Software manual
EGI with PROFINET interface

Superior Clamping and Gripping

SCHUNK

Imprint

Technical changes:

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

Document number: 1396640

Version: 01.00 | 25/06/2019 | en

Dear Customer,

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

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

Best regards,

Your SCHUNK team

SCHUNK GmbH & Co. KG
Spann- und Greiftechnik

Bahnhofstr. 106 – 134
D-74348 Lauffen/Neckar

Tel. +49-7133-103-0
Fax +49-7133-103-2399

info@de.schunk.com
schunk.com

Table of contents

1 General

1.1 About this document

This software manual describes the operational and parameterization options of an **i**ntelligent **e**lectric **g**ripper with **P**ROFI**N**ET interface (EGI PN).

EGI PN is hereinafter referred to as "module".

Admissibility

This version of the software manual describes the functions of firmware versions that bear the main version number 1.XX.

Conventions

The following conventions apply to this software manual:

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

Applicable documents

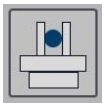
- General terms of business*
- Assembly and Operating Manual EGI *

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

1.2 Definitions and exact limits

1.2.1 Direction of movement and gripper type

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

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

1.2.2 Hardware and software limits

Hardware limits

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

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

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

Software limits

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

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

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

The software limits are only monitored in the referenced module state and may be parameterized by the user, Parameter list [▶ 30].

1.2.3 Reference point and zero point

Zero point

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

Reference point

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

1.2.4 Overview of important exact limits

The following table contains the module's most important exact limits. For a detailed description of the parameters, see chapter Parameter list [▶ 30].

Value	Minimum value	Maximum value
Ambient temperature	5°C	55°C
Logic voltage supply	21.6 V	26.4 V
Power voltage supply	21.6 V	26.4 V
Positioning speed	15 mm/s	200 mm/s
Gripping force	25 N	100 N
Software limits	0 mm	115 mm

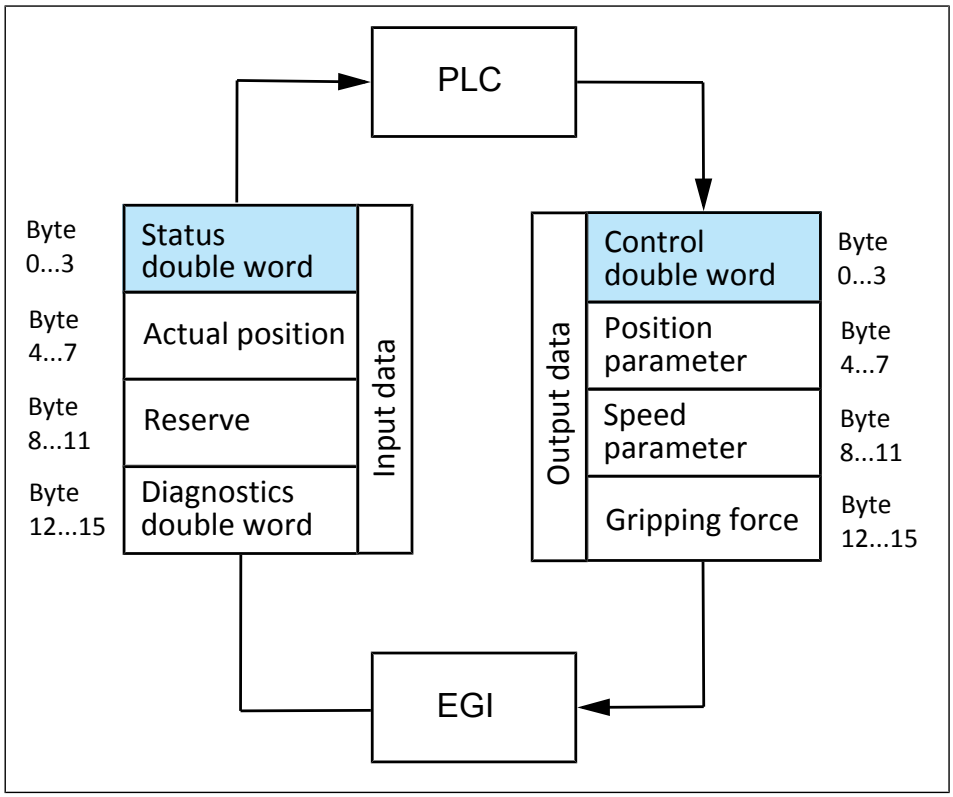
2 Communication

2.1 PROFINET

Via the integrated PROFINET interface, data can be exchanged cyclically and acyclically between the product and the controller.

2.1.1 Cyclical data exchange

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



Cyclical data exchange

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

2.1.1.1 Cyclical output data

The cyclical output data is transmitted from the PLC to the module, thereby sending requests to the module. For practical application examples, see chapter Application examples [▶ 51].

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".

Impermissible requests

Impermissible requests could be caused by the following:

- The request has been transmitted during an active movement of the module.
- The transmitted bit combination is impermissible.
- At least one movement parameter that has been transmitted is impermissible.

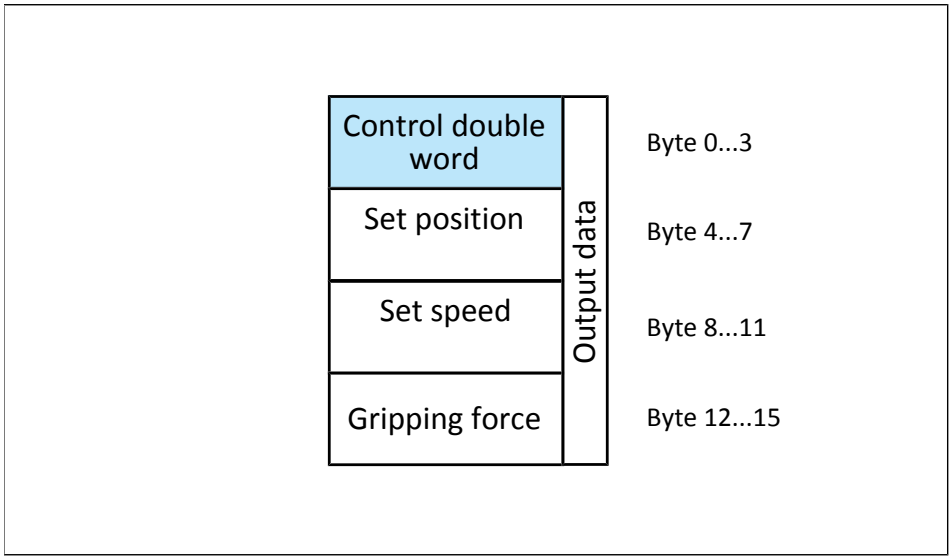
An immediate transition between active movements of the module is not permitted by the system, and results in a controlled termination of the module's current active movement.

Depending on the currently transmitted bit combination, impermissible bit combinations of the control double word will arise. For a detailed description of invalid bit combinations, see chapter Control double word [▶ 56].

A movement parameter is considered impermissible if values outside the permitted minimum and maximum limits are transmitted, Definitions and exact limits [▶ 5].

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

In bytes 0 – 3 of the cyclical output data, the control double word is transmitted. The structure of the control double word is shown in the following table. For a detailed description of the control double word, see chapter Control double word [▶ 56].

Word	Byte	Bit	Cyclical output data	For further information, see chapter
0	0	0	fast stop	Control double word [▶ 56], Bit0
		1	stop	Control double word [▶ 56], Bit1
		2	acknowledge	Control double word [▶ 56], Bit2
		3	prepare for shutdown	Control double word [▶ 56], Bit3
		4	softreset	Control double word [▶ 57], Bit4
		5	release brake	Control double word [▶ 57], Bit5
		6	reserved	-
		7	grip direction	Control double word [▶ 57], Bit7
	1	8	jog mode minus	Control double word [▶ 57], Bit8
		9	jog mode plus	Control double word [▶ 58], Bit9
		10	reference	Control double word [▶ 58], Bit10
		11	release work piece	Control double word [▶ 58], Bit11
		12	grip work piece	Control double word [▶ 58], Bit12
		13	move to absolute position	Control double word [▶ 59], Bit13
		14	move to relative position	Control double word [▶ 59], Bit14
		15	reserved	-
1	2	16	reserved	-
		17	reserved	-
		18	reserved	-
		19	reserved	-
		20	reserved	-
		21	reserved	-
		22	reserved	-
		23	reserved	-
	3	24	reserved	-
		25	reserved	-
		26	reserved	-
		27	reserved	-
		28	reserved	-
		29	reserved	-
		30	reserved	-
		31	reserved	-

Position

- In bytes 4 – 7 of the cyclical output data, data is transmitted that is used for positioning purposes, Parameter list [▶ 30].
- The data format of the parameter is signed 32 bits and represents a value in micrometers [µm].

Speed

- In bytes 8 – 11 of the cyclical output data, the value of the target speed of a movement is transmitted, Parameter list [▶ 30].
- The data format of the parameter is signed 32 bits and represents a value in micrometers per second [µm/s].

Gripping force

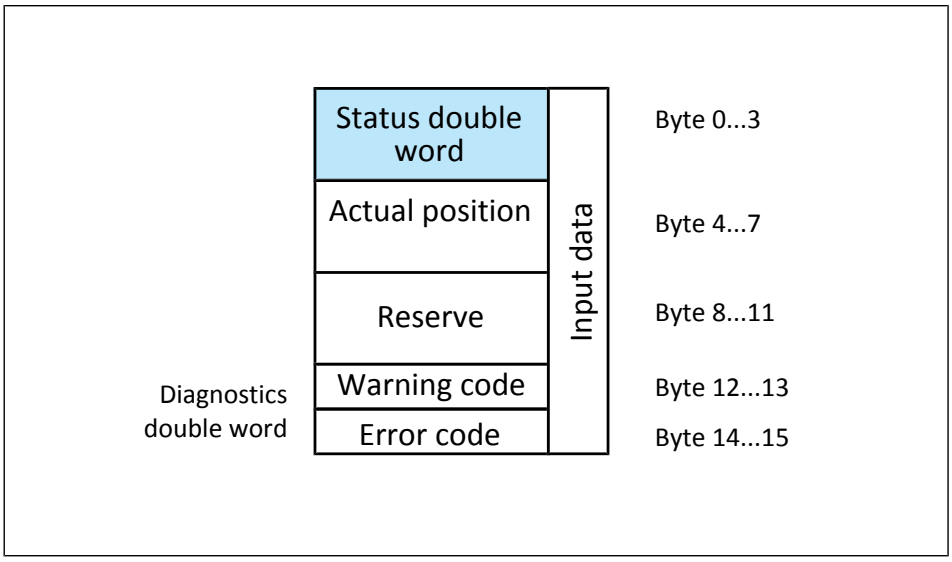
- In bytes 12 – 15 of the cyclical output data, the gripping force with which a workpiece is to be gripped is transmitted, Gripping a workpiece [▶ 24].
- The data format of the parameter is signed 32 bits and represents a value in millinewton [mN].

2.1.1.2 Cyclical input data

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

Data frame

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



Data frame of cyclical input data

In bytes 0 – 3 of the cyclical input data, the status double word is transmitted. In the following table, the structure of the status double word is shown. For a detailed description of the status double word, see chapter Status double word [▶ 60].

Word	Byte	Bit	Cyclical input data	For further information, see chapter
0	0	0	ready for operation	Status double word [▶ 60], Bit0
		1	control authority	Status double word [▶ 60], Bit1
		2	ready for shutdown	Status double word [▶ 60], Bit2
		3	not feasible	Status double word [▶ 60], Bit3
		4	success	Status double word [▶ 60], Bit4
		5	reserved	-
		6	warning	Status double word [▶ 60], Bit6
		7	error	Status double word [▶ 61], Bit7
	1	8	brake released	Status double word [▶ 61], Bit8
		9	softwarelimit	Status double word [▶ 61], Bit9
		10	referenced	Status double word [▶ 61], Bit10
		11	no part detected	Status double word [▶ 61], Bit11
		12	gripped	Status double word [▶ 61], Bit12
		13	position reached	Status double word [▶ 61], Bit13
		14	reserved	-
		15	reserved	-
1	2	16	reserved	-
		17	reserved	-
		18	reserved	-
		19	reserved	-
		20	reserved	-
		21	reserved	-
		22	reserved	-
		23	reserved	-
	3	24	reserved	-
		25	reserved	-
		26	reserved	-
		27	reserved	-
		28	reserved	-
		29	reserved	-
		30	reserved	-
		31	reserved	-

Actual position

- In bytes 4 – 7 of the cyclical input data, the current actual position of the module is transmitted, System parameters [▶ 29].
- The data format of the parameter is signed 32 bits and represents a value in micrometers [µm].

Reserve

- In input parameter 2, no user data is currently transmitted.

Diagnostic double word

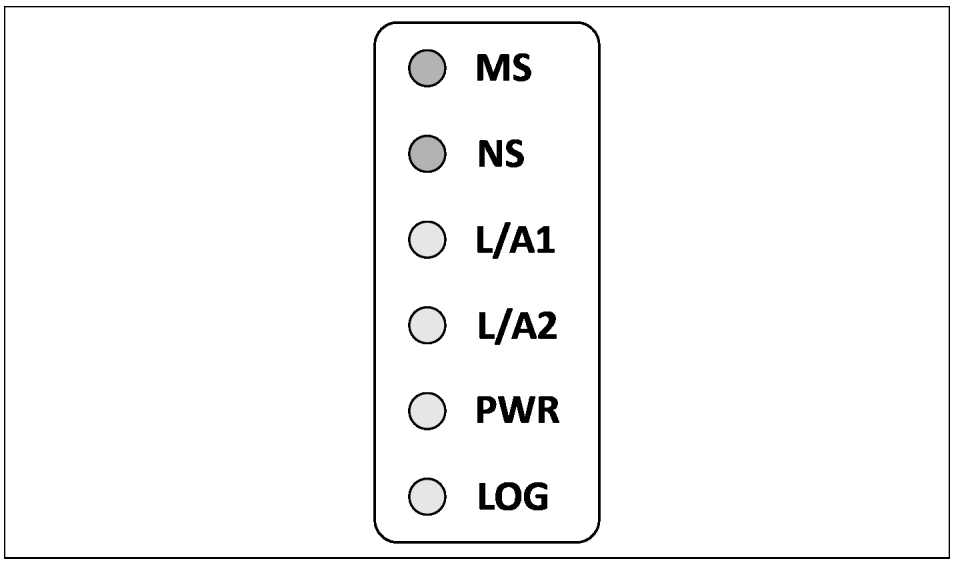
- In diagnostic double word, more detailed information about pending warnings and errors is transmitted.
- In bytes 12 – 13 of the cyclical input data, warning codes for the module are transmitted. In bytes 14 – 15 of the cyclical input data, error codes for the module are transmitted, Diagnostics [▶ 42].

2.1.2 Acyclical data exchange

Execution of the acyclic data exchange complies with the specifications of the PNO (Profibus User Organization, www.profibus.com). For all the required information pertaining to acyclic data exchange, see chapter System parameters [▶ 29].

2.2 LED status display

The LEDs indicate the user module states.



Arrangement of the LED status display

For further information on the display of module states, see chapter Status display via LED status display [▶ 62].

2.3 Other interfaces

Web server

An integrated web server can be used for commissioning and parameterization. The web server can be accessed with a browser via "http://IPADRESSE". "IPADRESSE" corresponds to the IP address of the module in the Ethernet network.

Service interface

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

2.4 Management of control logic

The control logic grants read and write permissions between the individual communication interfaces.

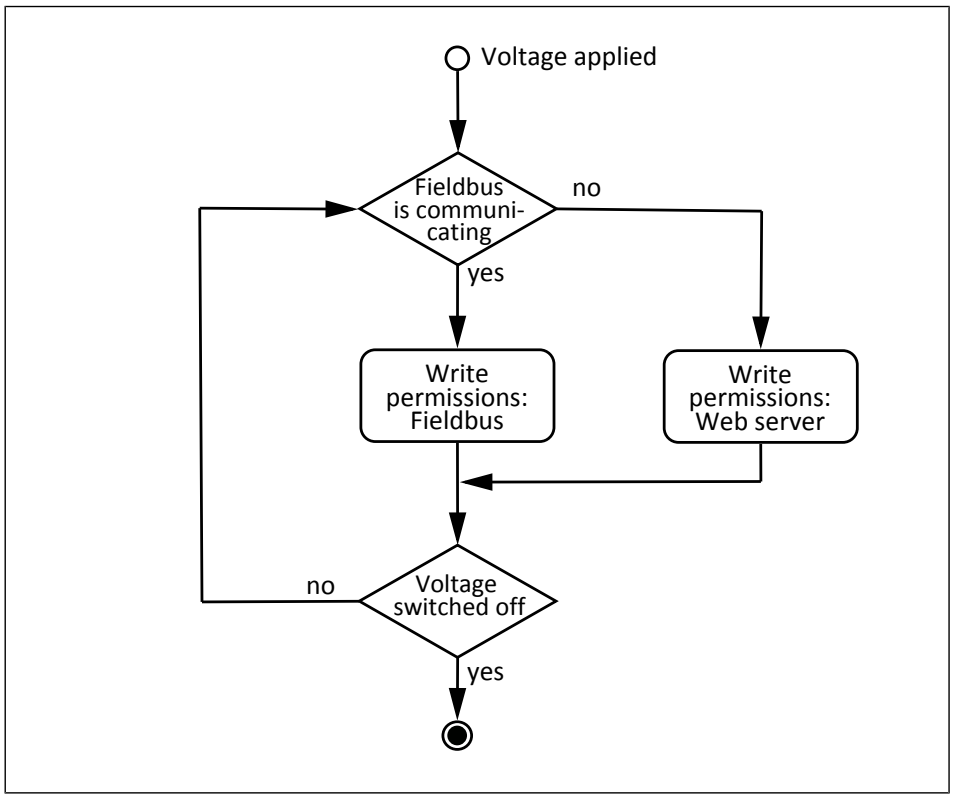
Read permissions

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 PROFINETcontrol has write permissions, this is indicated to the PLC by setting the status bit "control authority".



Management of the control logic

3 Module functions

3.1 Booting, shutting down and restarting the module

3.1.1 Booting

Short description

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

Trigger

The booting can be triggered on the hardware side by applying the logic supply voltage, or triggered on the software side by a restart, Restarting [▶ 18].

NOTE

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

Predictable diagnostic event

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

Diagnostic event	Diagnostic code *
The connected hardware is not recognized (anymore).	0xF5 - ERROR_BOOT_UNKOWN_HARDWARE_DETECTED
The internal memory is not recognized (anymore).	0xF6 - ERROR_BOOT_BLOCK_DEVICE_NOT_DETECTED
The communication module is not recognized (anymore).	0xF7 - ERROR_BOOT_ANYBUS_NOT_DETECTED

* For further information, see chapter Diagnostics [▶ 42].

3.1.2 Shutting down

Short description

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

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

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

Trigger

Preparations for a shut down are only permitted from within a defined system state and are triggered by setting the control bit "prepare for shutdown", Control double word [▶ 56]-Bit3.

System status

Preparations to shut down are only permitted from a stopped state. 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.

Predictable diagnostic event

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

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

* For further information, see chapter Diagnostics [▶ 42].

3.1.3 Restarting

Short description

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

Trigger

The restart is triggered by a setting of the control bit "softreset", Control double word [▶ 57]-Bit4.

This function is enabled via the parameter <enable_softreset>. If the content of this parameter is "0", then no restart is possible, Parameter list [▶ 40].

System status

Restarting is permitted from a stopped state 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.

Predictable diagnostic event

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

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

* For further information, see chapter Diagnostics [▶ 42].

3.2 Movement functions

3.2.1 Tip mode

Short description

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

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

Trigger

- In the positive direction, tip mode is triggered by setting the control bit "jog mode plus", Control double word [▶ 58]-Bit9.
- In the negative direction, tip mode is triggered by setting the control bit "jog mode minus, Control double word [▶ 57]-Bit8.

Movement parameter

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

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

Finish

Tip mode is terminated by the following options:

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

NOTE

The brake is applied with a delay time of 0.5 seconds after resetting the control bit "jog mode plus" or "jog mode minus". This makes a direct reversal in tip mode possible, without the brake having to switch several times. Transmitting other requests before this delay time expires is not permitted, Cyclical output data [▶ 9].

Predictable diagnostic event

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

Diagnostic event	Diagnostic code *
The lower software limit is reached in the referenced state.	0xD5 - ERROR_SOFT_LOW
The upper software limit is reached in the referenced state.	0xD6 - ERROR_SOFT_HIGH
Drive is already blocked at the start of movement.	0xF4 - ERROR _MOVE_BLOCKED
Drive blocked during movement.	0xF4 - ERROR _MOVE_BLOCKED
Sending a request that is not permitted.	0x94 - WARNING_CMD_ NOT_FEASIBLE
Movement terminated by user.	0xD9 - ERROR_FAST_STOP

* For further information, see chapter Diagnostics [▶ 42].

3.2.2 Referencing

Short description

The module defines its zero point during referencing, Reference point and zero point [▶ 6]. Successful referencing is indicated to the PLC by setting the status bit "referenced". When a reference movement is initiated, any status bit that may be set "referenced" is then reset.

Referencing is directed **inward** for modules with the firmware main version number 1.XX.

Trigger

Referencing is triggered by setting the "reference" control bit, Control double word [▶ 58]-Bit10.

Movement parameter

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

NOTE

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

Finish

Referencing is terminated by the following options:

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

Predictable diagnostic event

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

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

* For further information, see chapter Diagnostics [▶ 42].

3.2.3 Absolute positioning movement

Short description

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

A practical application example is described in chapter Application examples [▶ 51], example 1.

NOTE

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

Trigger

Absolute positioning is only permitted in the referenced state and is triggered by setting the control bit "move to absolute position", Control double word [▶ 59]-Bit13.

Movement parameter

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

- Position parameter
- Speed parameter

Absolute positioning is only permitted within the software limits.

Finish

Absolute positioning is terminated by the following options:

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

Predictable diagnostic event

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

Diagnostic event	Diagnostic code *
Positioning is taking too long.	0xF0 - ERROR_REFERENCE_ABORT_TIMEOUT
Lower software limit is reached.	0xD5 - ERROR_SOFT_LOW
Upper software limit is reached.	0xD6 - ERROR_SOFT_HIGH
Drive is already blocked at the start of movement.	0xF4 - ERROR _MOVE_BLOCKED
Drive blocked during movement.	0xF4 - ERROR _MOVE_BLOCKED
Sending a request that is not permitted.	0x94 - WARNING_CMD_NOT_FEASIBLE
Movement terminated by user.	0xD9 - ERROR_FAST_STOP

* For further information, see chapter Diagnostics [▶ 42].

3.2.4 Relative positioning movement

Short description

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

A practical application example is described in chapter Application examples [▶ 51], example 2.

NOTE

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

Trigger

Relative positioning is only permitted in a referenced state and is triggered by setting the control bit "move to relative position", Control double word [▶ 59]-Bit14.

Movement parameter

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

- Position parameter
- Speed parameter

Relative positioning is only permitted within the software limits.

Finish

Relative positioning is terminated by the following options:

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

Predictable diagnostic event

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

Diagnostic event	Diagnostic code *
Positioning is taking too long.	0xF1 - ERROR_MOVE_ABORT_TIMEOUT
Lower software limit is reached.	0xD5 - ERROR_SOFT_LOW
Upper software limit is reached.	0xD6 - ERROR_SOFT_HIGH
Drive is already blocked at the start of movement.	0xF4 - ERROR _MOVE_BLOCKED
Drive blocked during movement.	0xF4 - ERROR _MOVE_BLOCKED
Sending a request that is not permitted.	0x94 - WARNING_CMD_NOT_FEASIBLE
Movement terminated by user.	0xD9 - ERROR_FAST_STOP

* For further information, see chapter Diagnostics [▶ 42].

3.2.5 Controlled stop

Short description

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

Trigger

Controlled stops are triggered by setting the control bit "stop", Control double word [▶ 56]-Bit1.

Movement parameter

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

Finish

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

Predictable diagnostic event

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

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

* For further information, see chapter Diagnostics [▶ 42].

3.2.6 Terminating a movement

Short description

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

Trigger

Movement termination is triggered by resetting the control bit "fast stop", Control double word [▶ 56]-Bit0.

NOTE

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

3.3 Handling the workpiece

3.3.1 Gripping a workpiece

Short description

The module is able to grip workpieces in a gripping movement. A successful gripping movement is achieved by setting the status bit "gripped". An unsuccessful gripping movement is displayed by setting the status bit "no part detected". A practical application example is described in chapter Application examples [▶ 52], example 3 – 4.

NOTE

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

Trigger

Gripping workpieces is only permitted in a referenced state and is triggered by setting the control bit "grip work piece", Control double word [▶ 58]-Bit12.

Movement parameter

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

- Gripping force
- Gripping direction

Finish

Gripping workpieces is terminated by the following options:

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

Predictable diagnostic event

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

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

* For further information, see chapter Diagnostics [▶ 42].

3.3.2 Holding the workpiece

Short description

After a successful gripping movement, the gripped workpiece is held. The gripping force is maintained by the brake installed in the module.

Trigger

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

3.3.3 Releasing workpieces

Short description

The module is able to release gripped workpieces. In order to release workpieces, the module moves independently in the opposite direction of the last successful gripping command. Practical application examples are described in chapter Application examples [▶ 52], examples 5 – 6.

Trigger

Releasing workpieces is only permitted after a successful gripping movement and is triggered by setting the control bit "release work piece", Control double word [▶ 58] - Bit11.

NOTE

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

Movement parameter and parameterization

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

The traverse path in which the module moves relatively during the release is recorded in the parameter <release_delta_position> (Parameter list [▶ 33]). This parameter can be changed by the user. In delivery state, the value equals 5 mm.

Finish

Releasing workpieces is terminated by the following options:

- Target position 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.

Predictable diagnostic event

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

Diagnostic event	Diagnostic code *
The release is taking too long.	0xF1 - ERROR_MOVE_ABORT_ TIMEOUT
Drive is already blocked at the start of movement.	0xF4 - ERROR _MOVE_BLOCKED
Drive blocked during movement.	0xF4 - ERROR _MOVE_BLOCKED
Sending a request that is not permitted.	0x94 - WARNING_CMD_NOT_ FEASIBLE
Movement terminated by user.	0xD9 - ERROR_FAST_STOP

* For further information, see chapter Diagnostics [▶ 42].

3.3.4 Remove workpiece manually

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 in order to manually remove the workpiece.

NOTE

Because the user works directly on the module, the manual **removal** of workpieces **is only permitted in an emergency**. To ensure that the module does not perform any unpredictable 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", Control double word [▶ 57]-Bit5.

Movement parameter

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

Finish

The manual release of workpieces is terminated by the following options:

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

3.4 Additional functions

3.4.1 Position maintenance

After completing a movement, the module automatically changes to the position maintenance state when the brake is engaged.

3.4.2 Firmware updater

The user can update the installed software via the firmware updater. The module's firmware updater is integrated into the web server and can be applied via the web server.

The module's software can be updated via a tool. The tool can be downloaded via the SCHUNK website (https://schunk.com/de_de/services/downloads/software/).

3.4.3 Factory settings

The user can reset the module to factory settings. All parameters are reset to the respective default values or to default settings.

NOTICE

Material damage due to faulty usage!

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

4 System parameters

4.1 Value ranges

Value ranges

The following internal data types are used:

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

4.2 Parameter list

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

NOTE

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

0x0110 **0x0110 – <actual_control_autority>**

Short description:	The current owner of the control logic can be read out with this parameter.
Access rights:	ro (read permissions only)
Data type:	ENUM
Parameter value, min. – max.:	MIN_ENUM – 2
Unit/Format/ Enumeration:	0 = Service interface 1 = PROFINET control 2 = Web server
Factory settings:	2

0x0118 **0x0118 – <error_code>**

Short description:	The pending error code can be read out with this parameter.
Access rights:	ro (read permissions only)
Data type:	UINT8
Parameter value, min. – max.:	MIN_UINT8 – MAX_UINT8
Unit/Format/ Enumeration:	see chapter Error [▶ 45]
Factory settings:	N.A.

0x0120 **0x0120 – <warning_code>**

Short description:	The pending warning code can be read out with this parameter.
Access rights:	ro (read permissions only)
Data type:	UINT8
Parameter value, min. – max.:	MIN_UINT8 – MAX_UINT8
Unit/Format/ Enumeration:	see chapter Warnings [▶ 42]
Factory settings:	N.A.

0x0200 **0x0200 – <desired_position>**

Short description:	The default value of the absolute set position can be read out or written with this parameter.
Access rights:	rw (read and write permissions)
Data type:	FLOAT
Parameter value, min. – max.:	MIN_FLOAT – MAX_FLOAT
Unit/Format/ Enumeration:	Millimeter [mm]
Factory settings:	N.A.

0x0208 **0x0208 – <used_current_limit>**

Short description:	This parameter allows the maximum motor current used for the current movement to be read out.
Access rights:	ro (read permissions only)
Data type:	FLOAT
Parameter value, min. – max.:	MIN_FLOAT – MAX_FLOAT
Unit/Format/ Enumeration:	Ampere [A]
Factory settings:	N.A.

0x0210 **0x0210 – <desired_velocity>**

Short description:	The default value of the set speed can be read out or written with this parameter.
Access rights:	rw (read and write permissions)
Data type:	FLOAT
Parameter value, min. – max.:	MIN_FLOAT – MAX_FLOAT
Unit/Format/ Enumeration:	Millimeter per second [mm/s]
Factory settings:	N.A.

0x0220 **0x0220 – <desired_force>**

Short description:	The default value of the set gripping force can be read out or written with this parameter.
Access rights:	rw (read and write permissions)
Data type:	FLOAT
Parameter value, min. – max.:	MIN_FLOAT – MAX_FLOAT
Unit/Format/ Enumeration:	Newton [N]
Factory settings:	N.A.

0x0228 **0x0228 – <grip_direction>**

Short description:	The default value of the gripping direction can be read out or written with this parameter.
Access rights:	rw (read and write permissions)
Data type:	BOOL
Parameter value, min. – max.:	MIN_BOOL – MAX_BOOL
Unit/Format/ Enumeration:	0 = O.D. gripping 1 = I.D. gripping
Factory settings:	0

0x0230 **0x0230 – <actual_position>**

Short description:	This parameter can be used to read out the current actual position.
Access rights:	ro (read permissions only)
Data type:	FLOAT
Parameter value, min. – max.:	MIN_FLOAT – MAX_FLOAT
Unit/Format/ Enumeration:	Millimeter [mm]
Factory settings:	N.A.

0x0238 **0x0238 – <actual_velocity>**

Short description:	This parameter can be used to read out the current actual speed.
Access rights:	ro (read permissions only)
Data type:	FLOAT
Parameter value, min. – max.:	MIN_FLOAT – MAX_FLOAT
Unit/Format/ Enumeration:	Millimeter per second [mm/s]
Factory settings:	N.A.

0x0240 **0x0240 – <actual_current>**

Short description:	This parameter can be used to read out the current actual motor current.
Access rights:	ro (read permissions only)
Data type:	FLOAT
Parameter value, min. – max.:	MIN_FLOAT – MAX_FLOAT
Unit/Format/ Enumeration:	Ampere [A]
Factory settings:	N.A.

0x0540 **0x0540 – <release_delta_position>**

Short description:	This parameter can be used to read and write the path that the module uses relative to the current position in order to release a workpiece.
Access rights:	rw (read and write permissions)
Data type:	FLOAT
Parameter value, min. – max.:	1 – MAX_FLOAT
Unit/Format/ Enumeration:	Millimeter [mm]
Factory settings:	5

0x0600 **0x0600 – <minimum_position>**

Short description:	The lower software limit and therefore the smallest position value that can be approached by the module, can be read and written with this parameter.
Access rights:	rw (read and write permissions)
Data type:	FLOAT
Parameter value, min. – max.:	0 – 144
Unit/Format/ Enumeration:	Millimeter [mm]
Factory settings:	0

0x0608 **0x0608 – <maximum_position>**

Short description:	The upper software limit and therefore the largest position value that can be approached by the module, can be read and written with this parameter.
Access rights:	rw (read and write permissions)
Data type:	FLOAT
Parameter value, min. – max.:	1 – 115
Unit/Format/ Enumeration:	Millimeter [mm]
Factory settings:	115

0x0628 **0x0628 – <minimum_velocity>**

Short description:	The minimum movement speed with which the module can be moved can be read out with this parameter.
Access rights:	ro (read permissions only)
Data type:	FLOAT
Parameter value, min. – max.:	1 – MAX_FLOAT
Unit/Format/ Enumeration:	Millimeter per second [mm/s]
Factory settings:	15

0x0630 **0x0630 – <maximum_velocity>**

Short description:	The maximum positioning speed with which the module can be moved can be read out with this parameter.
Access rights:	ro (read permissions only)
Data type:	FLOAT
Parameter value, min. – max.:	15 – MAX_FLOAT
Unit/Format/ Enumeration:	Millimeter per second [mm/s]
Factory settings:	200

0x0658 **0x0658 – <minimum_motor_grip_force>**

Short description:	This parameter can be used to read out the minimum gripping force.
Access rights:	ro (read permissions only)
Data type:	FLOAT
Parameter value, min. – max.:	15 – 100
Unit/Format/ Enumeration:	Newton [N]
Factory settings:	25

0x0660 **0x0660 – <maximum_motor_grip_force>**

Short description:	This parameter can be used to read out the maximum gripping force.
Access rights:	ro (read permissions only)
Data type:	FLOAT
Parameter value, min. – max.:	20 – 120
Unit/Format/ Enumeration:	Newton [N]
Factory settings:	100

0x0800 **0x0800 – <minimum_error_motor_voltage>**

Short description:	The lower error limit value of the supply voltage of the drive can be read out with this parameter.
Access rights:	ro (read permissions only)
Data type:	FLOAT
Parameter value, min. – max.:	10 – 47
Unit/Format/ Enumeration:	Voltage [V]
Factory settings:	19.2

0x0808 **0x0808 – <maximum_error_motor_voltage>**

Short description:	The upper error limit value of the supply voltage of the drive can be read out with this parameter.
Access rights:	ro (read permissions only)
Data type:	FLOAT
Parameter value, min. – max.:	11 – 48
Unit/Format/ Enumeration:	Voltage [V]
Factory settings:	28.8

0x0810 **0x0810 – <minimum_error_logic_voltage>**

Short description:	The lower error limit value of the supply voltage of the main board can be read out with this parameter.
Access rights:	ro (read permissions only)
Data type:	FLOAT
Parameter value, min. – max.:	10 – 47
Unit/Format/ Enumeration:	Voltage [V]
Factory settings:	19.2

0x0818 **0x0818 – <maximum_error_logic_voltage>**

Short description:	The upper error limit value of the supply voltage of the main board can be read out with this parameter.
Access rights:	ro (read permissions only)
Data type:	FLOAT
Parameter value, min. – max.:	11 – 48
Unit/Format/ Enumeration:	Voltage [V]
Factory settings:	28.8

0x0820 **0x0820 – <minimum_error_logic_temperature>**

Short description:	The lower error limit value of the temperature of the main board can be read out with this parameter.
Access rights:	ro (read permissions only)
Data type:	FLOAT
Parameter value, min. – max.:	-20 – 89
Unit/Format/ Enumeration:	degrees Celsius [°C]
Factory settings:	-5

0x0820 **0x0820 – <maximum_error_logic_temperature>**

Short description:	The upper error limit value of the temperature of the main board can be read out with this parameter.
Access rights:	ro (read permissions only)
Data type:	FLOAT
Parameter value, min. – max.:	-19 – 90
Unit/Format/ Enumeration:	degrees Celsius [°C]
Factory settings:	75

0x0840 **0x0840 – <measured_temperature_logic>**

Short description:	The current measured temperature of the main board can be read out with this parameter.
Access rights:	ro (read permissions only)
Data type:	FLOAT
Parameter value, min. – max.:	MIN_FLOAT – MAX_FLOAT
Unit/Format/ Enumeration:	degrees Celsius [°C]
Factory settings:	N.A.

0x0870 **0x0870 – <measured_voltage_logic>**

Short description:	The current measured voltage supply of the main board can be read out with this parameter.
Access rights:	ro (read permissions only)
Data type:	FLOAT
Parameter value, min. – max.:	MIN_FLOAT – MAX_FLOAT
Unit/Format/ Enumeration:	Voltage [V]
Factory settings:	N.A.

0x0878 **0x0878 – <measured_voltage_motor>**

Short description:	The current measured voltage supply of the drive can be read out with this parameter.
Access rights:	ro (read permissions only)
Data type:	FLOAT
Parameter value, min. – max.:	MIN_FLOAT – MAX_FLOAT
Unit/Format/ Enumeration:	Voltage [V]
Factory settings:	N.A.

0x1000 **0x1000 – <device_serial_number>**

Short description:	The serial number of the module can be read out with this parameter.
Access rights:	ro (read permissions only)
Data type:	CHAR[16]
Parameter value, min. – max.:	N.A.
Unit/Format/ Enumeration:	ASCII-String
Factory settings:	N.A.

0x1008 **0x1008 – <order_number>**

Short description:	The order number of the module can be read out with this parameter.
Access rights:	ro (read permissions only)
Data type:	CHAR[16]
Parameter value, min. – max.:	N.A.
Unit/Format/ Enumeration:	ASCII-String
Factory settings:	N.A.

0x1100 **0x1100 – <software_build_date>**

Short description:	The creation date of the firmware version can be read out with this parameter.
Access rights:	ro (read permissions only)
Data type:	CHAR[12]
Parameter value, min. – max.:	N.A.
Unit/Format/ Enumeration:	ASCII-String
Factory settings:	N.A.

0x1108 **0x1108 – <software_build_time>**

Short description:	The creation time of the firmware version can be read out with this parameter.
Access rights:	ro (read permissions only)
Data type:	CHAR[9]
Parameter value, min. – max.:	N.A.
Unit/Format/ Enumeration:	ASCII-String
Factory settings:	N.A.

0x1110 **0x1110 – <software_version>**

Short description:	The version of the firmware can be read with this parameter.
Access rights:	ro (read permissions only)
Data type:	UINT16
Parameter value, min. – max.:	MIN_UINT16 – MAX_UINT16
Unit/Format/ Enumeration:	N.A.
Factory settings:	N.A.

0x1118 **0x1118 – <software_version_as_text>**

Short description:	The version of the software can be read as a text with this parameter.
Access rights:	ro (read permissions only)
Data type:	CHAR[22]
Parameter value, min. – max.:	N.A.
Unit/Format/ Enumeration:	ASCII-String
Factory settings:	N.A.

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

Short description:	This parameter can be used to enable the "Restart" function.
Access rights:	rw (read and write permissions)
Data type:	BOOL
Parameter value, min. – max.:	MIN_BOOL – MAX_BOOL
Unit/Format/ Enumeration:	0 = function switched off 1 = function switched on
Factory settings:	0

0x1400 **0x1400 – <system_uptime>**

Short description:	This parameter can be used to read the operating time that has elapsed since the last module (re)start.
Access rights:	ro (read permissions only)
Data type:	MIN_UINT32
Parameter value, min. – max.:	MIN_UINT32 – MAX_UINT32
Unit/Format/ Enumeration:	Seconds [s]
Factory settings:	N.A.

4.3 Parameter configuration

All system parameters for which the user has write permissions can be parameterized via acyclic data exchange Acyclical data exchange [▶ 13].

For further information on parameterization, see chapter Parameter list [▶ 30].

5 Diagnostics

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

Diagnostic events

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

5.1 Warnings

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

NOTE

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

Warning state

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

Warning code

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

Acknowledging

Warnings are both acknowledgeable and self-acknowledging.

By setting the control bit "acknowledge" the acknowledgment of an existing warning is triggered, Control double word [▶ 56]-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 code 0x9_

0x90 - WARNING_LOGIC_TEMP_LOW

Diagnostic event:	The measured logic temperature is too low.
Trigger:	The measured logic temperature is below the warning threshold.
Ability to acknowledge:	self-acknowledging

0x91 - WARNING_LOGIC_TEMP_HIGH

Diagnostic event:	The measured logic temperature is too high.
Trigger:	The measured logic temperature has exceeded the warning threshold.
Ability to acknowledge:	self-acknowledging

0x92 - WARNING_MOTOR_TEMP_LOW

Diagnostic event:	The measured motor temperature is too low.
Trigger:	The measured motor temperature is below the warning threshold.
Ability to acknowledge:	self-acknowledging

0x93 - WARNING_MOTOR_TEMP_HIGH

Diagnostic event:	The measured motor temperature is too high.
Trigger:	The measured motor temperature has exceeded the warning threshold.
Ability to acknowledge:	self-acknowledging

0x94 - WARNING_CMD_NOT_FEASIBLE

Diagnostic event:	The request cannot be executed.
Trigger:	The request transmitted with the cyclical output data to the module is impermissible and therefore cannot be executed.
Ability to acknowledge:	acknowledgeable /self-acknowledging

0x96 - WARNING _LOGIC_VOLTAGE_LOW

Diagnostic event:	The measured logic supply voltage is too low.
Trigger:	The measured logic supply voltage is below the warning value.
Ability to acknowledge:	self-acknowledging

0x97 - WARNING _LOGIC_VOLTAGE _HIGH

Diagnostic event:	The measured logic supply voltage is too high.
Trigger:	The measured logic supply voltage has exceeded the warning threshold.
Ability to acknowledge:	self-acknowledging

0x98 - WARNING _MOTOR_VOLTAGE _LOW

Diagnostic event:	The measured power supply voltage is too low.
Trigger:	The measured power supply voltage is below the warning value.
Ability to acknowledge:	self-acknowledging

0x99 - WARNING _MOTOR_VOLTAGE _HIGH

Diagnostic event:	The measured power supply voltage is too high.
Trigger:	The measured power supply voltage has exceeded the warning threshold.
Ability to acknowledge:	self-acknowledging

0x9D - WARNING_FACTORY_RESET_UNSUCCESSFUL

Diagnostic event:	Error when resetting to factory setting.
Trigger:	The module could not produce the factory settings due to an error.
Ability to acknowledge:	acknowledgeable /self-acknowledging

0x9F - WARNING _SHUTDOWN_NOT_PREPARED

Diagnostic event:	The shutdown was not requested.
Trigger:	The shutdown of the module was not requested.
Ability to acknowledge:	self-acknowledging

5.2 Error

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

NOTE

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

Error state

In an error state, the module is not longer ready for operation. By changing to an error state, the module is forced into a standstill and 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 code 0x6_

0x6C - ERROR_MOTOR_TEMP_LOW

Diagnostic event:	The measured motor temperature is too low.
Trigger:	The measured motor temperature is below the error threshold.
Ability to acknowledge:	requiring acknowledgment

0x6D - ERROR_MOTOR_TEMP_HIGH

Diagnostic event:	The measured motor temperature is too high.
Trigger:	The measured motor temperature has exceeded the error threshold.
Ability to acknowledge:	requiring acknowledgment

HEX code 0x7_

0x70 - ERROR_LOGIC_TEMP_LOW

Diagnostic event:	The measured logic temperature is too low.
Trigger:	The measured logic temperature is below the error threshold.
Ability to acknowledge:	requiring acknowledgment

0x71 - ERROR_LOGIC_TEMP_HIGH

Diagnostic event:	The measured logic temperature is too high.
Trigger:	The measured logic temperature has exceeded the error threshold.
Ability to acknowledge:	requiring acknowledgment

0x72 - ERROR_LOGIC_VOLTAGE_LOW

Diagnostic event:	The measured logic supply voltage is too low.
Trigger:	The measured logic supply voltage is below the error value.
Ability to acknowledge:	requiring acknowledgment

0x73 - ERROR_LOGIC_VOLTAGE _HIGH

Diagnostic event:	The measured logic supply voltage is too high.
Trigger:	The measured logic supply voltage has exceeded the error threshold.
Ability to acknowledge:	requiring acknowledgment

0x74 - ERROR_MOTOR_VOLTAGE _LOW

Diagnostic event:	The measured power supply voltage is too low.
Trigger:	The measured power supply voltage is below the error value.
Ability to acknowledge:	requiring acknowledgment

0x75 - ERROR_MOTOR_VOLTAGE _HIGH

Diagnostic event:	The measured power supply voltage is too high.
Trigger:	The measured power supply voltage has exceeded the error threshold.
Ability to acknowledge:	requiring acknowledgment

HEX code 0x8_

0x8A - ERROR_ENCODER_PHASESHIFT

Diagnostic event:	The phase shift of the encoder signals is deviating.
Trigger:	The phase shift between the sine and cosine signal deviates from 90°.
Ability to acknowledge:	not acknowledgeable

0x8B - ERROR_ENCODER_SINE_LOW

Diagnostic event:	The measured sine wave signal is too low.
Trigger:	The measured sine signal of the encoder has fallen below the lower exact limit.
Ability to acknowledge:	not acknowledgeable

0x8C - ERROR_ENCODER_SINE_HIGH

Diagnostic event:	The measured sine wave signal is too high.
Trigger:	The measured sine signal of the encoder has exceeded the upper exact limit.
Ability to acknowledge:	not acknowledgeable

0x8D - ERROR_ENCODER_COSINE_LOW

Diagnostic event:	The measured cosine wave signal is too low.
Trigger:	The measured cosine signal of the encoder has fallen below the lower exact limit.
Ability to acknowledge:	not acknowledgeable

0x8E - ERROR_ENCODER_COSINE_HIGH

Diagnostic event:	The measured cosine wave signal is too high.
Trigger:	The measured cosine signal of the encoder has exceeded the upper exact limit.
Ability to acknowledge:	not acknowledgeable

0x8F - ERROR_ENCODER_SHORTCUT

Diagnostic event:	The encoder signals are identical.
Trigger:	The values of the measured sine and cosine signals are too similar.
Ability to acknowledge:	not acknowledgeable

HEX code 0xD_

0xD5 - ERROR_SOFT_LOW

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

0xD6 - ERROR_SOFT_HIGH

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

0xD9 - ERROR_FAST_STOP

Diagnostic event:	A fast stop was triggered.
Trigger:	The diagnostics or the user triggered a quick stop.
Ability to acknowledge:	requiring acknowledgment

0xDE - ERROR_CURRENT

Diagnostic event:	The maximum current has been exceeded.
Trigger:	The maximum permissible motor current was exceeded for 500 milliseconds.
Ability to acknowledge:	requiring acknowledgment

HEX code 0xF_

0xF0 - ERROR_REFERENCE_ABORT_TIMEOUT

Diagnostic event:	Referencing could not be performed within a defined period of time.
Trigger:	Too much time was needed for referencing.
Ability to acknowledge:	requiring acknowledgment

0xF1 - ERROR_MOVE_ABORT_TIMEOUT

Diagnostic event:	Positioning could not be performed within a defined period of time.
Trigger:	Too much time was needed for positioning.
Ability to acknowledge:	requiring acknowledgment

0xF2 - ERROR_NO_REFERENCE

Diagnostic event:	The module could not find a reference point.
Trigger:	The module cannot find a reference point and therefor could not set a zero point.
Ability to acknowledge:	requiring acknowledgment

0xF4 - ERROR _MOVE_BLOCKED

Diagnostic event:	The drive was blocked.
Trigger:	The module has detected a blocked drive as an error.
Ability to acknowledge:	requiring acknowledgment

0xF5 - ERROR_BOOT_UNKOWN_HARDWARE_DETECTED

Diagnostic event:	Unknown hardware connected.
Trigger:	When booting the module, the firmware detected that unsupported hardware is connected.
Ability to acknowledge:	not acknowledgeable

0xF6 - ERROR_BOOT_BLOCK_DEVICE_NOT_DETECTED

Diagnostic event:	The internal memory is not recognized.
Trigger:	The internal memory was not recognized when booting the module.
Ability to acknowledge:	not acknowledgeable

0xF7 - ERROR_BOOT_ANYBUS_NOT_DETECTED

Diagnostic event:	The communication block is not recognized.
Trigger:	The communication block was not recognized when booting the module.
Ability to acknowledge:	not acknowledgeable

0xF8 - ERROR_BOOT_HARDWARE_NOT_SUPPORTED_BY_FIRMWARE

Diagnostic event:	Hardware and firmware are incompatible.
Trigger:	The firmware has detected that the firmware does not support the existing hardware.
Ability to acknowledge:	not acknowledgeable

0xFA - ERROR_PREPARE_FOR_SHUTDOWN_NOT_FEASIBLE

Diagnostic event:	Preparing to shut down is not feasible.
Trigger:	Preparing to shut down could not be performed.
Ability to acknowledge:	requiring acknowledgment

6 Appendix

6.1 Application examples

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

EXAMPLE 1 **Absolute positioning movement**

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

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

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

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

For further information, see chapter Absolute positioning movement [▶ 21].

EXAMPLE 2 **Relative positioning movement**

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

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

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

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

For further information, see chapter Relative positioning movement [▶ 22].

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

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

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

→ The module then performs a gripping movement as an I.D. gripper. The workpiece is gripped with 60 N and the status bit "gripped" is set.

For further information, see chapter Gripping a workpiece [▶ 24].

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

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

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

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

For further information, see chapter Gripping a workpiece [▶ 24].

EXAMPLE 5 **Releasing workpieces (1)**

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

- Control bit "release work piece" is set

The module knows to grip a workpiece from the inside and then, taking into account the software limits, moves -5 mm from the current actual position.

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

For further information, see chapter Releasing workpieces [▶ 25].

EXAMPLE 6 **Releasing workpieces (2)**

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

- Control bit "release work piece" is set

The module knows to grip a workpiece from the outside and then, taking into account the software limits, moves 10 mm from the current actual position.

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

For further information, see chapter Releasing workpieces [▶ 25].

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

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

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

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

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

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

→ Absolute positioning is triggered.

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

→ The movement ends when the target position is reached.

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

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

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

→ Absolute positioning is triggered.

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

→ Absolute positioning is ended with a controlled stop.

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

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

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

→ Absolute positioning is triggered.

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

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

→ Error: Impermissible bit combination during active movement.

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

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

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

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

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

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

→ Error: Impermissible bit combination

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

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

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

→ The status bit "not feasible" is set.

→ Error: The module is not referenced.

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

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

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

→ The status bit "not feasible" is set.

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

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

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

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

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

→ Error: Impermissible bit combination during active movement.

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

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

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

→ The status bit "not feasible" is set.

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

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

6.2 Control double word

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

Bit 0 - fast stop

Edge change or status bit	Module reaction
0 -> 1 or 1	no reaction
1 -> 0 or 0	The module performs a quick stop, Terminating a movement [▶ 23].

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

- all bits

Bit1 - stop

Edge change	Module reaction
0 -> 1	The module performs a controlled stop, Controlled stop [▶ 23].
1 -> 0	no reaction

If this bit changes to the state 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

Bit2 - acknowledge

Edge change	Module reaction
0 -> 1	The module tries to acknowledge all existing warnings and errors, Warnings [▶ 42], Error [▶ 45].
1 -> 0	no reaction

If this bit changes to the state 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

Bit3 - prepare for shutdown

Edge change	Module reaction
0 -> 1	The module is preparing for shutdown, Shutting down [▶ 17].
1 -> 0	no reaction

If this bit changes to the state 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

Bit4 - softreset

Edge change	Module reaction
0 -> 1	The module is restarted on the software side, Restarting [▶ 18].
1 -> 0	no reaction

If this bit changes to the state 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

Bit5 - release brake

Edge change	Module reaction
0 -> 1	The brake is released in order to manually remove a workpiece, Remove workpiece manually [▶ 27].
1 -> 0	no reaction

If this bit changes to the state 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

Bit6 - reserved

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

Bit7 - grip direction

Status bit	Module reaction
0	Defines an O.D. gripper
1	Defines an I.D. gripper

Bit8 - jog mode minus

Edge change	Module reaction
0 -> 1	The module executes a movement in the negative direction of movement, Tip mode [▶ 19].
1 -> 0	no reaction

If this bit changes to the state 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

Bit9 - jog mode plus

Edge change	Module reaction
0 -> 1	The module executes a movement in the positive direction of movement, Tip mode [▶ 19].
1 -> 0	no reaction

If this bit changes to the state 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

Bit10 - reference

Edge change	Module reaction
0 -> 1	The module executes a reference run, Referencing [▶ 20].
1 -> 0	no reaction

If this bit changes to the state 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

Bit11 - release work piece

Edge change	Module reaction
0 -> 1	The module releases a workpiece, Releasing workpieces [▶ 25].
1 -> 0	no reaction

If this bit changes to the state 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

Bit12 - grip work piece

Edge change	Module reaction
0 -> 1	The module performs a gripping movement, Gripping a workpiece [▶ 24]
1 -> 0	no reaction

If this bit changes to the state 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

Bit13 - move to absolute position

Edge change	Module reaction
0 -> 1	The module performs a positioning movement to an absolute position, Absolute positioning movement [▶ 21].
1 -> 0	no reaction

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

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

Bit 14 - move to relative position

Edge change	Module reaction
0 -> 1	The module performs a positioning movement to a relative position, Referencing [▶ 20].
1 -> 0	no reaction

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

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

Bit 15 – 31 - reserved

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

6.3 Status double word

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

Bit0 - ready for operation

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

Bit1 - control authority

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

Bit2 - ready for shutdown

Status bit	Module feedback
0	No information is reported.
1	The module is ready to be shut down.

Bit3 - not feasible

Status bit	Module feedback
0	No information is reported.
1	The request sent to the module is not feasible.

Bit4 - success

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

Bit5 - reserved

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

Bit6 - warning

Status bit	Module feedback
0	No information is reported.
1	The issue related to a warning is displayed.

Bit7 - error

Status bit	Module feedback
0	No information is reported.
1	The issue related to an error is displayed.

Bit8 - brake released

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

Bit9 - softwarelimit

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

Bit10 referenced

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

Bit11 - no part detected

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

Bit12 - gripped

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

Bit13 - position reached

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

Bit14 – 31 - reserved

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

6.4 Status display via LED status display

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

LED	Designation	Color	Function
MS	Module status	Red / Green	**LED off:** No supply voltage present. The product is in setup or NW_Init status.
			Lights up green: The product is in normal operating mode.
			LED flashes green once: The product is currently processing diagnostics processes.
			Lights up red: severe fault. The product is not ready for operation.
			LED flashing alternately red and green: A firmware update is in progress. **WARNING! Do not switch off the supply voltage otherwise the product may be damaged permanently.**
NS	Network status	Red / Green	**LED off:** No supply voltage is present and/or no connection to a PROFINET control.
			Lights up green: Connection to a PROFINET control present and PROFINET control is in run mode.
			LED flashing green once: Connection to a PROFINET control present and PROFINET control is in stop mode. The IRT synchronization is not yet finished.
			LED flashing green permanently: The network participant is in identification mode.
			Lights up red: severe network fault present.
			LED lights up red 1x: The station name is not known.
			LED lights up red 2x: The IP address is not known.
			LED lights up red 3x: A configuration fault is present.
L/A1	Connection and communication of PROFINET control	Green	**LED off:** Connection inactive, communication inactive
			LED lights up green: Connection active, communication inactive
			LED flickering: Connection active, communication active
L/A2	Connection and communication of PROFINET control	Green	**LED off:** Connection inactive, communication inactive
			LED lights up green: Connection active, communication inactive
			LED flickering: Connection active, communication active
PWR	Supply power	Green	**LED off:** No supply voltage power is present.
			LED lights up green: Supply voltage power is present.
LOG	Logic supply	Green	**LED off:** No logic supply voltage present.
			LED lights up green: Logic supply voltage power is present.

Notes