moves +10 mm in the opposite direction to the last gripping direction 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 ▶ 4.3.5 [🗋 42].

NOTE

Since the gripped status results from "workpiece gripping", only the value of the parameter <wp_release_delta> is used as the relative release position.

EXAMPLE 12

Workpiece release (3)

A workpiece was previously gripped from the **outside** with a combined gripping movement. The software limits are at 0 and 100 mm. The current gripping position is 70 mm.

The parameterization of the module is as follows:

- Parameter value <wp_release_delta> = 7 (≙ 7 mm)
- Parameter value <grp_pos_margin> = 3 (≙ 3 mm)

The controller cyclically transmits the following request to the module:

- Control bit "release workpiece" is set

→ The module calculates the relative release position. (Calculation: <wp_release_delta> 7 mm + <grp_pos_margin> 3 mm = relative release position 10 mm).

The module knows that it has previously gripped the workpiece from the outside at a defined position. The module then moves in the opposite direction of the last gripping direction and, in compliance with the software limits, from the current actual position to the calculated relative release position (+10 mm).

→ The process ends when the target position is reached. The current actual position of the module is 80 mm ± tolerance of the module-specific positioning accuracy. Status bit "position reached" is set.

For further information, see chapter ▸ 4.3.5 [🗋 42].

NOTE

In the case of a "Workpiece release" after "Workpiece gripping at expected position", the value of the parameter <grp_pos_margin> is taken into account. This ensures that "Workpiece gripping at expected position" can be performed repeatedly immediately after the release. Without taking this into account, it is possible that the actual position would be too close to the workpiece, so that "Workpiece gripping at expected position" could not be carried out.

EXAMPLE 13

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 cyclically transmitted movement parameters are OK. These prerequisites reflect those of a permissible request.

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

For further information, see chapter ▸ 3.1.1.1 [🗋 12].

EXAMPLE 14 **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 cyclically transmitted movement parameters are OK. These prerequisites 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. This prerequisite reflects that of a permissible request.

→ The movement ends when the target position is reached.

For further information, see chapter ▶ 3.1.1.1 [🗋 12].

EXAMPLE 15 **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 cyclically transmitted movement parameters are OK. These prerequisites 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 prerequisites 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 ▶ 3.1.1.1 [🗋 12].

EXAMPLE 16 **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 cyclically transmitted movement parameters are OK. These prerequisites 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 prerequisites reflect those of an impermissible request.

→ 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 ▶ 3.1.1.1 [🗋 12].

EXAMPLE 17 **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 cyclically transmitted movement parameters are OK. These prerequisites 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 prerequisites reflect those of an impermissible request.

→ The status bit "not feasible" is set.

→ Error: Impermissible bit combination

For further information, see chapter ▸ 3.1.1.1 [🗋 12].

EXAMPLE 18 **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 cyclically transmitted movement parameters are OK. These prerequisites reflect those of an impermissible request.

→ The status bit "not feasible" is set.

→ Error: The module is not referenced.

For further information, see chapter ▸ 3.1.1.1 [🗋 12].

EXAMPLE 19 **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 cyclically transmitted movement parameters are OK. These prerequisites reflect those of an impermissible 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 ▸ 3.1.1.1 [🗋 12].

EXAMPLE 20 **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 cyclically transmitted movement parameters are **incorrect**. These prerequisites reflect those of an impermissible request.

→ The status bit "not feasible" is set.

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

For further information, see chapter ▶ 3.1.1.1 [🗋 12].

EXAMPLE 21 **Workpiece re-gripping (1)**

A workpiece should be gripped from a blister pack from the outside with a simple gripping movement. However, the application requires re-gripping to prevent workpiece loss. The workpiece is available and is **not** lost during the re-gripping.

The parameterization of the module is as follows:

- Parameter value <grp_prehold_time> = 2300 ms (≙ 2.3 s)

The controller cyclically transmits the following request to the module:

- Control bit status grip direction = 0 (≙ O.D. gripping)
- Gripping force = 50000 mN (≙ 50 N)
- Control bit "grip workpiece" 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 displayed by setting the status bit "workpiece pre-grip started". Re-gripping is carried out for a further 2.3 seconds.

→ When the re-gripping is terminated, it is recognized that a workpiece has been gripped. This is displayed by setting the status bit "workpiece gripped". The status bit "workpiece pre-grip started" remains set.

For further information, see chapter ▶ 4.3.3 [🗋 39].

EXAMPLE 22

Workpiece re-gripping (2)

A workpiece should be gripped from a blister pack from the inside with a simple gripping movement. However, the application requires re-gripping to prevent workpiece loss. The workpiece is available, but is lost during re-gripping.

The parameterization of the module is as follows:

- Parameter value <grp_prehold_time> = 1600 ms (≙ 1.6 s)

The controller cyclically transmits the following request to the module:

- Control bit status "grip direction" = 1 (≙ I.D. gripping)
- Gripping force = 70000 mN (≙ 70 N)
- Control bit "grip workpiece" is set

→ The module then performs a gripping movement as an I.D. gripper. The base jaws move away from each other. The workpiece contact is detected and displayed by setting the status bit "workpiece pre-grip started". Re-gripping is carried out for a further 1.6 seconds.

→ The workpiece is lost during re-gripping.

→ When the re-gripping is terminated, it is recognized that the workpiece has been lost. This is displayed by setting the status bit "workpiece lost". In this case the status bit "workpiece pre-grip started" is reset again.

For further information, see chapter ▶ 4.3.3 [🗋 39].

EXAMPLE 23

Workpiece re-gripping (3)

A workpiece should be gripped from a blister pack from the outside with a combined gripping movement. However, the application requires re-gripping to prevent workpiece loss. The workpiece is gripped at 15 mm in the aligned status. Since the workpiece is removed from the blister, the first workpiece contact is detected at 31 mm. The current actual position of the module is 85 mm. The workpiece is available, but is lost during re-gripping.

The parameterization of the module is as follows:

- Parameter value <grp_prehold_time> = 4300 ms (≙ 4.3 s)
- Parameter value <grp_prepos_delta> = 22 (≙ 22 mm)
- Parameter value <grp_pos_margin> = 3 (≙ 3 mm)

The controller cyclically transmits the following request to the module:

- Position parameter = 15000 µm (≙ 15 mm)
- Speed parameter = 167000 µm/s (≙ 167 mm/s)
- Gripping force = 90000 mN (≙ 90 N)
- Control bit status "grip direction" = 0 (≙ O.D. gripping)
- Control bit "grip workpiece at expected position" is set

→ The module calculates the pre-position (calculation: gripping position 15 mm + [<grp_pos_margin> 3 mm + <grp_prepos_delta> 22 mm] = pre-position 40 mm). Since the current position is not between the pre-position and the gripping position, the module then moves to the calculated absolute pre-position 40 mm relative to the zero point. The maximum speed that the module can reach during this positioning process is 167 mm/s.

→ When the pre-position is reached, the module automatically switches to the gripping movement with the gripping speed <grp_vel> = 50 mm/s.

→ The workpiece contact is detected at 31 mm. The holding pressure is started and carried out for 4.3 s. The start of re-gripping is displayed by setting the status bit "workpiece pre-grip started".

→ The workpiece is lost during re-gripping.

→ When the workpiece position window is exceeded at 12 mm (gripping position 15 mm - <grp_pos_margin> 3 mm), the movement is terminated in a controlled manner and the workpiece loss is displayed by setting the status bit "workpiece lost". In this case the status bit "workpiece pre-grip started" is reset again.

For further information, see chapter ▸ 4.3.3 [🗋 39].

EXAMPLE 24

Repeat request with optimized time

The module should execute two relative positioning movements one after the other at different speeds. The first movement should be 25 mm and the second movement 60 mm. The current actual position of the module is 10 mm.

The controller cyclically transmits the following request to the module:

- Position parameter = 25000 µm (≙ 25 mm)
- Speed parameter = 35000 µm/s (≙ 35 mm/s)
- Control bit "move to relative position" is set

The module then moves +25 mm from the current actual position. The maximum speed that the module can reach during this positioning process is 35 mm/s.

→ After completing the movement, the current actual position of the module is 35 mm ± tolerance of the module-specific positioning accuracy.

→ The request is changed as follows and is again transmitted cyclically from the controller to the module:

- Position parameter = 60000 µm (≙ 60 mm)
- Speed parameter = 200000 µm/s (≙ 200 mm/s)
- Control bit "move to relative position" remains set
- Control bit "repeat command toggle" is toggled

The module then moves +60 mm from the current actual position. 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 95 mm ± tolerance of the module-specific positioning accuracy.

NOTE

The fact that the control bit "move to relative position" remains set and the same movement could be executed again by toggling the control bit "repeat command toggle" results in a time advantage compared to the standard mechanism, which states that requests can only be sent by 0 -> 1 edges.

For further information, see chapter ▶ 4.4.2 [🗋 45].

7.2 Troubleshooting

EXAMPLE 1

Reading out the IP address of a module

One employee has connected his computer directly to the module via an appropriate network cable. He wants to access the module, but does not know the IP address.

→ The IP address of a module can be read out via the "Anybus-IP Config Tool".

EXAMPLE 2

Parameterization of a replacement module

A module must be replaced by an identical module in an existing application due to a defect. The communication to the defective module is not disrupted and the replacement module must be parameterized identically to the defective module.

- The parameters can be read out acyclically via the connected controller.
- If the module has the integrated web application [🗋 22], a list of the current parameterization of the module can be downloaded.

→ The replacement module is parameterized acyclically via the connected controller or via the integrated web application.

7.3 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 ▶ 3.1.1.1 [🗋 13].

Bit 0 - fast stop

Edge change	Module reaction
0 -> 1	no reaction
1 -> 0	The module performs a quick stop, ▶ 4.2.6 [🗋 33].

If this bit has the status 0, setting another control double word bit is considered an invalid bit combination.

Bit 1 - stop

Edge change	Module reaction
0 -> 1	The module performs a controlled stop, ▶ 4.2.5 [🗋 32].
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, 16

Bit 2 - acknowledge

Edge change	Module reaction
0 -> 1	The module tries to acknowledge all existing warnings and errors, ▶ 6.1 [🗋 59], ▶ 6.2 [🗋 62].
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, 16

Bit 3 - prepare for shutdown

Edge change	Module reaction
0 -> 1	The module is preparing for shutdown, ▶ 4.1.2 [🗋 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, 4, 5, 8, 9, 10, 11, 12, 13, 14, 16

Bit 4 - softreset

Edge change	Module reaction
0 -> 1	The module is restarted on the software side, ▶ 4.1.3 [🗋 26].
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, 16

Bit 5 - release brake

Edge change	Module reaction
0 -> 1	The brake is released in order to manually remove a workpiece, ▶ 4.3.6 [🗋 44].
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, 16

Bit 6 - repeat command toggle

Edge change	Module reaction
0 -> 1	The module once again carries out the request whose bit is still pending.
1 -> 0	The module once again carries out the request whose bit is still pending.

Note: Depending on the current status of the module, there may be feedback that movements cannot be carried out again.

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 negative

Edge change	Module reaction
0 -> 1	The module executes a movement in the negative direction of movement, ▶ 4.2.1 [🗋 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: 1, 2, 3, 4, 5, 9, 10, 11, 12, 13, 14, 16

Bit 9 - jog mode positive

Edge change	Module reaction
0 -> 1	The module executes a movement in the positive direction of movement, ▶ 4.2.1 [🗋 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: 1, 2, 3, 4, 5, 8, 10, 11, 12, 13, 14, 16

Bit 10 - reference

Edge change	Module reaction
0 -> 1	The module executes a reference run, ▶ 4.2.2 [🗋 29].
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, 16

Bit 11 - release workpiece

Edge change	Module reaction
0 -> 1	The module releases a workpiece, ▶ 4.3.5 [🗋 42].
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, 16

Bit 12 - grip workpiece

Edge change	Module reaction
0 -> 1	The module performs a gripping movement, ▶ 4.3.1 [🗋 34]
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, 16

Bit 13 - move to absolute position

Edge change	Module reaction
0 -> 1	The module performs a positioning movement to an absolute position, ▶ 4.2.3 [🗋 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, 11, 12, 14, 16

Bit 14 - move to relative position

Edge change	Module reaction
0 -> 1	The module performs a positioning movement to a relative position, ▶ 4.2.2 [🗋 29].
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, 16

Bit 15 - reserved

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

Bit 16 - grip workpiece at expected position

Edge change	Module reaction
0 -> 1	The module performs a combined gripping movement to the specified, expected workpiece position.
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, 14

Bit 17 – 31 - reserved

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

7.4 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 ▶ 3.1.1.2 [🗋 15].

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 - control authority fieldbus

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	No information is reported.
1	The request sent to the module is not feasible.

Bit 4 - command successfully processed

Status	Module feedback
0	No information is reported.
1	The following requests sent to the module were successfully *processed*. • Bit 1 - stop • Bit 10 - referenced • Bit 11 - release workpiece • Bit 12 - grip workpiece • Bit 13 - move to absolute position • Bit 14 - move to relative position • Bit 16 - grip workpiece at expected position The following requests sent to the module are currently being successfully *executed*. • Bit 5 - release brake • Bit 8 - jog mode negative • Bit 9 - jog mode positive

Bit 5 - command received toggel

Status change	Module feedback
0 -> 1	The module acknowledges receipt of a request.
1 -> 0	The module acknowledges receipt of a request.

Bit 6 - warning

Status	Module feedback
0	There is no warning.
1	There is a warning.

Bit 7 - error

Status	Module feedback
0	There is no error.
1	There is an error.

Bit 8 - brake released

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

Bit 9 - software limit reached

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 workpiece detected

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

Bit 12 - workpiece gripped

Status	Module feedback
0	No information is reported.
1	The previous gripping process was successful or the correct workpiece was gripped.

Bit 13 - position reached

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

Bit 14 - workpiece pre-grip started

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 - wrong workpiece gripped

Status	Module feedback
0	No information is reported.
1	The wrong workpiece was gripped during the previous combined gripping process.

Bit 18 – 31 - reserved

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

7.5 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.

7.5.1 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.

7.5.2 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.

7.5.3 EtherCAT

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

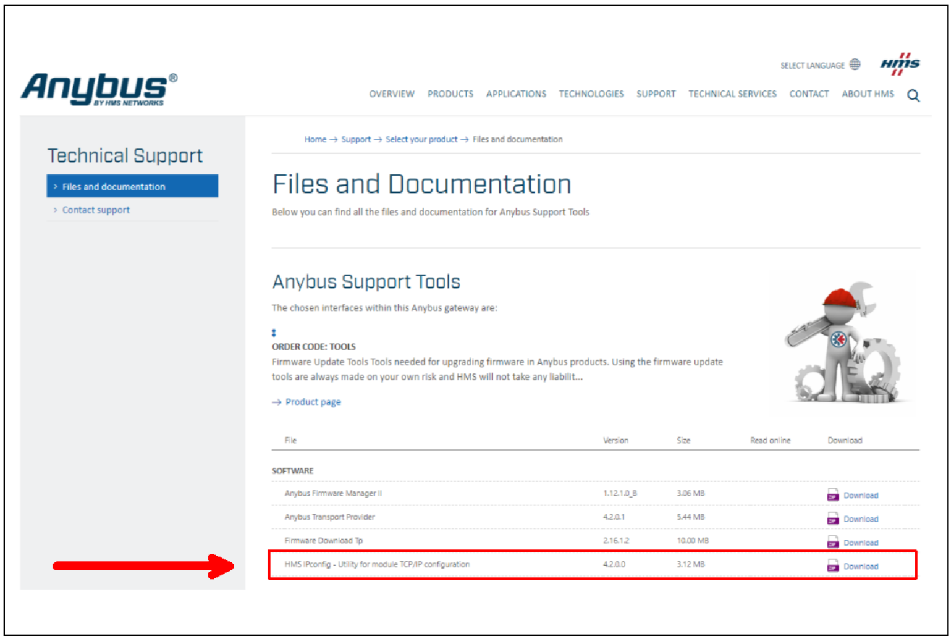
LED	Designation	Color	Function
LOG	Supply logic	Green	**LED off:** No supply voltage logic present, or supply voltage logic is outside the operating range for the module.
			LED lights up green: Supply voltage logic is present.
PWR	Supply power	Green	**LED off:** No supply voltage power present, or supply 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: A serious fault is present. The bus interface has been put into a physically passive state. Contact SCHUNK Service.
ERROR	Error 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: A serious fault is present. 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

7.6 Download the Anybus-IP Config tool

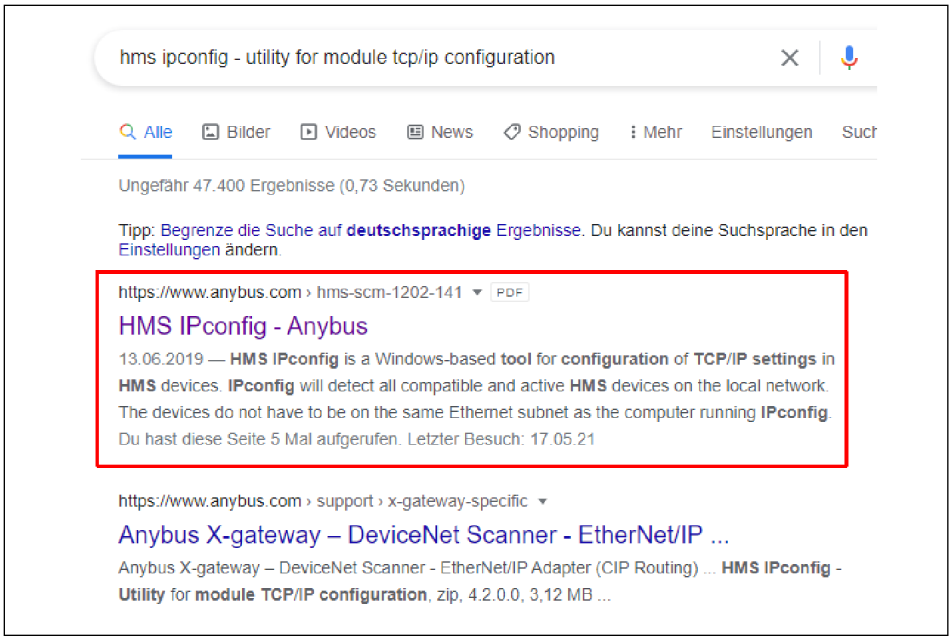
Method 1: Download via direct link

1. Go to the website at **https://www.anybus.com/support/file-doc-downloads/anybus-support-tools?orderCode=tools**.
   ✓ The "Files and Documentation" website opens.
2. Click Download under "HMS IPconfig - Utility for module TCP/IP configuration" to start the download.



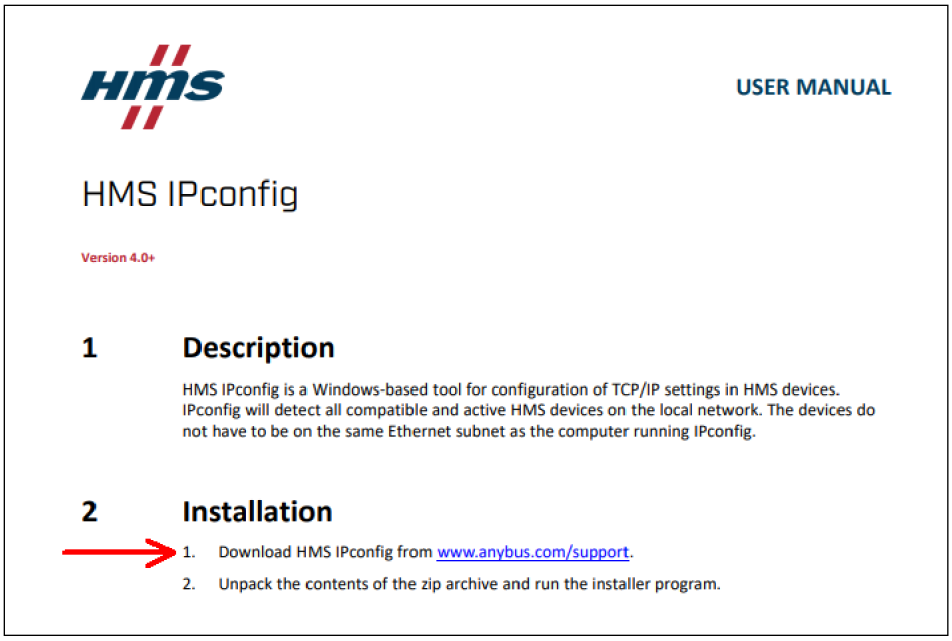
Method 2: Alternatively search for the tool on the Internet, if method 1 does not work

1. Search the Internet using the search term "*hms ipconfig - utility for module tcp/ip configuration*".
2. Select the first search result listed (HMS IPconfig - Anybus).



✓ A PDF file (USER MANUAL - HMS IPconfig) opens.

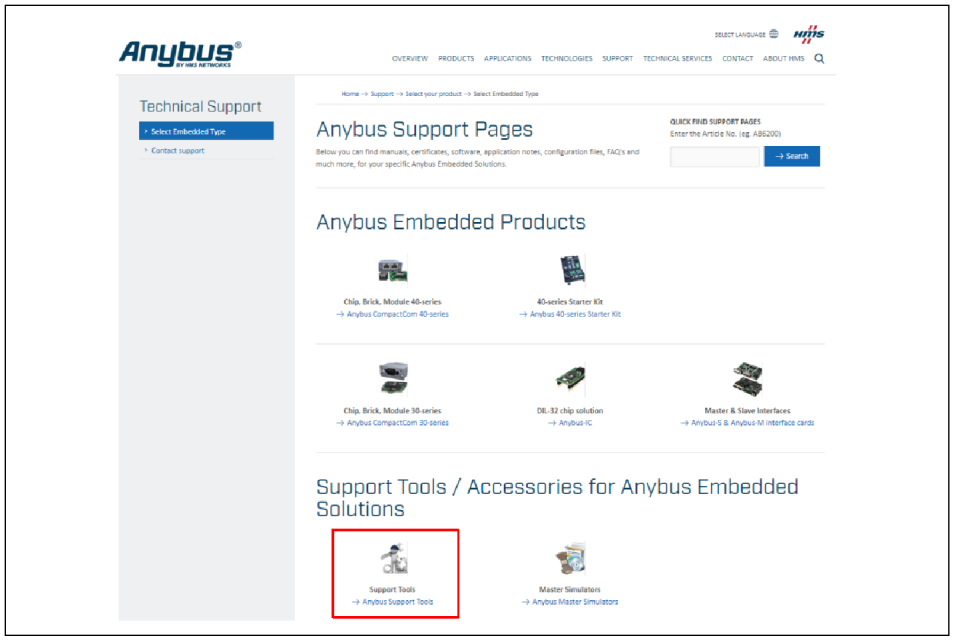
3. In the "Installation" section, click on the "www.anybus.com/support" link.



✓ The "Technical Support" website opens.

4. Select "Embedded Solution".

5. Click "Anybus Support Tools".



✓ The "Files and Documentation" website opens.

6. Click Download under "HMS IPconfig - Utility for module TCP/IP configuration" to start the download.

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

Folgen Sie uns I *Follow us*

04-2022 © 2022 SCHUNK GmbH & Co. KG 02.00 | EGI with PROFINET, EtherCAT or EtherNet/IP™ | Software Manual, Firmware 4 | en | 1491302