

# **Commissioning instructions, Firmware 5.3.1**

## **EGU with Modbus RTU interface**

### **Universal gripper, electric**

Translation of original commissioning  
instructions

## Imprint

### Copyright:

This manual is protected by copyright. The author is SCHUNK SE & Co. KG.  
All rights reserved.

### Technical changes:

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

**Document number:** 1514034-EGU-MB-FW5.3.1

**Version:** 01.00 | 23/09/2025 | 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

Changes from software release V5.3.0 to V5.3.1 .....	5
Changes from software release V5.2 to V5.3.....	6
<b>1 General .....</b>	<b>7</b>
1.1 About this document .....	7
1.2 Definitions.....	8
1.2.1 Minimum and maximum position.....	8
1.2.2 Directions of movement and gripping.....	9
1.2.3 Gripping modes .....	10
1.2.4 Gripping force and position maintenance (GPE) .....	11
1.2.5 Zero point .....	11
<b>2 Communication .....</b>	<b>12</b>
2.1 Data exchange.....	12
2.1.1 Cyclical data exchange .....	12
2.1.2 Acyclical data exchange Modbus RTU .....	21
<b>3 Module functions .....</b>	<b>23</b>
3.1 Booting, shutting down and restarting .....	23
3.1.1 Booting and establishing operational readiness.....	23
3.1.2 Shutting down.....	24
3.1.3 Restart.....	26
3.2 Movement functions .....	27
3.2.1 Tip mode .....	27
3.2.2 Absolute positioning movement .....	28
3.2.3 Relative positioning movement .....	30
3.2.4 Controlled stop .....	32
3.2.5 Terminating a movement.....	33
3.3 Handling a workpiece .....	33
3.3.1 Workpiece gripping (simple gripping movement) .....	33
3.3.2 Workpiece gripping at expected position (combined gripping movement).....	35
3.3.3 Workpiece re-gripping .....	40
3.3.4 Workpiece loss detection .....	42
3.3.5 Workpiece release .....	43
3.3.6 Remove workpiece manually .....	44
3.4 Additional functions .....	45
3.4.1 Zero point offset .....	45
3.4.2 Handshake .....	46
3.4.3 Brake test.....	47
3.4.4 LifeSign.....	47
3.4.5 Repeat control command with optimized time .....	48
3.4.6 Factory settings .....	48

<b>4 System parameters</b>	<b>49</b>
4.1 Value ranges	49
4.2 Parameter list	49
<b>5 Start-up</b>	<b>67</b>
5.1 Safety	67
5.2 System integration	67
5.3 Commissioning Modbus RTU	68
5.3.1 General communication settings	68
5.3.2 Data-specific communication settings	69
5.3.3 Data exchange	70
<b>6 Diagnostics</b>	<b>75</b>
6.1 Warnings	75
6.2 Error	77
<b>7 Appendix</b>	<b>81</b>
7.1 Application examples	81
7.2 Control double word	90
7.3 Status double word	93
7.4 Additional code when warning is present WRN_NOT_FEASIBLE	96
7.5 Software copyright notices	99

## Changes from software release V5.3.0 to V5.3.1

The following enhancements/improvements have been made in comparison to software release V5.3.0:

- The behavior when releasing elastic workpieces has been improved.
- Improved jog behavior: Overshoot has been reduced when leaving the software limits.
- Error correction (bug fixing):
  - Incorrectly generated absolute encoder error "0xB9" at gripper start has been fixed.
  - In rare cases, pre-positioning was skipped during "Grip workpiece at expected position". This has been fixed.
  - The "Restart" module function is also executed correctly in the "ready for shutdown" state.
  - When the "Switch off" module function is executed, the control of the module is deactivated and, if present, the brake is applied.
  - The status bits "workpiece pre-grip started" and "workpiece lost" are reset correctly after a workpiece loss.
  - The status bits "workpiece pre-grip started" and "workpiece gripped" are reset correctly at the end of the re-gripping process.
  - The setting of the reserved control bit 15 is ignored.
  - Incremental encoder errors are saved in the diagnostic memory.

## Changes from software release V5.2 to V5.3

The following enhancements/improvements have been made in comparison to software release V5.2:

- Default value parameter <wp\_lost\_dst> (► 4.2 [□ 53]) was changed to 2 mm
- Time sequence for gripping the workpiece at the expected position has been optimized
- Name of the additional code 0x28 when the warning WRN\_NOT\_FEASIBLE is present has been changed
- Detection of MODBUS communication interruptions has been made more tolerant (this has improved compatibility with UR Polycope controllers)
- Error correction (bug fixing):
  - Status bit "workpiece pre-grip started" is reset correctly after a function is triggered
  - After triggering "release workpiece" during regripping, the error event ERR\_MOVE\_BLOCKED is no longer displayed (only occurred in rare cases)
  - Speed oscillation after restart is fixed (only occurred in rare cases)
  - Procedure for determining the motor temperature has become more robust (only occurred in rare cases)

# 1 General

## 1.1 About this document

This manual describes the commissioning as well as the operating and parameterization options for an electric gripper EGU with the following interface:

- Modbus RTU (MB)

### Validity

This version of the manual describes the functions of firmware versions that have the version number 5.3.1

The firmware version can be read out. Information on the corresponding parameter can be found in section ▶ 4.2 [ 64].

### Conventions

The following conventions apply to this guide:

- The gripper is hereinafter referred to as the "module".
- Actions initiated by the user that the module is to perform are hereafter referred to as a "control commands".
- Identification of parameters: <parameter>
- Identification of events: WARNING
- Page number in references: [ ▶ 4]

**NOTE:** The illustrations in this manual are intended to provide a basic understanding and may deviate from the actual version.

### Abbreviations

The following abbreviations are used:

- GPE: Gripping force and position maintenance

### 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/downloads](https://www.schunk.com/downloads).

---

### NOTE

Commissioning instructions for older firmware versions can be requested from SCHUNK.

---

## 1.2 Definitions

### 1.2.1 Minimum and maximum position

The parameters <min\_pos> and <max\_pos> define the position limits within which movements are permitted.

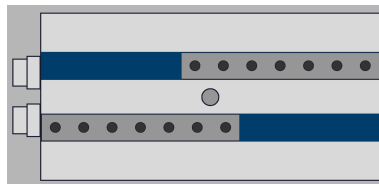
The value of the parameter <min\_pos> corresponds to the *smallest* position value that can be approached.

The value of the parameter <max\_pos> corresponds to the *largest* position value that can be approached.

In the delivery state, the position values of the parameters <min\_pos> and <max\_pos> correspond to the positions of the base jaws shown below.

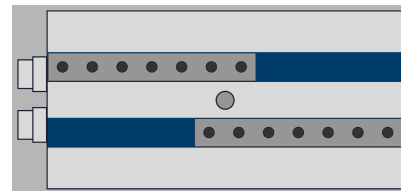
#### Minimum position

In the top view, the upper base jaw is on the right and the lower one on the left.



#### Maximum position

In the top view, the upper base jaw is on the left and the lower one on the right.



In the delivery state, this position corresponds to the **zero point** of the module.

---

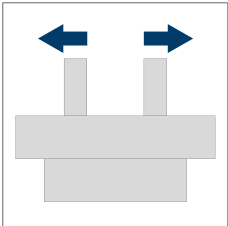
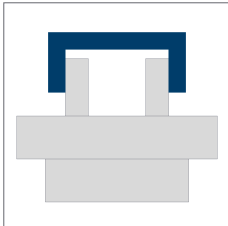
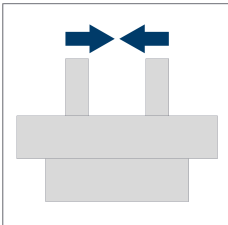
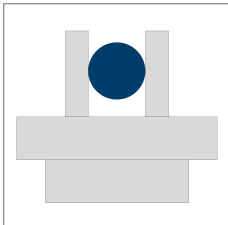
#### NOTE

If the gripper fingers are moved beyond the traversing range, the module switches to the error state and reports back the diagnostic event ERR\_SOFT\_LOW or ERR\_SOFT\_HIGH.

---

## 1.2.2 Directions of movement and gripping

Directions of movement and gripping are shown below.

Directions of movement	Directions of gripping
<p><b>outward</b></p> <p>The movement from the minimum to the maximum position value corresponds to the <i>outward</i> movement.</p> 	<p><b>I.D. gripping</b></p> <p>By moving outward, a workpiece can be gripped from the <i>inside</i>, hence the name <i>I.D. gripping</i>.</p> 
<p><b>inward</b></p> <p>The movement from the maximum to the minimum position value corresponds to the <i>inward</i> movement.</p> 	<p><b>O.D. gripping</b></p> <p>By moving inward, a workpiece can be gripped from the <i>outside</i>, hence the name <i>O.D. gripping</i>.</p> 

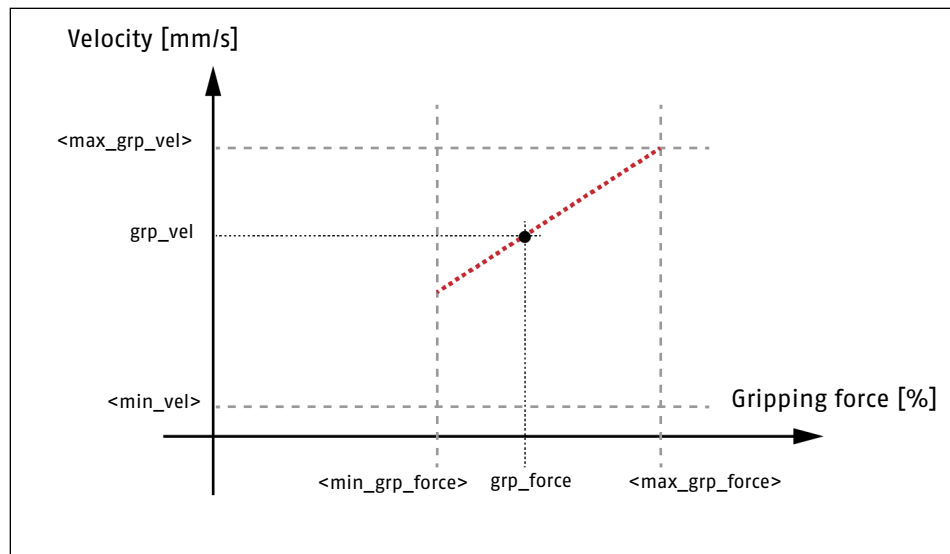
### 1.2.3 Gripping modes

The module provides different gripping modes for gripping workpieces:

- BasicGrip
- StrongGrip

#### BasicGrip

BasicGrip is the default gripping mode for the module. The module calculates the gripping velocity with which the workpiece is gripped, depending on the gripping force transferred. This reduces the force pulse generated when gripping the workpiece.



*Gripping velocity as a function of gripping force*

#### Calculation of the gripping velocity in BasicGrip mode

Gripping velocity = gripping force [%] \* <max\_grp\_vel> [mm/s]

#### StrongGrip

This mode is only available for modules of variant "M" (with GPE), ▶ 1.2.4 [11].

The StrongGrip mode can be used to grip workpieces with a gripping force greater than 100%.

In this mode, the module always uses the maximum gripping velocity <max\_grp\_vel>.

Due to technical reasons, the module can re-grip for a maximum of 2 seconds with this function before the GPE is automatically activated.

### 1.2.4 Gripping force and position maintenance (GPE)

Workpieces and positions are held by the module's drive control as standard. The "M" variant modules have gripping force and position maintenance (GPE). When sending control commands with these modules, you can specify whether workpieces and positions are to be held by the drive control *or* by the GPE.

Selecting the type of workpieces and positions to be held is done via the control bit "Activate grip force and position maintenance", ▶ 7.2 [📄 93].

---

#### NOTE

For modules without GPE *the* control bit "Activate grip force and position maintenance" always has to be 0. When trying to activate the GPE, the status bit "not feasible" and the diagnostic code WRN\_NOT\_FEASIBLE are reported back including additional code.

---

### 1.2.5 Zero point

The zero point of the module corresponds to a position of the gripper fingers at which the position value 0 mm is output. The zero point can be individually adapted to the conditions within an application, ▶ 3.4.1 [📄 45].

## 2 Communication

### 2.1 Data exchange

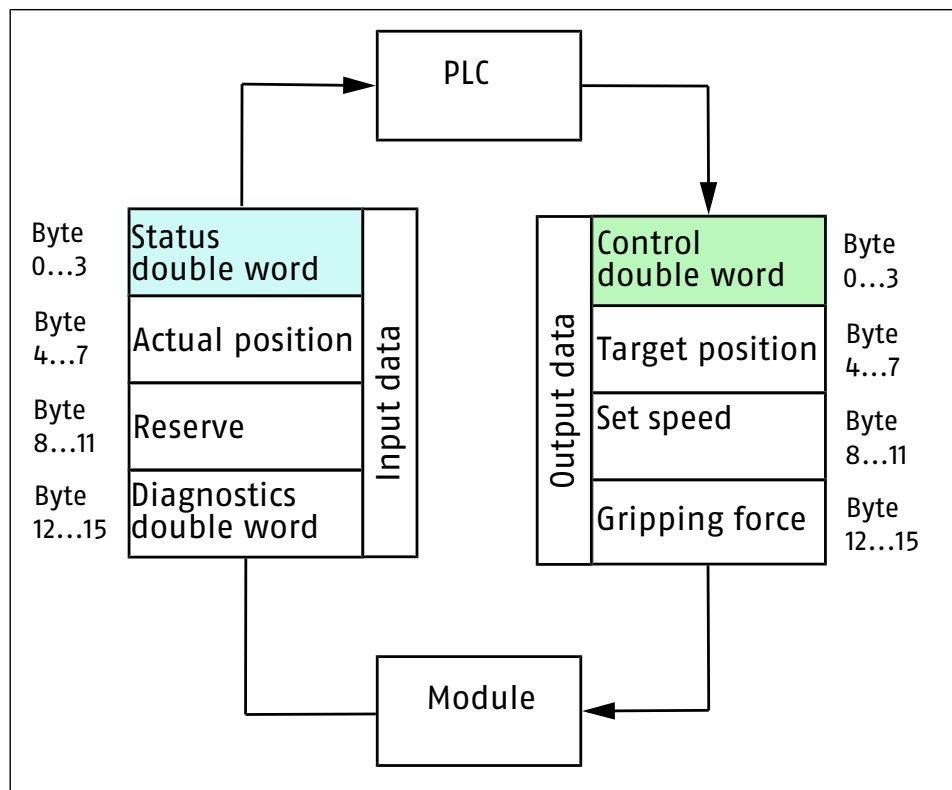
Integrated fieldbus interfaces can be used to exchange data cyclically and acyclically between the module and the controller.

#### NOTE

If communication between the module and the controller is interrupted, e.g. by a cable break or by changing the controller to the "Stop" status, the module performs a quick stop. The diagnostic event ERR\_COMM\_LOST is also reported back.

#### 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

Modbus RTU does **not** distinguish between cyclical and acyclical data exchange. There are only the described register accesses, ▶ 2.1.2 [21].

For cyclic communication, the cyclic parameters <plc\_sync\_input> (▶ 4.2 [50]) and <plc\_sync\_output> (▶ 4.2 [65]) must be explicitly accessed repeatedly for reading or writing.

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 control commands to the module. For practical application examples, see chapter ▶ 7.1 [□ 81].

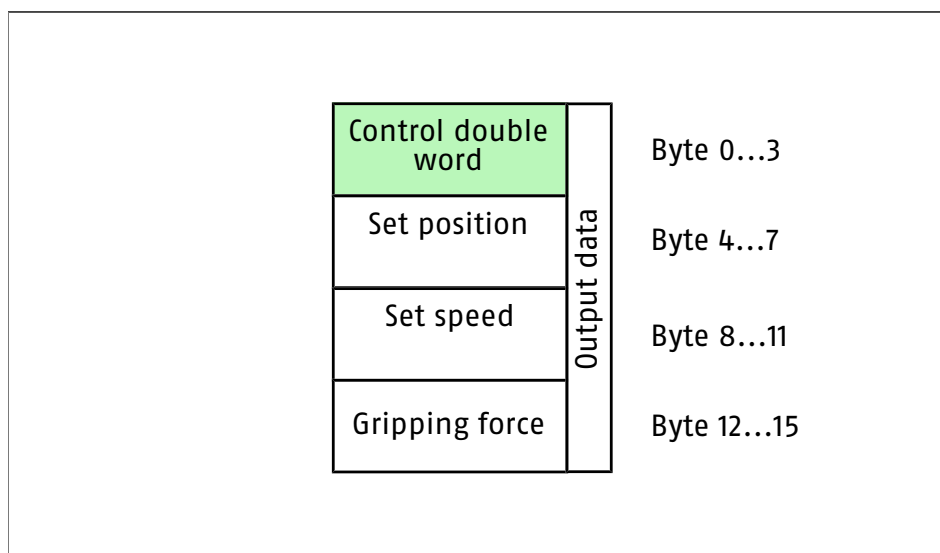
#### Execution of the control commands

Control commands sent to the module may be permissible or impermissible.

- Permissible control commands are executed by the module.
- Impermissible control commands are not executed. The PLC is displayed by setting the status bit "not feasible".  
Furthermore, the warning WRN\_NOT\_FEASIBLE including additional code is transmitted via the diagnostics double word, ▶ 6.1 [□ 76].

#### 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



## Overview of control double word

Word	Byte	Bit	Cyclical output data
0	0	0	<p>▶ fast stop [ 90]</p> <p>EN – short: fast stop</p> <p>DE – short: Schnellstopp</p>
		1	<p>▶ stop [ 90]</p> <p>EN – short: stop</p> <p>DE – short: Stopp</p>
		2	<p>▶ acknowledge [ 90]</p> <p>EN – short: ack</p> <p>DE – short: Quittieren</p>
		3	<p>▶ prepare for shutdown [ 90]</p> <p>EN – short: prep shutdown</p> <p>DE – short: Herunterfahren vorbereiten</p>
		4	<p>▶ softreset [ 90]</p> <p>EN – short: softreset</p> <p>DE – short: Neustart</p>
		5	<p>▶ release for manual movement [ 90]</p> <p>EN – short: release manual movement</p> <p>DE – short: Man. Bwg. freigeben</p>
		6	<p>▶ repeat command toggle [ 91]</p> <p>EN – short: rpt cmd tgl</p> <p>DE – short: Kdo. wiederh.</p>
		7	<p>▶ grip direction [ 91]</p> <p>EN – short: grip dir</p> <p>DE – short: Greifrichtung</p>
0	1	8	<p>▶ jog mode negative [ 91]</p> <p>EN – short: jog -</p> <p>DE – short: Tipp -</p>
		9	<p>▶ jog mode positive [ 91]</p> <p>EN – short: jog +</p> <p>DE – short: Tipp +</p>
		10	reserved
		11	<p>▶ release workpiece [ 91]</p> <p>EN – short: release wp</p> <p>DE – short: Werkst. freigeben</p>
		12	<p>▶ grip workpiece [ 92]</p> <p>EN – short: grp wp</p> <p>DE – short: Werkst. greifen</p>
		13	<p>▶ move to absolute position [ 92]</p>

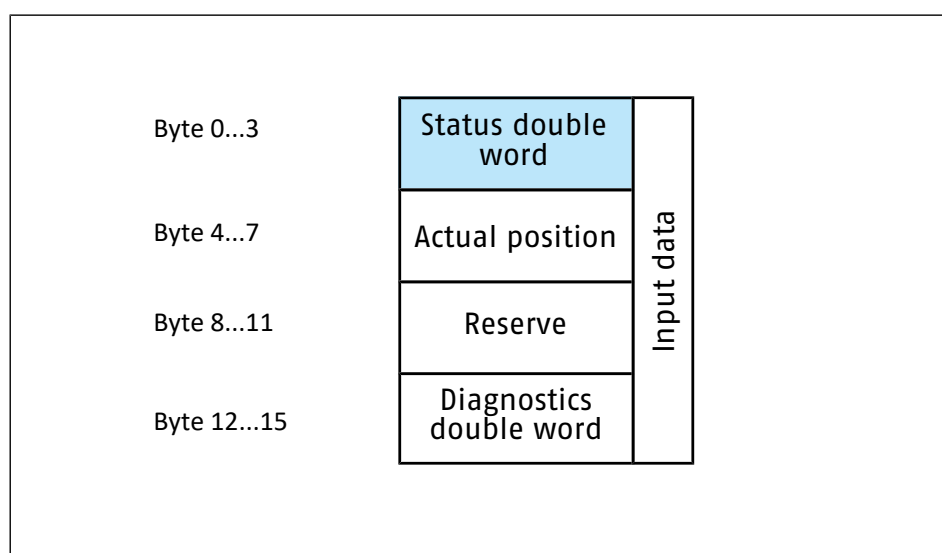
Word	Byte	Bit	Cyclical output data
			EN – short: pos absolute DE – short: Pos. absolut
		14	▶ move to relative position [ 92] EN – short: pos relative DE – short: Pos. relativ
		15	reserved
1	2	16	▶ grip workpiece at expected position [ 92] EN – short: grp wp at pos DE – short: Werkst. greifen an erw. Pos.
		17	reserved
		18	reserved
		19	reserved
		20	reserved
		21	reserved
		22	reserved
		23	reserved
1	3	24	reserved
		25	reserved
		26	reserved
		27	reserved
		28	reserved
		29	reserved
		30	▶ brake test [ 93] EN – short: brake test DE – short: Bremsentest
		31	▶ Activate grip force and position maintenance [ 93] EN – short: activate GPE DE – short: GPE aktivieren

- Target position**
  - In bytes 4 – 7 of the cyclical output data, data is transmitted that is used for positioning purposes, ▶ 4.2 [49].
  - The data format of the parameter is *signed 32 bits* and represents a value in micrometers [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, ▶ 4.2 [49].
  - The data format of the parameter is *signed 32 bits* and represents a value in micrometers 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, ▶ 3.3.1 [33].
  - The data format of the parameter is *signed 32 bits* and represents a value in percent [%]. The percentage value refers to the parameter <max\_grp\_force>, ▶ 4.2 [58].  
If an invalid gripping force is set, an entry is created in the diagnostic memory which also refers to the force limits in newtons.

### 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



## Overview of status double word

Word	Byte	Bit	Cyclical input data
0	0	0	▶ ready for operation [ 93] EN – short: ready for op DE – short: Betriebsbereit
		1	▶ control authority fieldbus [ 93] EN – short: ctrl authority fb DE – short: Feldbus
		2	▶ ready for shutdown [ 93] EN – short: ready for sd DE – short: Abschaltbereit
		3	▶ not feasible [ 93] EN – short: not feasible DE – short: Nicht durchführb.
		4	▶ command successfully processed [ 94] EN – short: cmd success DE – short: Kdo. erfolgreich
		5	▶ command received toggle [ 94] EN – short: cmd rcvd tgl DE – short: Kommandowechsel
		6	▶ warning [ 94] EN – short: warning DE – short: Warnung
0	1	7	▶ error [ 94] EN – short: error DE – short: Fehler
		8	▶ released for manual movement [ 94] EN – short: manual movement released DE – short: Man. Bwg. freigegeben
		9	▶ software limit reached [ 94] EN – short: softlimit reached DE – short: Softlimit
		10	reserved
		11	▶ no workpiece detected [ 95] EN – short: no wp detected DE – short: Kein Werkstück
		12	▶ workpiece gripped [ 95] EN – short: wp gripped DE – short: Gegriffen

Word	Byte	Bit	Cyclical input data
		13	▶ position reached [📄 95] EN – short: pos reached DE – short: Positioniert
		14	▶ workpiece pre-grip started [📄 95] EN – short: wp pre-grip started DE – short: Nachgreifen
		15	reserved
1	2	16	▶ workpiece lost [📄 95] EN – short: wp lost DE – short: Werkst. verloren
		17	▶ wrong workpiece gripped [📄 95] EN – short: wrong wp gripped DE – short: Falsches Werkst.
		18	reserved
		19	reserved
		20	reserved
		21	reserved
		22	reserved
		23	reserved
1	3	24	reserved
		25	reserved
		26	reserved
		27	reserved
		28	reserved
		29	reserved
		30	reserved
		31	▶ Grip force and position maintenance activated [📄 96] EN – short: GPE activated DE – short: GPE aktiviert

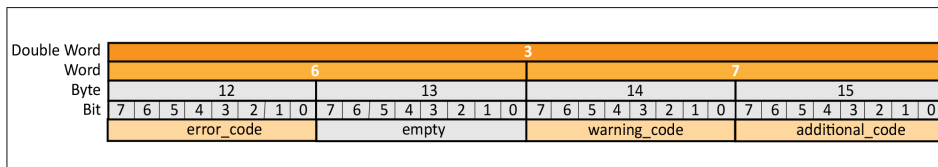
**Actual position**

- In bytes 4 – 7 of the cyclical input data, the current actual position of the module is transmitted, ▶ 4 [□ 49].
- The data format of the parameter is *signed 32 bits* and represents a value in micrometers [µm] (1000 µm ≙ 1 mm).

**Reserve**

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

**Diagnostics double word**



Code sequence diagnostics double word

- The diagnostics double word is transmitted in bytes 12 – 15 of the cyclical input data.
- Confusion between warning and error codes is eliminated since each code is assigned only once.

**NOTE**

If the warning WRN\_NOT\_FEASIBLE, ▶ 6.1 [□ 76] is output via byte 14, the reason for the non-feasibility can be read via byte 15 (additional\_code). The list of causes can be found in the attachment to this document, ▶ 7.4 [□ 96].

**2.1.2 Acyclical data exchange Modbus RTU**

For the data exchange, the firmware implements the Modbus RTU communication protocol according to [www.modbus.com](http://www.modbus.com).

Modbus RTU is organized using registers that can be accessed for reading or writing. A register is always 16 bits wide.

**Function codes**

The registers are accessed via function codes, which in turn access either individual or multiple registers. The following function codes are supported by the firmware:

- "Read Holding Registers" decimal 03 (0x03 hexadecimal) for reading multiple registers
- "Write Multiple Register" decimal 16 (0x10 hexadecimal) for writing multiple registers

## Approach

The module uses a simple approach to make its functionality available via Modbus RTU:

- Register addresses correspond to the known parameter IDs, ▶ 4.2 [49].
- Parameters have different data types (16-bit integer, 32-bit float, arrays of 32-bit integers, etc.) and therefore also different lengths. Parameters can be read or written by accessing several consecutive registers. Since Modbus RTU calculates in units of 16-bit registers, the size specifications must be adjusted accordingly:
  - A 32-bit float is therefore 2 registers in size, a 32-bit integer array with 4 array elements corresponding to 8 registers.
- Access restrictions such as read-only parameters still apply.
- Parameters can only ever be read or written completely. Therefore, even with array parameters, individual array elements cannot be accessed, but all elements must always be read or written together.
- Parameter values are transmitted in little-endian format, which means with the least significant byte first.
- Modbus RTU-specific protocol data, such as register addresses, are transmitted in big-endian format, which means with the most significant byte first.

## 3 Module functions

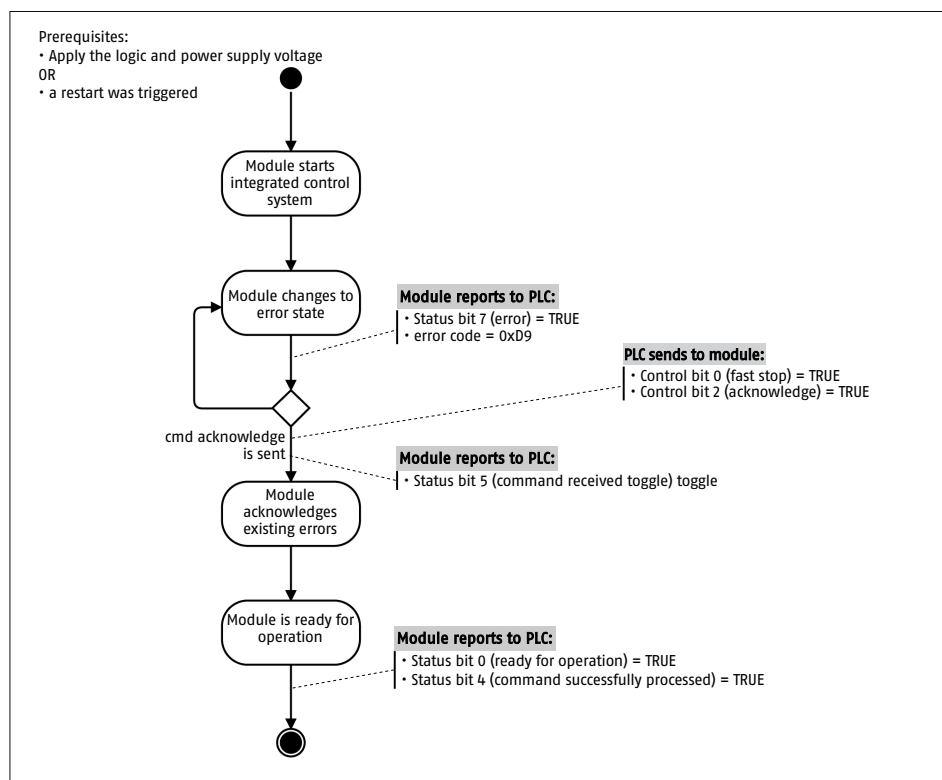
### 3.1 Booting, shutting down and restarting

#### 3.1.1 Booting and establishing operational readiness

##### Short description

When booting, the internal hardware and the connected communication interfaces are checked after the electronics have booted up. The module is in error state after booting. From this state, operational readiness can be established by acknowledgment.

The following example shows the sequence for establishing operational readiness:



*Booting the module and establishing operational readiness*

##### Trigger

Booting can be triggered on the hardware side by applying the logic supply voltage, or triggered on the software side by a restart, ▶ 3.1.3 [26].

If the module is in the error state after booting, establishing operational readiness is triggered by setting the control bit "acknowledge" (bit 2), ▶ 7.2 [90].

---

**NOTE**

- To prevent any unexpected behavior of the module, all control bits equal to 0 should be cyclically transmitted to the module during booting.
  - The module sets all control bits internally 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 toggle" reflects the inverted value of the control bit "fast stop" initially received.  
Background: If the controller transmits a 0 into the control bit "fast stop" during or after booting, this 1 -> 0 transition is interpreted as a request for a "fast stop" and accordingly "command received toggle" is set from the start value 0 to 1.
- 

**Module feedback**

- The receipt of the control command is displayed by an immediate status change of the status bit "command received toggle". This confirmation occurs regardless of whether the control command is subsequently completed successfully or if it can be processed at all.
- If establishing operational readiness was *successful*, this is displayed by setting the status bit "ready for operation". The status bit "error" and the displayed diagnostic code are reset.
- If establishing operational readiness was *not successful*, the module remains in the error state. The status bit "error" and a corresponding diagnostic code are still displayed. In this case, contact SCHUNK Service.

**3.1.2 Shutting down****Short description**

When the module is switched off in a controlled manner, data required for operation is stored permanently. If the module reports back that it is ready for shutdown, the logic supply voltage can be disconnected or a software restart can be triggered.

---

**NOTE**

As soon as the module indicates that it is ready for shutdown, no further functions can be triggered except for the restart (▶ 3.1.3 [□ 26]). Furthermore, no changes to parameter values are accepted.

---

**Trigger**

A controlled shutdown is only permitted from within a defined system status and is triggered by setting the control bit "prepare for shutdown", (Bit 3), ▶ 7.2 [□ 90].

**System status**

To trigger preparation for shutdown, the module must be in one of the following states:

- Position maintenance
- Workpiece holding
- Error state

**NOTE**

- If the shutdown is triggered from the workpiece holding on modules *with* GPE, the module stores this information. After the restart, the corresponding status bit "workpiece gripped" or "wrong workpiece gripped" is displayed again.
- If the shutdown is triggered from the workpiece holding on modules *without* GPE, the module does not store any information about a gripped workpiece.

**NOTE**

- For modules *without* GPE or in case of a hard restart (disconnect voltage/reconnect voltage) the last sent gripping command can be repeated.
- If the workpiece has not been lost, this is displayed by the status bit "workpiece gripped" or "wrong workpiece gripped".
- If the workpiece has been lost, this is displayed by the status bit "no workpiece detected".

**Module feedback**

- The receipt of the control command is displayed by an immediate status change of the status bit "command received toggle". This confirmation occurs regardless of whether the control command is subsequently completed successfully or if it 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. In this case, contact SCHUNK Service.

### 3.1.3 Restart

<b>Short description</b>	When the module is restarted, data required for operation is permanently saved and then booting is initiated, see chapter ▶ 3.1.1 [□ 23].
<b>Trigger</b>	Restarting the module is only permitted from within a defined system status and is triggered by setting the control bit "softreset", (Bit 4), ▶ 7.2 [□ 90].
<b>System status</b>	To trigger the restart, the module must be in one of the following states: <ul style="list-style-type: none"> <li>• Ready to switch off</li> <li>• Position maintenance</li> <li>• Workpiece holding</li> <li>• Error state</li> </ul>

---

#### NOTE

- If the shutdown is triggered from the workpiece holding on modules *with* GPE, the module stores this information. After the restart, the corresponding status bit "workpiece gripped" or "wrong workpiece gripped" is displayed again.
  - If the shutdown is triggered from the workpiece holding on modules *without* GPE, the module does not store any information about a gripped workpiece.
- 

---

#### NOTE

- For modules *without* GPE or in case of a hard restart (disconnect voltage/reconnect voltage) the last sent gripping command can be repeated.
  - If the workpiece has not been lost, this is displayed by the status bit "workpiece gripped" or "wrong workpiece gripped".
  - If the workpiece has been lost, this is displayed by the status bit "no workpiece detected".
-

## 3.2 Movement functions

### 3.2.1 Tip mode

<b>Short description</b>	In jog mode, an outward or inward movement is executed as long as one of the corresponding control bits is set. If the movement reaches the minimum or maximum position, the movement ends automatically.
<hr/>	
<b>NOTE</b>	
The jog mode is exclusively a function for commissioning the module. Do not use this function during automated operation!	
<hr/>	
<b>Trigger</b>	<ul style="list-style-type: none"> <li>• Outward jog mode is triggered by setting the control bit "jog mode positive", (Bit 9) ▶ 7.2 [ 91].</li> <li>• Inward jog mode is triggered by setting the control bit "jog mode negative", (Bit 8) ▶ 7.2 [ 91].</li> </ul>
<b>Movement parameter</b>	<p>The following movement parameters must be transmitted cyclically to the module:</p> <ul style="list-style-type: none"> <li>• <math>\langle \text{min\_vel} \rangle \leq \text{Velocity of movement } [\mu\text{m/s}] \leq \langle \text{max\_grp\_vel} \rangle</math></li> <li>• Application GPE <ul style="list-style-type: none"> <li>– Module with GPE: Use of the GPE is indicated by the state of the control bit "Activate grip force and position maintenance" (Bit 31), ▶ 7.2 [ 93].</li> <li>– Module without GPE: Control bit "Activate grip force and position maintenance" must be equal to 0.</li> </ul> </li> </ul>
<b>Finish</b>	<p>Jog mode is terminated by the following events:</p> <ul style="list-style-type: none"> <li>• Resetting the control bit "jog mode positive" or "jog mode negative"</li> <li>• When reaching the minimum or maximum position</li> </ul>
<b>Module feedback</b>	<ul style="list-style-type: none"> <li>• The receipt of the control command is displayed by an immediate status change of the status bit "command received toggle". This confirmation occurs regardless of whether the control command is subsequently completed successfully or if it can be processed at all.</li> <li>• Termination of jog mode by <i>resetting one of the control bits</i> is indicated by setting the status bits "position reached" and "command successfully processed". The status bit "command received toggle" also changes the state.</li> <li>• Termination of jog mode by <i>reaching the minimum or maximum position</i> is displayed by setting the status bit "position reached" and "command successfully processed". Furthermore the module returns the diagnostic code WRN_POS_LIMIT.</li> </ul>

**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 *
Lower software limit is reached.	ERR_SOFT_LOW
Upper software limit is reached.	ERR_SOFT_HIGH
Drive blocked during movement.	ERR_MOVE_BLOCKED
Sending an impermissible control command.	WRN_NOT_FEASIBLE + additional code
Movement terminated by user.	ERR_FAST_STOP

\* For further information, see chapter ▶ 6 [📄 75].

**3.2.2 Absolute positioning movement**

**Short description**

During absolute positioning, the module moves to the cyclically transferred position value. This position value refers to the parameterized zero point of the module. A practical application example is described in chapter ▶ 7.1 [📄 82], example 1.

**NOTE**

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

**Trigger**

Absolute positioning is triggered by setting the control bit "move to absolute position" (Bit 13), ▶ 7.2 [📄 92].

If the control bit "move to absolute position" is set, new absolute positioning can be triggered by changing the control bit "repeat command toggle", (Bit 6), ▶ 7.2 [📄 91].

- Movement parameter** The following movement parameters must be transmitted cyclically to the module:
- $\langle \text{min\_pos} \rangle \leq \text{absolute position } [\mu\text{m}] \leq \langle \text{max\_pos} \rangle$
  - $\langle \text{min\_vel} \rangle \leq \text{Velocity of movement } [\mu\text{m/s}] \leq \langle \text{max\_vel} \rangle$
  - Application GPE
    - Module with GPE: Use of the GPE is indicated by the state of the control bit "Activate grip force and position maintenance" (Bit 31), ▶ 7.2 [□ 93].
    - Module without GPE: Control bit "Activate grip force and position maintenance" must be equal to 0.

**Finish** Absolute positioning is terminated by the following events:

- Target position reached
  - Setting the control bit "stop"
- Module feedback**
- The receipt of the control command is displayed by an immediate status change of the status bit "command received toggle". This confirmation occurs regardless of whether the control command is subsequently completed successfully or if it can be processed at all.
  - Once the target position has been reached, it is displayed by setting the status bit "position reached" and "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 *
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
Sending an impermissible control command.	WRN_NOT_FEASIBLE + additional code
Movement terminated by user.	ERR_FAST_STOP

\* For further information, see chapter ▶ 6 [□ 75].

### 3.2.3 Relative positioning movement

#### Short description

With relative positioning, the module moves from the current position by the cyclically transferred and signed position value. A practical application example of this is described in chapter ▶ 7.1 [□ 83], Example 2.

---

#### NOTE

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

---

#### Trigger

Relative positioning is triggered by setting the control bit "move to relative position" (Bit 14), ▶ 7.2 [□ 92].

If the control bit "move to relative position" is set, new relative positioning can be triggered by changing the control bit "repeat command toggle" (Bit 6), ▶ 7.2 [□ 91].

#### Movement parameter

The following movement parameters must be transmitted cyclically to the module:

- signed relative position [ $\mu\text{m}$ ]
- $\langle \text{min\_vel} \rangle \leq \text{Velocity of movement } [\mu\text{m/s}] \leq \langle \text{max\_vel} \rangle$
- Application GPE
  - Module with GPE: Use of the GPE is indicated by the state of the control bit "Activate grip force and position maintenance" (Bit 31), ▶ 7.2 [□ 93].
  - Module without GPE: Control bit "Activate grip force and position maintenance" must be equal to 0.

---

#### NOTE

- The transmitted signed relative position must be selected so that the permissible range of movement from  $\langle \text{min\_pos} \rangle$  to  $\langle \text{max\_pos} \rangle$  is not exceeded.
  - If the position is not within the permissible movement range, the module will set the status bit "not feasible" and report back the diagnostics code WRN\_NOT\_FEASIBLE including additional code.
- 

#### Finish

Relative positioning is terminated by the following events:

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

**Module feedback**

- The receipt of the control command is displayed by an immediate status change of the status bit "command received toggle". This confirmation occurs regardless of whether the control command is subsequently completed successfully or if it can be processed at all.
- Once the target position has been reached, it is displayed by setting the status bit "position reached" and "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 *
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
Sending an impermissible control command.	WRN_NOT_FEASIBLE + additional code
Movement terminated by user.	ERR_FAST_STOP

\* For further information, see chapter ▶ 6 [75].

### 3.2.4 Controlled stop

<b>Short description</b>	During controlled stops, the current movement is decelerated as quickly as possible until it comes to a standstill.
<b>Trigger</b>	Controlled stops are triggered by setting the control bit "stop", (Bit 1), ▶ 7.2 [□ 90].
<b>Movement parameter</b>	The following movement parameters must be transmitted cyclically to the module: <ul style="list-style-type: none"> <li>• Application GPE                 <ul style="list-style-type: none"> <li>– Module with GPE: Use of the GPE is indicated by the state of the control bit "Activate grip force and position maintenance" (Bit 31), ▶ 7.2 [□ 93].</li> <li>– Module without GPE: Control bit "Activate grip force and position maintenance" must be equal to 0.</li> </ul> </li> </ul>
<b>Finish</b>	The controlled stop is terminated automatically when a standstill has been reached.
<b>Module feedback</b>	<ul style="list-style-type: none"> <li>• The receipt of the control command is displayed by an immediate status change of the status bit "command received toggle". This confirmation occurs regardless of whether the control command is subsequently completed successfully or if it can be processed at all.</li> <li>• The controlled stop of an active movement is displayed by setting the status bit "position reached" and "command successfully processed".</li> </ul>
<b>Possible diagnostic events</b>	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 an impermissible control command.	WRN_NOT_FEASIBLE + additional code
Movement terminated by user.	ERR_FAST_STOP

\* For further information, see chapter ▶ 6 [□ 75].

### 3.2.5 Terminating a movement

<b>Short description</b>	If the movement is terminated, the current movement is forced to a standstill.
<b>Trigger</b>	Since the control bit "fast stop" is executed in a fail-safe and thus "low-active" manner, the termination of an active movement is triggered by resetting the control bit "fast stop" (1 -> 0), ▶ 7.2 [□ 90].
<b>Module feedback</b>	<ul style="list-style-type: none"> <li>• The receipt of the control command is displayed by an immediate status change of the status bit "command received toggle". This confirmation occurs regardless of whether the control command is subsequently completed successfully or if it can be processed at all.</li> <li>• The movement termination is displayed by setting the status bit "error" in connection with the diagnostic code ERR_FAST_STOP.</li> </ul>

## 3.3 Handling a workpiece

### 3.3.1 Workpiece gripping (simple gripping movement)

<b>Short description</b>	In workpiece gripping, a workpiece is gripped with the specified gripping force value <b>without</b> specifying the workpiece position. A practical application example is described in chapter ▶ 7.1 [□ 84], example 3 – 5.
<b>Trigger</b>	Workpiece gripping is triggered by setting the control bit "grip workpiece", (Bit 12), ▶ 7.2 [□ 92].

---

#### NOTE

As long as a workpiece is held, it is permissible to trigger workpiece gripping with changed movement parameters.

---

<b>Movement parameter for BasicGrip</b>	<p>In order to grip in the BasicGrip mode, the following movement parameters and information must be transmitted cyclically to the module:</p> <ul style="list-style-type: none"> <li>• <math>50 \leq \text{Gripping force [\%]} \leq 100</math></li> <li>• Gripping direction is indicated by the control bit "grip direction" (Bit 7), ▶ 7.2 [□ 91].</li> <li>• Application GPE <ul style="list-style-type: none"> <li>– Module with GPE: Use of the GPE is indicated by the state of the control bit "Activate grip force and position maintenance" (Bit 31), ▶ 7.2 [□ 93].</li> <li>– Module without GPE: Control bit "Activate grip force and position maintenance" must be equal to 0.</li> </ul> </li> </ul>
---	---

**Movement parameter for StrongGrip**

In order to grip in the StrongGrip mode, the following movement parameters and information must be transmitted cyclically to the module:

- EGU 50, 60, 80:  $101 \leq \text{gripping force [\%]} \leq 200$
- EGU 70:  $101 \leq \text{gripping force [\%]} \leq 150$
- Gripping direction is indicated by the control bit "grip direction" (Bit 7), ▶ 7.2 [91].
- Use of the GPE is indicated by the state of the control bit "Activate grip force and position maintenance", ▶ 7.2 [93].

**Finish**

Workpiece gripping is terminated by the following options:

- Workpiece was gripped successfully, the module automatically switches to workpiece holding.
- Workpiece was detected successfully and must be re-gripped, the module automatically switches to workpiece re-gripping.
- Automatic when reaching the minimum or maximum position
- Setting the control bit "stop"

**Module feedback**

- The receipt of the control command is displayed by an immediate status change of the status bit "command received toggle". This confirmation occurs regardless of whether the control command is subsequently completed successfully or if it can be processed at all.
- Successful gripping of a workpiece is displayed by setting the status bit "workpiece gripped" and "command successfully processed".
- The change to workpiece re-gripping is displayed by setting the status bit "workpiece pre-grip started".
- Reaching the minimum or maximum position 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 an impermissible control command.	WRN_NOT_FEASIBLE + additional code
Movement terminated by user.	ERR_FAST_STOP

\* For further information, see chapter ▶ 6 [75].

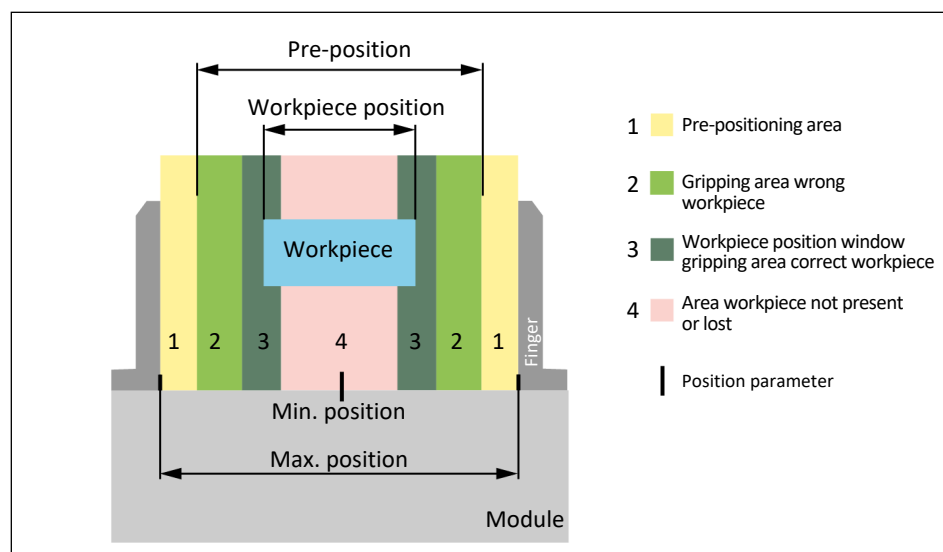
### 3.3.2 Workpiece gripping at expected position (combined gripping movement)

#### Short description

In workpiece gripping at an expected position, a workpiece is gripped at the specified workpiece position with the specified gripping force value using a combined gripping movement. A practical application example is described in chapter ▶ 7.1 [ 87], example 6 – 8.

#### NOTE

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



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

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), then no workpiece has been detected or the workpiece was lost during re-gripping.

---

**NOTE**

- The pre-position and the workpiece position window are calculated from the cyclically transferred workpiece position and the parameterization of the module.
  - Pre-positioning is performed at the maximum positioning speed.
  - 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 [▶ workpiece re-gripping](#) [40] or workpiece holding status.
- 

**Trigger**

The combined gripping movement is triggered by setting the control bit "grip workpiece at expected position" (Bit 16), [▶ 7.2](#) [92].

---

**NOTE**

As long as a workpiece is held, it is permissible to trigger workpiece gripping with changed movement parameters.

---

**Movement parameter for BasicGrip**

In order to grip in the BasicGrip mode, the following movement parameters and information must be transmitted cyclically to the module:

- $50 \leq \text{Gripping force [\%]} \leq 100$
- $\langle \text{min\_pos} \rangle \leq \text{Workpiece position } [\mu\text{m}] \leq \langle \text{max\_pos} \rangle$
- Gripping direction is indicated by the control bit "grip direction" (Bit 7), [▶ 7.2](#) [91].
- Application GPE
  - Module with GPE: Use of the GPE is indicated by the state of the control bit "Activate grip force and position maintenance" (Bit 31), [▶ 7.2](#) [93].
  - Module without GPE: Control bit "Activate grip force and position maintenance" must be equal to 0.

**Movement parameter for StrongGrip**

In order to grip in the StrongGrip mode, the following movement parameters and information must be transmitted cyclically to the module:

- EGU 50, 60, 80:  $101 \leq \text{gripping force [\%]} \leq 200$
- EGU 70:  $101 \leq \text{gripping force [\%]} \leq 150$
- $\langle \text{min\_pos} \rangle \leq \text{Workpiece position } [\mu\text{m}] \leq \langle \text{max\_pos} \rangle$
- Gripping direction is indicated by the control bit "grip direction" (Bit 7), [▶ 7.2](#) [91].
- Use of the GPE is indicated by the state of the control bit "Activate grip force and position maintenance", [▶ 7.2](#) [93].

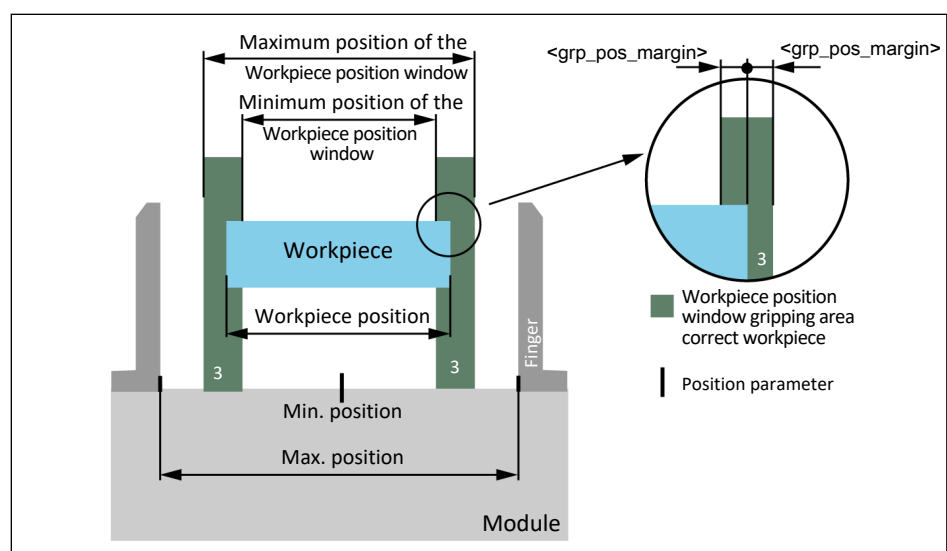
## Parameterization

### 1. Workpiece position window

The parameter `<grp_pos_margin>` ([▶ 4.2 \[ 54\]](#)) 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>`.



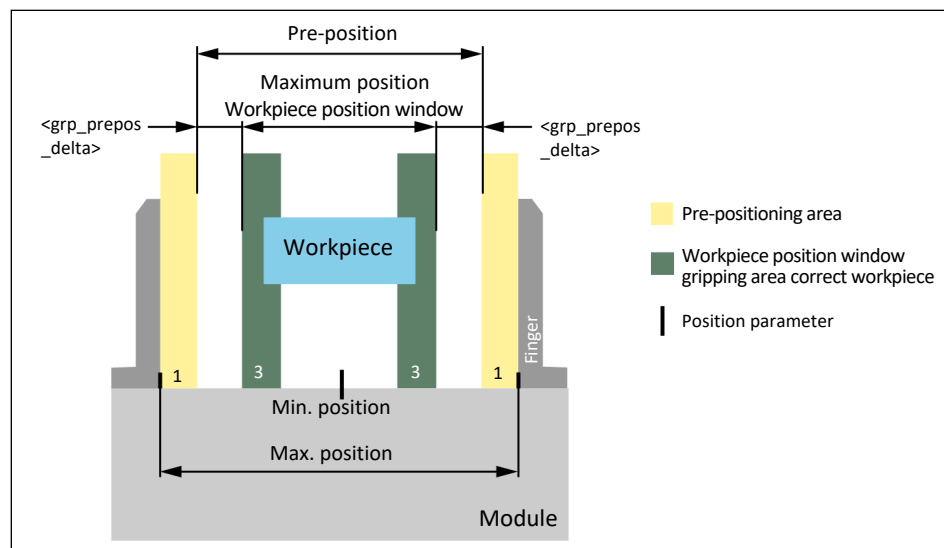
Minimum and maximum position of the workpiece position window

## 2. Pre-position

The parameter `<grp_prepos_delta>` ([▶ 4.2 \[55\]](#)) 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

Workpiece gripping at an expected position is terminated by the following options:

- Expected workpiece was gripped
- Unexpected workpiece was gripped
- Automatic switchover to re-gripping
- Gripping position was exceeded
- Automatic when reaching the minimum or maximum position
- Setting the control bit "stop"

**Module feedback**

- The receipt of the control command is displayed by an immediate status change of the status bit "command received toggle". This confirmation occurs regardless of whether the control command is subsequently completed successfully or if it can be processed at all.
- Gripping of the expected workpiece is displayed by setting the status bit "workpiece gripped" and "command successfully processed".
- Gripping of an unexpected workpiece is displayed by setting the status bit "wrong workpiece gripped" and "command successfully processed".
- The automated change to re-gripping is displayed by setting the status bit "workpiece pre-grip started".
- 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 an impermissible control command.	WRN_NOT_FEASIBLE + additional code
Movement terminated by user.	ERR_FAST_STOP
The drive is blocked during pre-positioning.	ERR_MOVE_BLOCKED

\* For further information, see chapter [▶ 6 \[75\]](#).

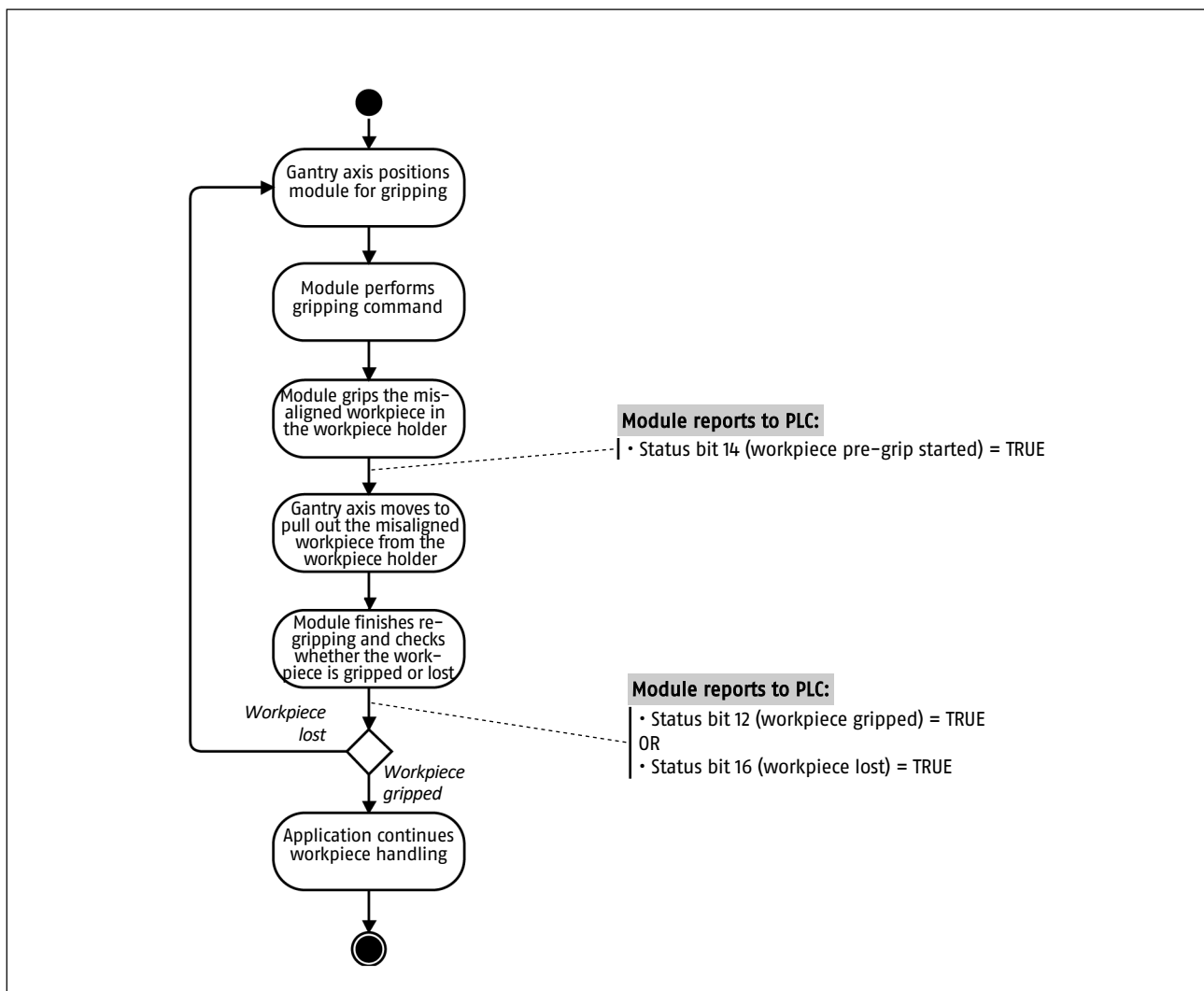
### 3.3.3 Workpiece re-gripping

#### Short description

Workpiece re-gripping is an optional extension for the gripping modes (▶ 1.2.3 [10]) of the module. This allows the module to grip workpieces that align with a delay during the gripping movement. The module detects the first contact with a workpiece and then starts the re-gripping. At the end of the re-gripping process, the module reports back to the user whether the workpiece was gripped or lost. Practical application examples are described in chapter ▶ 7.1 [81].

#### EXAMPLE

#### Workpiece re-gripping in a linear gantry application



Workpiece re-gripping in a linear gantry application

---

**NOTE**

In the application example shown above, it can be seen that starting the re-gripping can serve as a trigger for an action of another application component (PLC reads status bit 14 = TRUE). The feedback from the module as to whether the workpiece has been gripped or lost can in turn be used as a trigger for distinguishing further cases in the application.

---

**Trigger**

The re-gripping behavior is determined by the parameter <grp\_prehold\_time>. If a re-gripping time is set in the parameter, re-gripping with the set time takes place for all executed gripping commands.

**Parameterization**

The parameter <grp\_prehold\_time> (► 4.2 [□ 52]) can be used to parameterize the time span of the re-gripping. The maximum time span for re-gripping is 60,000 ms (1 minute).

---

**NOTE**

If a time of 0 ms is stored in this parameter (factory setting), re-gripping is *not* carried out when carrying out a gripping movement.

---

**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 previous workpiece gripping:**

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

**Feedback after previous workpiece gripping at an expected position:**

- Gripping of the expected workpiece is displayed by setting the status bit "workpiece gripped" and "command successfully processed".
- 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 an impermissible control command.	WRN_NOT_FEASIBLE + additional code
Movement terminated by user.	ERR_FAST_STOP

\* For further information, see chapter ▶ 6 [ 75].

**3.3.4 Workpiece loss detection**

**Short description**

The module can detect loss of the workpiece. Due to a workpiece loss, the module performs a movement. As soon as this movement results in a position change greater than a parameterizable distance, the workpiece is recognized as lost.

**NOTE**

For modules with GPE: Due to fixed compliance of the drive train and adjustability of the parameter <wp\_lost\_dst>, it is possible for a workpiece loss to not be recognized when GPE is activated. In case of uncertainty, the last gripping command can be executed again. In this case, the workpiece loss is displayed by setting the status bit "no workpiece detected".

**Trigger**

The workpiece loss detection does not have to be triggered. It is automatically activated as soon as a workpiece is held.

**Parameterization**

The parameter <wp\_lost\_dst> (▶ 4.2 [ 53]) can be used to parameterize the distance that the gripper fingers are allowed to travel after the workpiece is lost before a workpiece loss is detected.

- Module feedback**
- A workpiece loss is displayed by setting the status bit "workpiece lost".
  - A set status bit "workpiece pre-grip started" is reset.
  - A set status bit "workpiece gripped" is reset.
  - A set status bit "wrong workpiece gripped" is reset.

### 3.3.5 Workpiece release

**Short description** During workpiece release, the module executes a relative positioning movement. Starting from the current position, a defined distance of the parameter <wp\_release\_delta> is moved in the opposite direction to the gripping direction of the last gripping movement.

---

#### NOTE

- Since all necessary movement parameters are calculated internally during workpiece release, a maximum of two control bits (bit 11, optional bit 31) must be changed.
  - Workpieces can also be released by triggering an absolute or relative positioning movement.
- 

**Trigger** Releasing workpieces is only permitted from workpiece holding and is triggered by setting the control bit "release workpiece" (Bit 11), ▶ 7.2 [ 91].

**Movement parameter** The following movement parameters must be transmitted cyclically to the module:

- Use of the GPE is indicated by the state of the control bit "Activate grip force and position maintenance", ▶ 7.2 [ 93].

**Parameterization** The parameter <wp\_release\_delta> (▶ 4.2 [ 53]) can be used to parameterize the distance which the module moves relatively during release.

**Finish** Workpiece release is terminated by the following options:

- Calculated release position was reached
- Setting the control bit "stop"

- Module feedback**
- The receipt of the control command is displayed by an immediate status change of the status bit "command received toggle". This confirmation occurs regardless of whether the control command is subsequently completed successfully or if it can be processed at all.
  - The release of workpieces is displayed by setting the status bit "position reached" and "command successfully processed".
  - A set status bit "workpiece pre-grip started" is reset.
  - A set status bit "workpiece gripped" is reset.
  - A set status bit "wrong workpiece gripped" is 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 an impermissible control command.	WRN_NOT_FEASIBLE + additional code
Movement terminated by user.	ERR_FAST_STOP

\* For further information, see chapter ▶ 6 [□ 75].

**3.3.6 Remove workpiece manually**

**Short description**

If the module is in the **error state**, the GPE of the module can be deactivated. The user can manually remove a gripped 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 unexpected movements, it is only possible 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 for manual movement", (Bit 5), ▶ 7.2 [□ 90].

If the module is **not** in the error state, proceed as follows:

- Resetting the control bit "fast stop" (bit 0)
- Setting the control bit "fast stop" (bit 0)
- Setting the control bit "release for manual movement" (bit 5)

**Movement parameter**

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

**Finish**

The manual gripping of workpieces is terminated by the following events:

- Resetting the control bit "fast stop" to 0
- Automatically after 30 minutes

**Module feedback**

- The receipt of the control command is displayed by an immediate status change of the status bit "command received toggle". This confirmation occurs regardless of whether the control command is subsequently completed successfully or if it can be processed at all.
- The release for manual workpiece removal is displayed by setting the status bit "released for manual movement".

## 3.4 Additional functions

### 3.4.1 Zero point offset

When using application-specific gripper fingers, the zero point can be "shifted" so that the displayed position values match the geometry of the gripper fingers. Moving the zero point automatically changes the values of the parameters <actual\_pos>, <min\_pos> and <max\_pos>.

---

#### NOTE

In the delivery state, the zero point corresponds to the minimum position of the base jaws, see ▶ 1.2.1 [ 8].

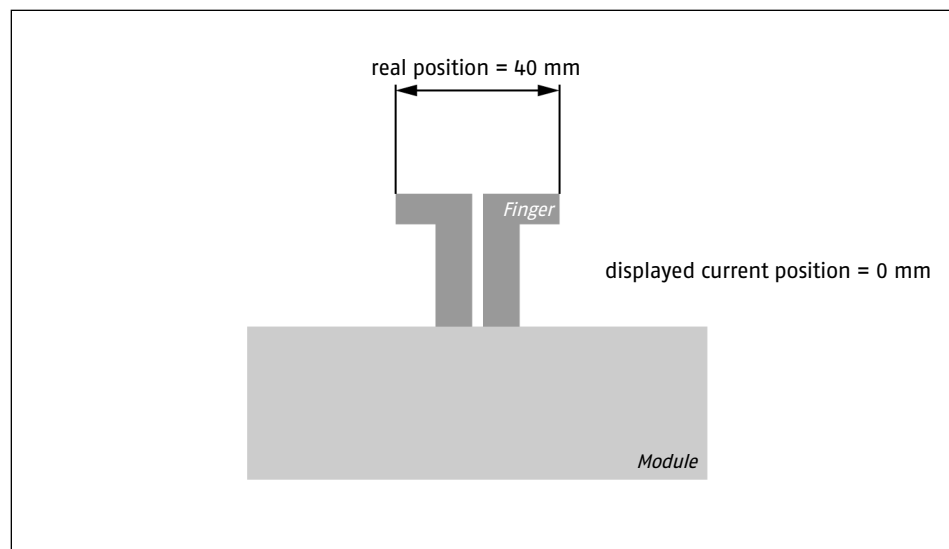
---

#### Parameterization

With the parameter <zero\_pos\_ofs> (▶ 4.2 [ 56]) the distance by which the zero point is shifted can be parameterized with a sign.

#### Example: Zero point offset EGU 50

- Application:
  - Values of actual, minimum and maximum position **before** displacement.
    - > actual\_pos = 0 mm
    - > min\_pos = 0 mm
    - > max\_pos = 102 mm
- The module is to be used for I.D. gripping with the gripper fingers shown.



#### Zero point offset

- Shifting of the zero point:
  - Write the value +40 mm in the parameter <zero\_pos\_ofs>.

- Values of actual, minimum and maximum position **after** displacement.
  - > actual\_pos = 40 mm
  - > min\_pos = 40 mm
  - > max\_pos = 142 mm

### 3.4.2 Handshake

**Short description**

If the module detects the input of a control command, it reports the input back to the control system.

**Module feedback**

The receipt of the control command is displayed by an immediate status change of the status bit "command received toggle".

### 3.4.3 Brake test

<b>Short description</b>	During the brake test, the module checks the holding force of the brake by applying a defined moment alternately in both directions against the applied brake.
<b>Trigger</b>	<ul style="list-style-type: none"> <li>• The brake test is triggered by setting the control bit "brake test" (bit 30), ▶ 7.2 [93].</li> <li>• The module must be at a standstill and not holding any workpiece.</li> </ul>
<b>Module feedback</b>	<ul style="list-style-type: none"> <li>• The receipt of the control command is displayed by an immediate status change of the status bit "command received toggle". This confirmation occurs regardless of whether the control command is subsequently completed successfully or if it can be processed at all.</li> <li>• A successfully performed brake test is displayed by setting the status bit "command successfully processed".</li> <li>• An unsuccessful brake test is indicated by setting the status bit "error" in conjunction with the diagnostic code ERR_BT_FAILED, ▶ 6.2 [78].</li> </ul>

### 3.4.4 LifeSign

<b>Short description</b>	The acknowledge feature can be used to check communication between the control system and module. As soon as the command is triggered, the module reports the arrival of the control command back to the control system. If no feedback is received, it is assumed that there is a fault in the communication.
<b>Trigger</b>	Acknowledgment is triggered by setting the control bit "acknowledge" (Bit 2), ▶ 7.2 [90].
<b>Module feedback</b>	The receipt of the control command is displayed by an immediate status change of the status bit "command received toggle".

### 3.4.5 Repeat control command with optimized time

**Short description**

This function makes it possible to send consecutive identical control commands to the module in a time-optimized manner.

---

**NOTE**

By default, control commands are sent to the module by 0 -> 1 edges of individual control bits. If the same function is to be executed again, the corresponding control bit must first be reset and then set again.

**Trigger**

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

### 3.4.6 Factory settings

**Short description**

The module can be reset from the error state to the factory settings using the software. This restores the default parameterization to that when the module was delivered. In addition, the diagnostic memory is deleted.

---

**NOTE**

The parameter <enable\_softreset> is not reset when the factory settings are restored.

**Trigger**

Resetting to factory settings is triggered by writing the value HEX 0x8C (= 140 decimal) to the parameter HEX 0x0100 (= 256 decimal). The data type of the HEX 0x100 parameter is UINT16.

**Module feedback**

Successful resetting to factory settings is displayed by setting the status bit "'ready for shutdown'".

---

**NOTE**

It is mandatory to wait for this bit to be set before restarting or switching off the module.

**Possible diagnostic events**

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

Diagnostic event	Diagnostic code *
Resetting to factory settings is not possible	WRN_NOT_FEASIBLE + additional code

**NOTICE**

**Material damage due to faulty usage!**

After resetting to factory settings, application-specific parameters must be adjusted again. 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
INT32	MIN_INT32	-2147483648
	MAX_INT32	2147483647
FLOAT	MIN_FLOAT	-3.402823E+38
	MAX_FLOAT	3.402823E+38
CHAR	MIN_CHAR	0
	MAX_CHAR	255
ENUM	MIN_ENUM	0
	MAX_ENUM	255

### 4.2 Parameter list

In the following, all system-relevant parameters are listed according to the diagram "HEX-Code/DEC-Code <Parametername>"

#### NOTE

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

Some of the parameters listed here as "read only" can be changed in principle, but the user does not have the right to change these parameters.

All parameters that do not appear in this list are internal or reserved parameters.

#### Parameter configuration

All system parameters whereby the user has write permissions can be configured via acyclical data exchange.

**HEX 0x0040**  
**DEC 64**

**<plc\_sync\_input>**

Short description: This parameter illustrates the cyclical input data (PLC frame input data)

Parameter name: PLC synch. data frame input

Access rights: Read

Data type: 1x UINT32, 1x INT32, 2x UINT32 (together 16 bytes)

Number of registers: 8

**HEX 0x0048**  
**DEC 72**

**<plc\_sync\_output>**

Short description: This parameter illustrates the cyclical output data (PLC frame output data)

Parameter name: PLC synch. data frame output

Access rights: Read and write

Data type: 1x UINT32, 2x INT32, 1x UINT32 (together 16 bytes)

Number of registers: 8

**HEX 0x0118**  
**DEC 280**

**<err\_code>**

Short description: The existing error code can be read out with this parameter.

Parameter name: Error Code

Access rights: Read

Data type: ENUM

Enumeration: see chapter ▶ 6.2 [77]

Number of registers: 1

**HEX 0x0120**  
**DEC 288**

**<wrn\_code>**

Short description: The existing warning code can be read out with this parameter.

Parameter name: Warning Code

Access rights: Read

Data type: ENUM

Enumeration: see chapter ▶ 6.1 [75]

Number of registers: 1

**HEX 0x0128**  
**DEC 296**

**<sys\_msg\_req>**

Short description: With this parameter, an entry in the diagnostic memory can be selected for reading out via <sys\_msg\_buffer> by writing an index

Parameter name: Request system message

Access rights: Read and write

Data type: UINT16

Number of registers: 1

---

**NOTE**

The 32 (index 0 - 31) most recent diagnostic events are stored in the diagnostic memory.

---

**HEX 0x0130**  
**DEC 304**

**<sys\_msg\_buffer>**

Short description: The requested diagnostic memory entry can be read out via this parameter.

Parameter name: System message buffer

Access rights: Read

Data type: CHAR[214]

Format: ASCII-String

Number of registers: 107

**HEX 0x0230**  
**DEC 560**

**<actual\_pos>**

Short description: This parameter can be used to read out the current actual position.

Parameter name: Position

Access rights: Read

Data type: FLOAT

Unit: Millimeter [mm]

Number of registers: 2

**HEX 0x0238**  
**DEC 568**

**<actual\_vel>**

Short description: This parameter can be used to read out the current actual speed.

Parameter name: Velocity

Access rights: Read

Data type: FLOAT

Unit: Millimeter per second [mm/s]

Number of registers: 2

**HEX 0x0380**  
**DEC 896**

**<grp\_prehold\_time>**

Short description: This parameter can be used to read and write the time span for the re-gripping.

Parameter name: Grip prehold time

Access rights: Read and write

Data type: UINT16

Unit: Millisecond [ms]

Min; Max; Default: 0; 5000; 0

Number of registers: 1

**HEX 0x03A8**  
**DEC 936**

**<dead\_load\_kg>**

Short description: The mass of the module can be read out and written with this parameter.

Parameter name: Net mass of the gripper

Access rights: Read

Data type: FLOAT

Unit: Kilogram [kg]

Number of registers: 2

**HEX 0x03B0**  
**DEC 944**

**<tool\_cent\_point>**

Short description: The tool center point (TCP) of the module can be read out and written with this parameter.

Parameter name: Tool center point 6D-frame

Access rights: Read

Data type: 6x FLOAT (24 byte)

Unit: x [mm], y [mm], z [mm], a [°], b [°], c [°]

Number of registers: 12

**HEX 0x03B8**  
**DEC 952**

**<cent\_of\_mass>**

Short description: The center of mass and the mass moments of inertia of the module can be read out and written with this parameter.

Parameter name: Center of mass- 6D-frame

Access rights: Read

Data type: 6x FLOAT (24 byte)

Unit: x [mm], y [mm], z [mm], a [kg\*m<sup>2</sup>], b [kg\*m<sup>2</sup>], c [kg\*m<sup>2</sup>]

Number of registers: 12

**HEX 0x0500**  
**DEC 1280**

**<module\_type>**

Short description: The module type can be read out with this parameter.

Parameter name: Modul type

Access rights: Read

Data type: ENUM

Enumeration: The enumeration value matching the module is read out.

Number of registers: 1

**HEX 0x0528**  
**DEC 1320**

**<wp\_lost\_dst>**

Short description: This parameter can be used to set the traverse path from which a workpiece loss is detected.

Parameter name: Max. distance after workpiece lost

Access rights: Read and write

Data type: FLOAT

Unit: Millimeter [mm]

Min; Max; Default 0.1; 50; 1

Number of registers: 2

**HEX 0x0540**  
**DEC 1344**

**<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.

Parameter name: Workpiece release delta positione

Access rights: Read and write

Data type: FLOAT

Unit: Millimeter [mm]

Min; Max; Default 1; 50; 2

Number of registers: 2

**HEX 0x0580**  
**DEC 1408**

**<grp\_pos\_margin>**

Short description: With this parameter the tolerance value of the workpiece position window can be read and written.

Parameter name: Margin for workpiece detection

Access rights: Read and write

Data type: FLOAT

Unit: Millimeter [mm]

Min; Max; Default 1; 10; 2

Number of registers: 2

**HEX 0x0588**  
**DEC 1416**

**<max\_phys\_stroke>**

Short description: With this parameter the maximum physical stroke of the module without fingers can be read.

Parameter name: Max. physical stroke

Access rights: Read

Data type: FLOAT

Unit: Millimeter [mm]

Number of registers: 2

---

**NOTE**

Depending on the size, the maximum strokes are as follows:

- EGU50 = 102 mm / EGU50-SD = 82 mm
  - EGU60 = 120 mm / EGU60-SD = 100 mm
  - EGU70 = 140 mm / EGU70-SD = 120 mm
  - EGU80 = 160 mm / EGU80-SD = 140 mm
-

HEX 0x05A8  
DEC 1448

### <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.

Parameter name: Gripping pre-position delta

Access rights: Read and write

Data type: FLOAT

Unit: Millimeter [mm]

Min; Max; Default 1; 50; 5

Number of registers: 2

HEX 0x0600  
DEC 1536

### <min\_pos>

Short description: The smallest position value that can be approached by the module can be read out and written with this parameter.

Parameter name: Min. absolut position

Access rights: Read and write

Data type: FLOAT

Unit: Millimeter [mm]

Number of registers: 2

---

### NOTE

- Values within the following limits can be written to this parameter:  
 $\langle \text{zero\_pos\_ofs} \rangle \leq \text{Value} < \langle \text{max\_phys\_stroke} \rangle + \langle \text{zero\_pos\_ofs} \rangle$
  - Furthermore, the value must be smaller than the value of the parameter <max\_pos>.
-

**HEX 0x0608**  
**DEC 1544**

**<max\_pos>**

Short description: The largest position value that can be approached by the module can be read out and written with this parameter.

Parameter name: Max. absolut position

Access rights: Read and write

Data type: FLOAT

Unit: Millimeter [mm]

Number of registers: 2

---

**NOTE**

- Values within the following limits can be written to this parameter:  
 $\langle \text{zero\_pos\_ofs} \rangle < Value \leq \langle \text{max\_phys\_stroke} \rangle + \langle \text{zero\_pos\_ofs} \rangle$
  - Furthermore, the value must be larger than the value of the parameter  $\langle \text{min\_pos} \rangle$ .
- 

**HEX 0x0610**  
**DEC 1552**

**<zero\_pos\_ofs>**

Short description: The zero point can be adapted to the application with this parameter.

Parameter name: Zero position Offset

Access rights: Read and write

Data type: FLOAT

Unit: Millimeter [mm]

Min; Max; Default -10000; 10000; 0

Number of registers: 2

**HEX 0x0628**  
**DEC 1576**

### <min\_vel>

Short description: The minimum movement/gripping velocity with which the module can be moved can be read out with this parameter.

Parameter name: Min. velocity  
Access rights: Read  
Data type: FLOAT  
Unit: Millimeter per second [mm/s]  
Number of registers: 2

---

### NOTE

Regardless of the size, the minimum movement velocity is 10 mm/s.

---

**HEX 0x0630**  
**DEC 1584**

### <max\_vel>

Short description: The maximum positioning speed with which the module can be moved can be read out with this parameter.

Parameter name: Max. velocity  
Access rights: Read  
Data type: FLOAT  
Unit: Millimeter per second [mm/s]  
Number of registers: 2

---

### NOTE

Depending on the size, the maximum movement velocities are as follows:

- EGU50 = 110 mm/s
  - EGU60 = 110 mm/s
  - EGU70 = 70 mm/s
  - EGU80 = 70 mm/s
-

**HEX 0x0650**  
**DEC 1616**

**<max\_grp\_vel>**

Short description: The maximum gripping velocity with which the module can be moved can be read out with this parameter.

Parameter name: Max. grip velocity  
 Access rights: Read  
 Data type: FLOAT  
 Unit: Millimeter per second [mm/s]  
 Number of registers: 2

---

**NOTE**

- Regardless of the size, the maximum gripping velocity is 25 mm/s.
  - For more information on gripping speed, see the chapter [▶ Gripping modes \[10\]](#).
- 

**HEX 0x0658**  
**DEC 1624**

**<min\_grp\_force>**

Short description: The minimum gripping force can be read out with this parameter.

Parameter name: Min. grip force  
 Access rights: Read  
 Data type: FLOAT  
 Unit: Newton [N]  
 Number of registers: 2

**HEX 0x0660**  
**DEC 1632**

**<max\_grp\_force>**

Short description: The maximum gripping force can be read out with this parameter.

Parameter name: Max. grip force  
 Access rights: Read  
 Data type: FLOAT  
 Unit: Newton [N]  
 Number of registers: 2

**HEX 0x06A8**  
**DEC 1704**

**<max\_allow\_force>**

Short description: The maximum gripping force can be read out for the StrongGrip mode with this parameter.

Parameter name: Max.allowed grip force StrongGrip

Access rights: Read

Data type: FLOAT

Unit: Newton [N]

Number of registers: 2

**HEX 0x0800**  
**DEC 2048**

**<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.

Parameter name: Min. error motor voltage

Access rights: Read

Data type: FLOAT

Unit: Volt [V]

Number of registers: 2

**HEX 0x0808**  
**DEC 2056**

**<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.

Parameter name: Max. error motor voltage

Access rights: Read

Data type: FLOAT

Unit: Volt [V]

Number of registers: 2

**HEX 0x0810**  
**DEC 2064**

**<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.

Parameter name: Min. error logic voltage

Access rights: Read

Data type: FLOAT

Unit: Volt [V]

Number of registers: 2

**HEX 0x0818**  
**DEC 2072**

**<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.

Parameter name: Max. error logic voltage

Access rights: Read

Data type: FLOAT

Unit: Volt [V]

Number of registers: 2

**HEX 0x0820**  
**DEC 2080**

**<min\_err\_lgc\_temp>**

Short description: With this parameter the lower exact error limit of the temperature of the logic part can be read out.

Parameter name: Min. error logic voltage

Access rights: Read

Data type: FLOAT

Unit: Degrees Celsius [°C]

Number of registers: 2

**HEX 0x0828**  
**DEC 2088**

**<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.

Parameter name: Max. error logic temperature

Access rights: Read

Data type: FLOAT

Unit: Degrees Celsius [°C]

Number of registers: 2

**HEX 0x0840**  
**DEC 2112**

**<meas\_lgc\_temp>**

Short description: With this parameter the current measured temperature of the logic part can be read out.

Parameter name: Measured logic temperature

Access rights: Read

Data type: FLOAT

Unit: Degrees Celsius [°C]

Number of registers: 2

<b>HEX 0x0870</b> <b>DEC 2160</b>	<b>&lt;meas_lgc_volt&gt;</b> Short description: With this parameter the current measured supply voltage of the logic part can be read out. Parameter name: Measured logic voltage Access rights: Read Data type: FLOAT Unit: Volt [V] Number of registers: 2
<b>HEX 0x0878</b> <b>DEC 2168</b>	<b>&lt;meas_mot_volt&gt;</b> Short description: With this parameter the current measured supply voltage of the motor can be read out. Parameter name: Measured motor voltage Access rights: Read Data type: FLOAT Unit: Volt [V] Number of registers: 2
<b>HEX 0x0880</b> <b>DEC 2176</b>	<b>&lt;min_wrn_mot_volt&gt;</b> Short description: With this parameter the lower exact warning limit of the supply voltage of the motor can be read out and written. Parameter name: Min. warning motor voltage Access rights: Read and write Data type: FLOAT Unit: Volt [V] Number of registers: 2
<b>HEX 0x0888</b> <b>DEC 2184</b>	<b>&lt;max_wrn_mot_volt&gt;</b> Short description: With this parameter the upper exact warning limit of the supply voltage of the motor can be read out and written. Parameter name: Max. warning motor voltage Access rights: Read and write Data type: FLOAT Unit: Volt [V] Number of registers: 2

**HEX 0x0890**  
**DEC 2192**

**<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.

Parameter name: Min. warning logic voltage

Access rights: Read and write

Data type: FLOAT

Unit: Volt [V]

Number of registers: 2

**HEX 0x0898**  
**DEC 2200**

**<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.

Parameter name: Max. warning logic voltage

Access rights: Read and write

Data type: FLOAT

Unit: Volt [V]

Number of registers: 2

**HEX 0x08A0**  
**DEC 2208**

**<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.

Parameter name: Min. warning logic logic temperaturure

Access rights: Read

Data type: FLOAT

Unit: Degrees Celsius [°C]

Number of registers: 2

**HEX 0x08A8**  
**DEC 2216**

**<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.

Parameter name: Max. warning logic logic temperaturure

Access rights: Read

Data type: FLOAT

Unit: Degrees Celsius [°C]

Number of registers: 2

<b>HEX 0x1000</b> <b>DEC 4096</b>	<b>&lt;serial_no_txt&gt;</b>	<p>Short description: The serial number of the module can be read out with this parameter.</p> <p>Parameter name: Device serial number</p> <p>Access rights: Read</p> <p>Data type: CHAR[16]</p> <p>Format: ASCII-String</p> <p>Number of registers: 8</p>
<b>HEX 0x1008</b> <b>DEC 4104</b>	<b>&lt;order_no_txt&gt;</b>	<p>Short description: The order number of the module can be read out with this parameter.</p> <p>Parameter name: Order number</p> <p>Access rights: Read</p> <p>Data type: CHAR[16]</p> <p>Format: ASCII-String</p> <p>Number of registers: 8</p>
<b>HEX 0x1020</b> <b>DEC 4128</b>	<b>&lt;serial_no_num&gt;</b>	<p>Short description: The serial number of the module can be read out numerically with this parameter.</p> <p>Parameter name: Device serial number encoded</p> <p>Access rights: Read</p> <p>Data type: UINT32</p> <p>Number of registers: 2</p>

**HEX 0x1100**  
**DEC 4352**

**<fw\_app\_date>**

Short description: The creation date of the firmware version can be read out with this parameter.  
Parameter name: Applikation firmware creation date  
Access rights: Read  
Data type: CHAR[12]  
Format: ASCII-String  
Number of registers: 6

**HEX 0x1108**  
**DEC 4360**

**<fw\_app\_time>**

Short description: The creation time of the firmware version can be read out with this parameter.  
Parameter name: Application firmware creation time  
Access rights: Read  
Data type: CHAR[9]  
Format: ASCII-String  
Number of registers: 5

**HEX 0x1110**  
**DEC 4368**

**<fw\_app\_ver\_num>**

Short description: The version of the software can be read out as a number with this parameter.  
Parameter name: Application firmware version short  
Access rights: Read  
Data type: UINT16  
Number of registers: 1

**HEX 0x1118**  
**DEC 4376**

**<fw\_app\_ver\_txt>**

Short description: The version of the software can be read out as a text with this parameter.  
Parameter name: Application firmware version  
Access rights: Read  
Data type: CHAR[22]  
Format: ASCII-String  
Number of registers: 11

**HEX 0x1138**  
**DEC 4408**

**<mac\_addr>**

Short description: The MAC address of the module can be read out with this parameter.

Parameter name: MAC adress

Access rights: Read

Data type: UINT8

Format: MAC

Number of registers: 3

**HEX 0x11A0**  
**DEC 4512**

**<baudrate>**

Short description: The RS-485 communication on which the Modbus RTU communication is based can be read out and written with this parameter

Parameter name: Modbus-RTU baudrate

Access rights: Read and write

Data type: UINT32

Permissible values: 19200, 115200, 230400, 460800, 921600

Special features: A change is only applied after a restart.

Number of registers: 2

**HEX 0x11A8**  
**DEC 4520**

**<slave\_id>**

Short description: The Modbus slave ID of the module can be read out and written with this parameter.

Parameter name: Modbus-RTU slave ID

Access rights: Read and write

Data type: UINT8

Special features: Specifically for Modbus RTU, only values from 1 to 247 are allowed.

Number of registers: 1

**HEX 0x1330**  
**DEC 4912**

**<enable\_softreset>**

Short description: The "Restart" function can be enabled with this parameter.

Parameter name: Enable software reboot

Access rights: Read and write

Data type: BOOL

Values: 0 = function switched off  
1 = function switched on

Number of registers: 1

**HEX 0x1400**  
**DEC 5120**

**<system\_uptime>**

Short description: The operating time that has elapsed since the last (re)start of the module can be read out with this parameter.

Parameter name: System uptime

Access rights: Read

Data type: UINT32

Unit: Seconds [s]

Number of registers: 2

## 5 Start-up

### 5.1 Safety

Commissioning of the module may only be carried out by qualified personnel with programming and interface knowledge!



#### ⚠ WARNING

##### Risk of injury from crushing and impacts!

Serious injury could occur during movement of the base jaw, due to breakage or loosening of the gripper fingers or if the workpiece is lost.

- Wear suitable protective equipment.
- Do not reach into the open mechanism or the movement area of the product.



#### ⚠ CAUTION

##### Risk of injury due to electromagnetic interference!

Electromagnetic interference can cause malfunctions and lead to unexpected movements.

- Use electrical components, e.g. sensors, controllers, etc. according to EN 61000-5-7.

### 5.2 System integration

The communication protocol "SCHUNK Flexible Protocol" is available for operation within the plant.

For further information on communication, module functions and parameters, see the corresponding sections in this manual.

#### Overview

- The module is mounted and electrically connected. For more information, see Assembly and Operating Manual, ▶ 1.1 [7].
1. Activate logic and power supply.
    - ⇒ LED LOG and PWR light up green.
  2. Connect the cables for communication.
    - ⇒ Communication is reported back by LED status indicator.
  3. Configure controller and module, ▶ 5.3 [68].
  4. Determine the program sequence.

## 5.3 Commissioning Modbus RTU

### 5.3.1 General communication settings

For commissioning a module as a Modbus RTU slave, the basic communication settings (operating mode, transmission mode, baud rate and slave ID) must be made known to the Modbus RTU master.

#### Operating mode

The module uses the operating mode "half duplex" via RS485 (EIA/TIA-485) in two-wire operation. For the communication, 3 lines are used: ground and the two differential signal lines A and B.

#### Transmission mode

The transmission mode of the module is set to 8 data bits, even parity and 1 stop bit (8E1). That means that 11 bits (1 start bit, 8 data bits, 1 even parity bit and 1 stop bit) are transmitted per data frame. Flow control is "none", RTS on and off delay is 0, no delay.

---

#### NOTE

In the Modbus communication network, make sure that the RTU transmission mode is the same for all bus nodes.

---

#### Baud rate

In the delivery state, the baud rate of the module is set to 115,200 bit/s. The baud rate can be changed by adjusting the parameter <baudrate> (► 4.2 [📄 65]) or via Modbus RTU register accesses. Possible baud rates are 19,200, 115,200 (default), 230,400, 460,800, 921,600 bit/s.

---

#### NOTE

In the Modbus communication network, make sure that the baud rate is the same for all bus nodes.

---

#### Slave ID

In the delivery state, the slave ID of the module is set to decimal 12 (= 0x0C hexadecimal). The slave ID can be set by changing the parameter <slave\_id> (► 4.2 [📄 65]) or via Modbus RTU register accesses. Possible slave IDs are decimal 1 - 247 (0x01 - 0xF7 hexadecimal).

---

#### NOTE

In the Modbus communication network, make sure that the slave IDs are different for all bus nodes.

---

The following table shows a summary of the general communication settings. The designations may differ depending on the controller manufacturer. For this reason, common alternative designations have also been listed.

Modbus designation	Default value	Description	Alternative designations
Transmission mode	8E1	Number of data bits, parity, number of stop bits	Transmission mode, communication parameters / CommParam
Baud rate	115,200	Transmission rate in bit/s	
SlaveID	12	Unique identification of a Modbus RTU slave	Station address / UnitID / Modbus address / mbAddr

**NOTE**

In the case of control manufacturers, the alternative designations overlap in some cases. Thus, Siemens designates the *SlaveID* with *mbAddr*, while *MbAddr* at Beckhoff designates the *register address*.

**5.3.2 Data-specific communication settings**

For the acyclical data exchange with the module, the Modbus function codes "Read Holding Registers" for reading and "Write Multiple Registers" for writing parameters are used, ▶ 2.1.2 [21]. Furthermore, the following must be specified when exchanging data:

- The start address of the parameter to be transmitted must be specified via the register address.
- The number (quantity) of 16-bit registers to be transmitted *and/or* a byte count with the number of transmitted 8-bit bytes must be specified.

The data transmission is secured with a 16-bit checksum (CRC – Cyclic Redundancy Check), see "Appendix B CRC Generation" at [www.modbus.org/docs/](http://www.modbus.org/docs/).

The following table shows a summary of the general data-specific communication settings. The designations may differ depending on the controller manufacturer. For this reason, common alternative designations have also been listed.

Modbus designation	Description	Alternative designation
Read holding registers	Function code 3: read one or more 16-bit registers	ReadInputRegs
Write multiple registers	Function code 16: write one or more 16-bit registers	WriteRegs
Register address	Address of the data to be read or written	Modbus data address / MAddr (Beckhoff), data start address / dataAddr (Siemens)
Quantity	Number of 16-bit registers to be transmitted	Quantity (Beckhoff), Data length / dataLen (Siemens)
Byte count	Number of transmitted 8-bit bytes	cbLength (Beckhoff)
CRC	16-bit checksum	CRC

**NOTE**

In the case of control manufacturers, the alternative designations overlap in some cases. Thus, Siemens designates the *SlaveID* with *mbAddr*, while *MAddr* at Beckhoff designates the *register address*.

**5.3.3 Data exchange**

The function codes "Read Holding Registers" and "Write Multiple Registers" can be used to access any parameter. Read and write restrictions are retained, e.g. read-only parameters can *not* be accessed for writing.

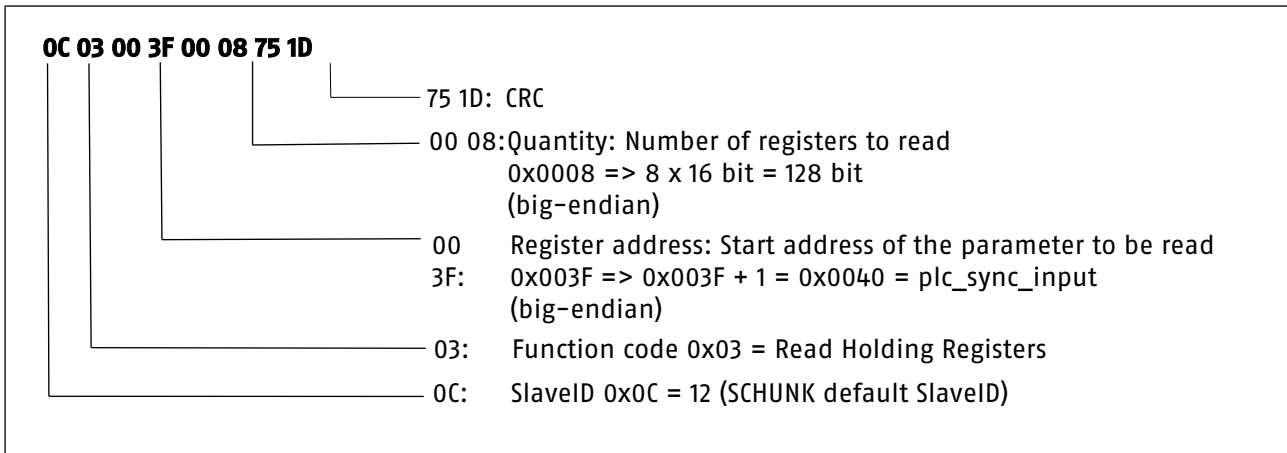
For controlling the module, in addition to the cyclic reading out of the parameter <plc\_sync\_input> (► 4.2 [50]), the cyclic writing of the parameter <plc\_sync\_output> (► 4.2 [50]) is also required:

- The master receives information about the status of the module by reading the status double word contained in the parameter <plc\_sync\_input>, the current position as well as the diagnostics double word (warning and error code).
- Accordingly, the master can control the module by writing the control double word contained in the parameter <plc\_sync\_output>, the position value, the velocity value and the gripping force value, ► 2 [12].

The following examples show a possible Modbus data exchange. Here the data is transferred via RS485, the master communicates with the module (standard slave ID 12 (0x0C)).

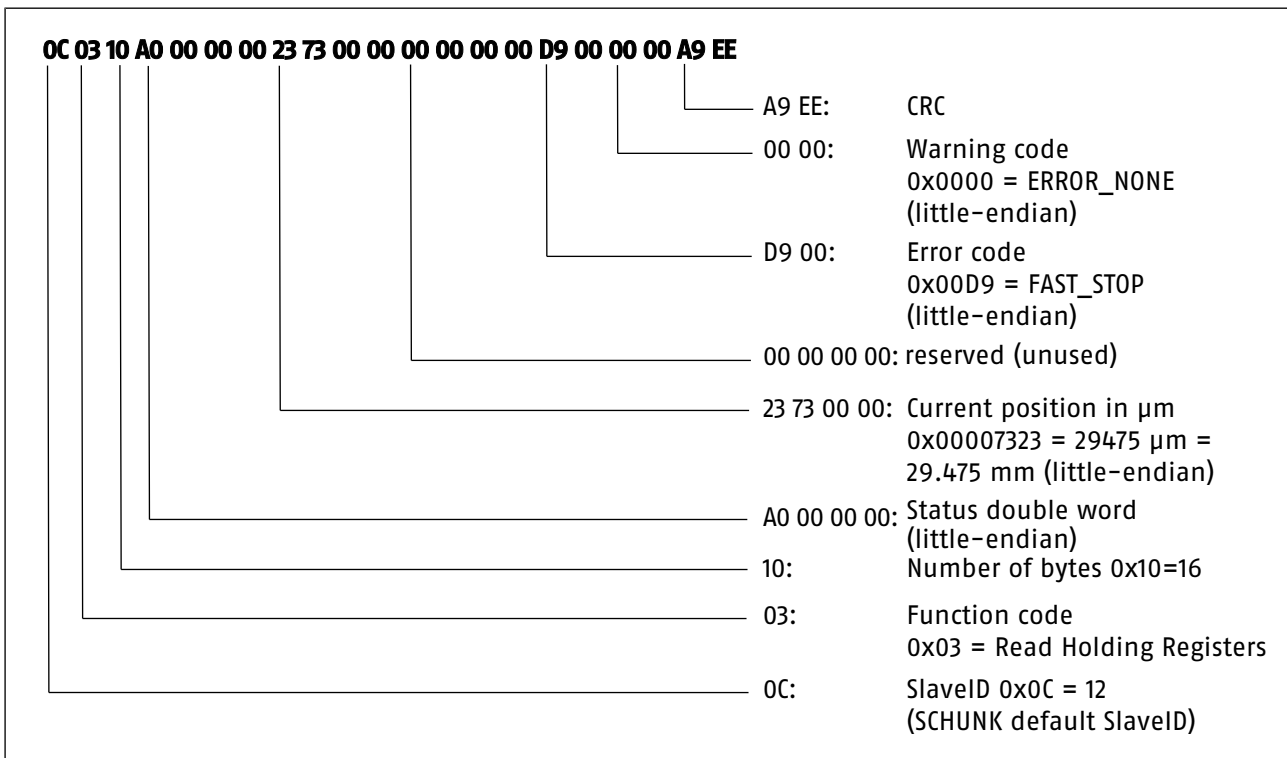
**Example 1**      **Reading the cyclic input data from the parameter <plc\_sync\_input>**

Request from the master to the slave (8 bytes):



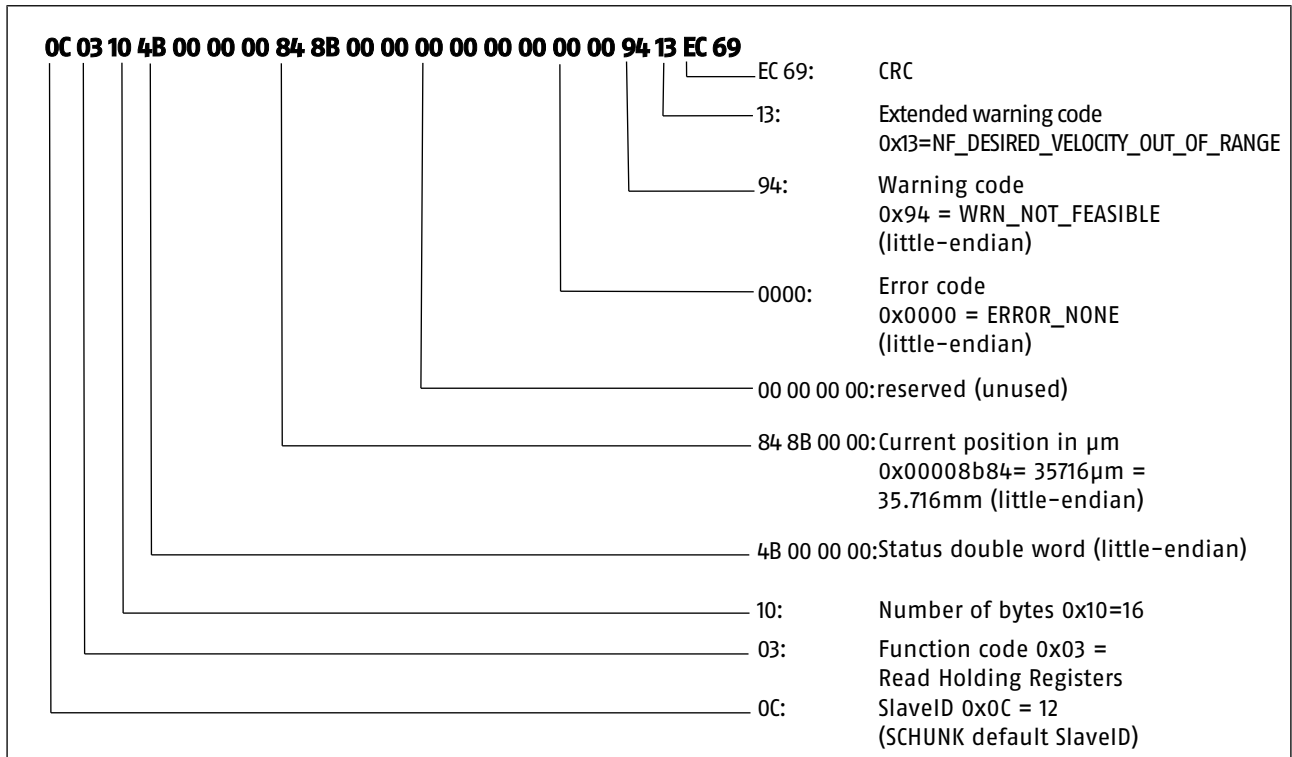
*Request from the master to the slave when reading input data*

Response from the slave to the master (21 bytes):



*Response from the slave to the master when reading input data*

Alternative response from the slave to the master (21 bytes) with extended warning code:

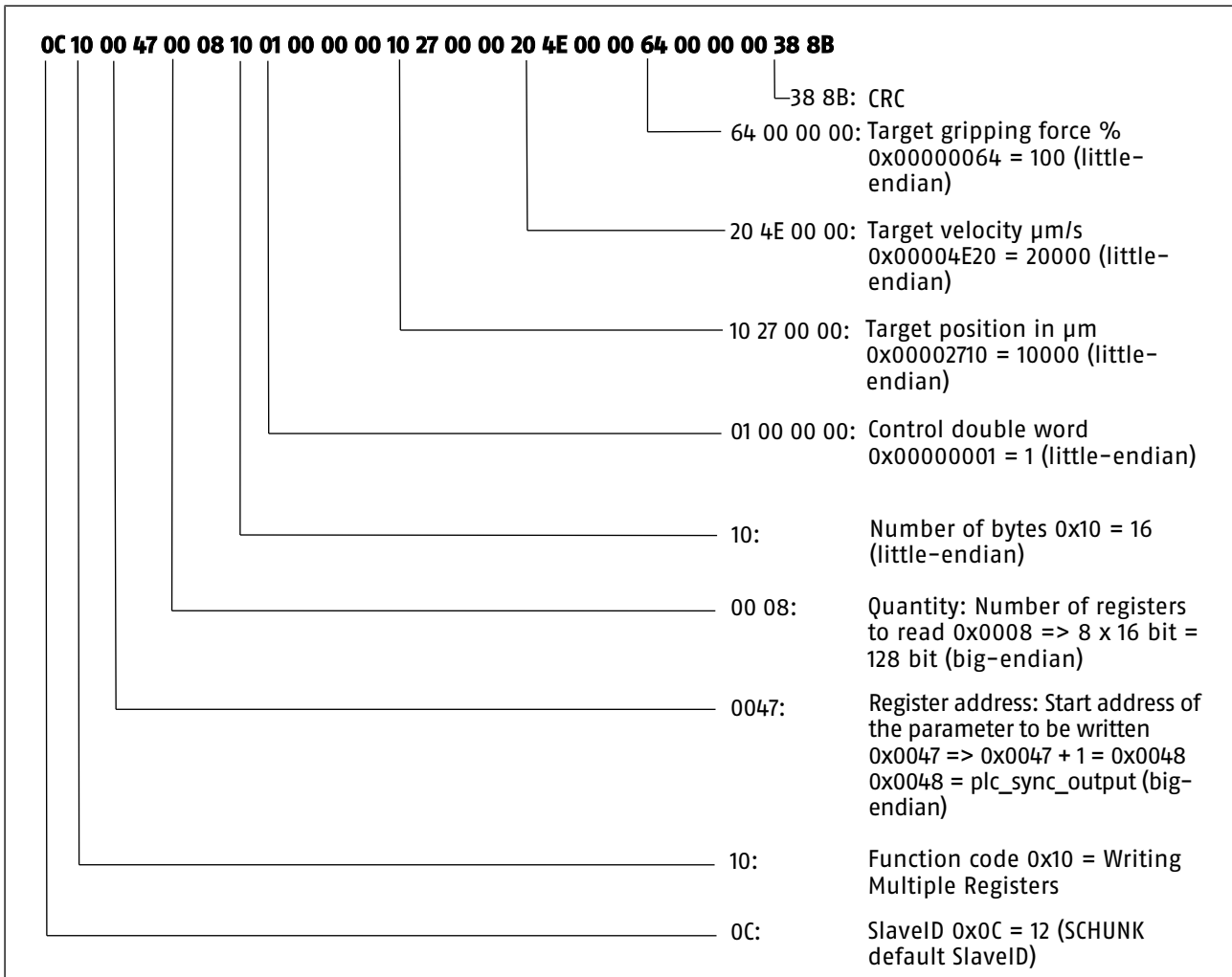


Response from the slave to the master when reading input data with extended warning code

**Example 2**

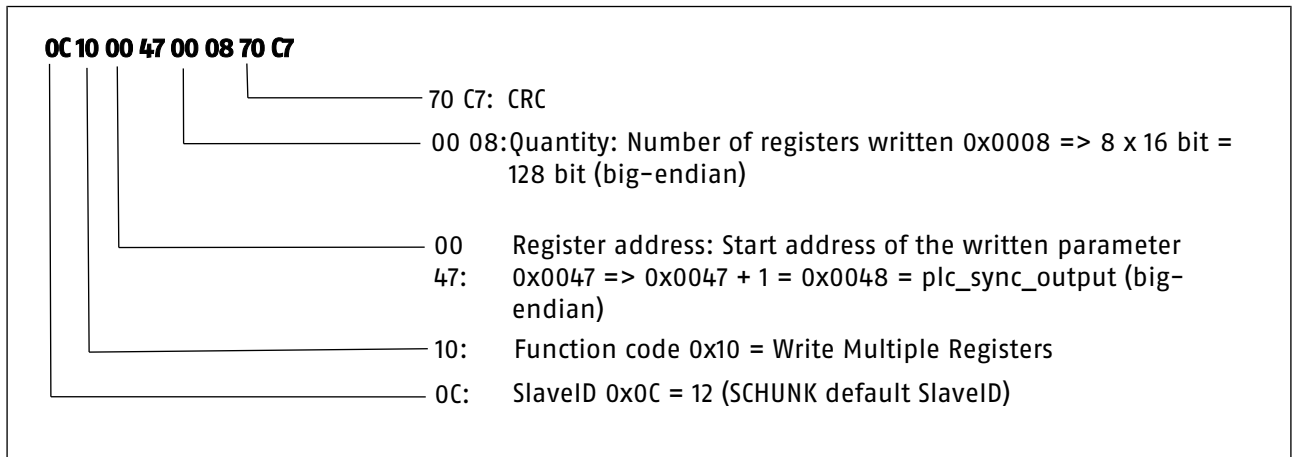
**Writing the cyclic output data to the parameter  
<plc\_sync\_output>**

Request from the master to the slave (25 bytes):



*Request from the master to the slave when writing output data*

Response from the slave to the master (8 bytes):



*Response from the slave to the master when writing output data*

---

**NOTE**

- It can be seen from the examples that the register address transmitted on the bus in each case is one less than the address actually addressed in the slave. This is a Modbus convention. Depending on the controller manufacturer, either the register address reduced by one or the standard register address must be specified. In the second case, the controller automatically adjusts the address during transmission.
  - Modbus RTU Protocol data with more than 8 bits, e.g. quantity (number of registers) or the register address are transmitted in big-endian format.
  - Module-specific parameters are transmitted in little-endian format.
-

## 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.

### 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.
  - If there is a warning that is not listed below, contact SCHUNK Service.
- 

### 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" (Bit 2) the acknowledgment of an existing warning is triggered, ▶ 7.2 [ 90].

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 self-acknowledging  
 acknowledge:

**HEX 0x92 / DEC 146****WRN\_MOT\_TEMP\_LO****WARNING\_MOTOR\_TEMP\_LOW**

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

**HEX 0x93 / DEC 147****WRN\_MOT\_TEMP\_HI****WARNING\_MOTOR\_TEMP\_HIGH**

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

**HEX 0x94 / DEC 148****WRN\_NOT\_FEASIBLE****WARNING\_CMD\_NOT\_FEASIBLE**

Diagnostic event: The control command sent to the module is not feasible.  
 Ability to acknowledgeable/self-acknowledging  
 acknowledge:

---

**NOTE**

- Further information on the cause of this warning is stored in the diagnostic memory. Read out diagnostic memory, see parameters <sys\_msg\_req> (► 4.2 [□ 51]).
  - If this warning is displayed, the reason for the non-feasibility can be read out via the cyclic data, ► 2.1.1 [□ 12]. The list of causes can be found in the attachment to this document, ► 7.4 [□ 96].
- 

**HEX 0x95 / DEC 149****WRN\_POS\_LIMIT****WARNING\_POSITION\_LIMIT\_REACHED**

Diagnostic event: The movement during jog mode was automatically terminated by reaching the minimum or maximum position.  
 Ability to self-acknowledging  
 acknowledge:

**HEX 0x96 / DEC 150****WRN\_LGC\_VOLT\_LO****WARNING\_LOGIC\_VOLTAGE\_LOW**

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

<b>HEX 0x97 / DEC 151</b>	<b>WRN_LGC_VOLT_HI</b> <b>WARNING_LOGIC_VOLTAGE_HIGH</b> Diagnostic event: The logic supply voltage measured is too high. Ability to self-acknowledging acknowledge:
<b>HEX 0x98 / DEC 152</b>	<b>WRN_MOT_VOLT_LO</b> <b>WARNING_MOTOR_VOLTAGE_LOW</b> Diagnostic event: The motor supply voltage measured is too low. Ability to self-acknowledging acknowledge:
<b>HEX 0x99 / DEC 153</b>	<b>WRN_MOT_VOLT_HI</b> <b>WARNING_MOTOR_VOLTAGE_HIGH</b> Diagnostic event: The motor supply voltage measured is too high. Ability to self-acknowledging acknowledge:

## 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.
  - If there is an error that is not listed below, contact SCHUNK Service.
- 

<b>Error state</b>	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 GPE: the GPE is activated.
<b>Error code</b>	Each detectable error event includes a unique error code that is transmitted in the cyclical input data.
<b>Acknowledging</b>	Errors can be separated into those requiring acknowledgment and errors that are non-acknowledgeable.  <b>Errors requiring acknowledgment:</b> 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. The error state cannot be exited. In cases such as this, contact SCHUNK Service.

**Recognizable error events**

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

**HEX 0x28 / DEC 040**

**ERR\_BT\_FAILED**  
**ERROR\_BRAKE\_TEST\_FAILED**

Diagnostic event: The brake test was performed unsuccessfully.  
Ability to requiring acknowledgment  
acknowledge:

**HEX 0x6C / DEC 108**

**ERR\_MOT\_TEMP\_LO**  
**ERROR\_MOTOR\_TEMP\_LOW**

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

**HEX 0x6D / DEC 109**

**ERR\_MOT\_TEMP\_HI**  
**ERROR\_MOTOR\_TEMP\_HIGH**

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

**HEX 0x70 / DEC 112**

**ERR\_LGC\_TEMP\_LO**  
**ERROR\_LOGIC\_TEMP\_LOW**

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

**HEX 0x71 / DEC 113**

**ERR\_LGC\_TEMP\_HI**  
**ERROR\_LOGIC\_TEMP\_HIGH**

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

HEX 0x72 / DEC 114

**ERR\_LGC\_VOLT\_LO****ERROR\_LOGIC\_VOLTAGE\_LOW**

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

HEX 0x73 / DEC 115

**ERR\_LGC\_VOLT\_HI****ERROR\_LOGIC\_VOLTAGE\_HIGH**

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

HEX 0x74 / DEC 116

**ERR\_MOT\_VOLT\_LO****ERROR\_MOTOR\_VOLTAGE\_LOW**

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

---

**NOTE**

For modules *with* GPE: As long as GPE is activated, this error is *not* monitored.

For modules *without* GPE: 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 requiring acknowledgment  
 acknowledge:

---

**NOTE**

For modules *with* GPE: As long as GPE is activated, this error is *not* monitored.

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

---

HEX 0xD5 / DEC 213

**ERR\_SOFT\_LOW****ERROR\_SOFT\_LOW**

Diagnostic event: The lower software limit has been reached or exceeded.

Ability to requiring acknowledgment  
 acknowledge:

<b>HEX 0xD6 / DEC 214</b>	<p><b>ERR_SOFT_HIGH</b> <b>ERROR_SOFT_HIGH</b></p> <p>Diagnostic event: The upper software limit has been reached or exceeded.</p> <p>Ability to acknowledge: requiring acknowledgment</p>
<b>HEX 0xD9 / DEC 217</b>	<p><b>ERR_FAST_STOP</b> <b>ERROR_FAST_STOP</b></p> <p>Diagnostic event: A fast stop was triggered.</p> <p>Ability to acknowledge: requiring acknowledgment</p>
<b>HEX 0xE4 / DEC 228</b>	<p><b>ERR_TOO_FAST</b> <b>ERROR_TOO_FAST</b></p> <p>Diagnostic event: The maximum permissible speed was exceeded by a factor of 1.2.</p> <p>Ability to acknowledge: requiring acknowledgment</p>
<b>HEX 0xEF / DEC 239</b>	<p><b>ERR_COMM_LOST</b> <b>ERROR_COMMUNICATION_LOST</b></p> <p>Diagnostic event: The communication link between the module and the receiver (controller or <i>Mechatronic Grippers</i> app Link ##) has been interrupted.</p> <p>Ability to acknowledge: requiring acknowledgment</p>
<b>HEX 0xF1 / DEC 241</b>	<p><b>ERR_MOV_ABORT_TO</b> <b>ERROR_MOVE_ABORT_TIMEOUT</b></p> <p>Diagnostic event: Positioning could not be performed within the expected period of time.</p> <p>Ability to acknowledge: requiring acknowledgment</p>
<b>HEX 0xF4 / DEC 244</b>	<p><b>ERR_MOVE_BLOCKED</b> <b>ERROR_MOVE_BLOCKED</b></p> <p>Diagnostic event: The drive was blocked.</p> <p>Ability to acknowledge: requiring acknowledgment</p>

## 7 Appendix

### 7.1 Application examples

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

#### NOTE

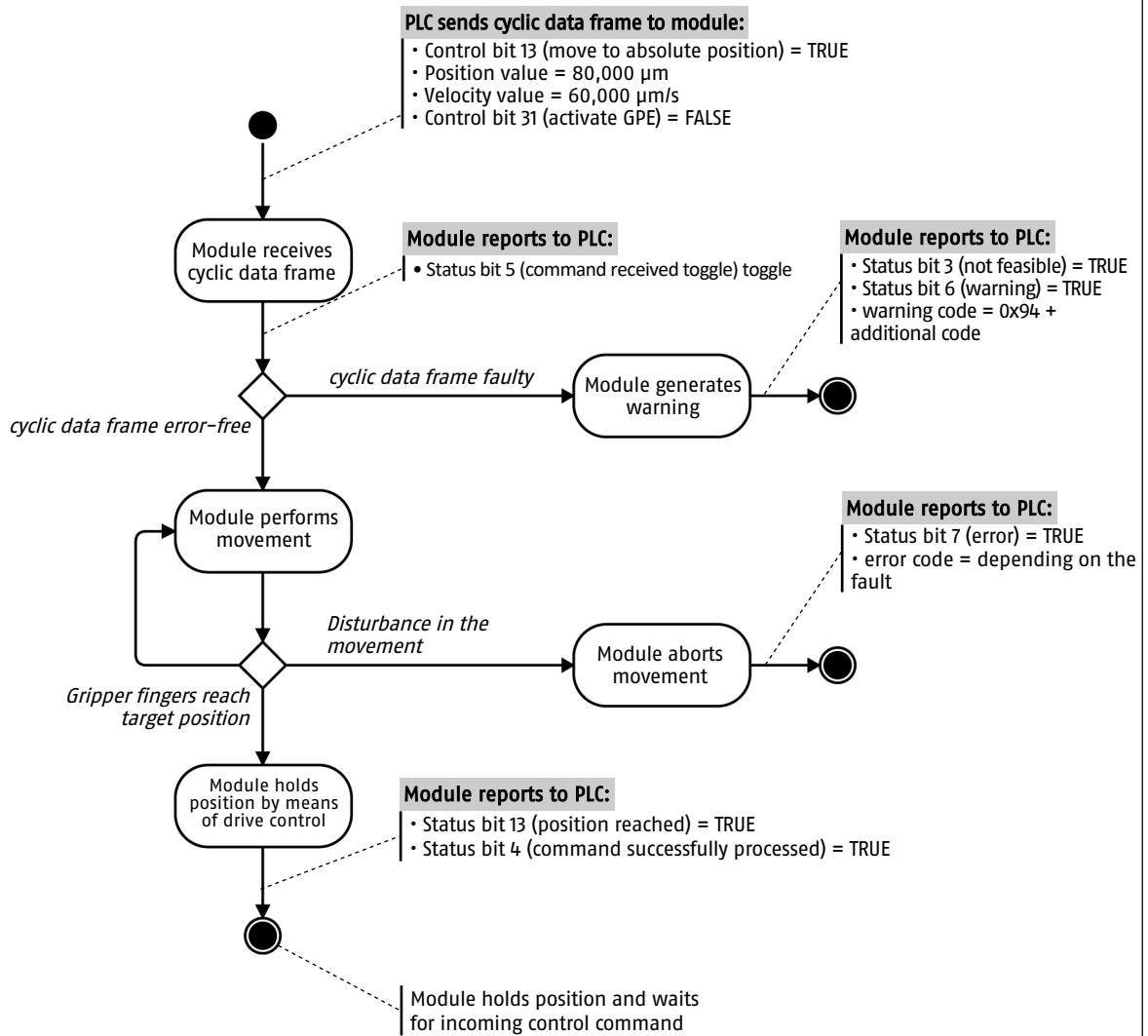
Due to the sometimes significant deviations between the modules, it is possible that the application examples described here may lead to different results for a specific module. For example, not every module is able to move to the absolute position 80 mm. However, this does not change the basic statements of the application examples.

Scenario description	Example
An absolute position movement is carried out.	▶ <a href="#">Example 1 [ 82]</a>
A relative position movement is carried out.	▶ <a href="#">Example 2 [ 83]</a>
A workpiece is gripped: <ul style="list-style-type: none"> <li>• without re-gripping</li> <li>• Workpiece holding by drive control</li> </ul>	▶ <a href="#">Example 3 [ 84]</a>
A workpiece is gripped: <ul style="list-style-type: none"> <li>• with re-gripping</li> <li>• Workpiece holding by GPE</li> </ul>	▶ <a href="#">Example 4 [ 85]</a>
A workpiece is gripped in StrongGrip mode: <ul style="list-style-type: none"> <li>• without re-gripping</li> <li>• Workpiece holding by GPE</li> </ul>	▶ <a href="#">Example 5 [ 86]</a>
A workpiece is gripped at the expected position: <ul style="list-style-type: none"> <li>• without re-gripping</li> <li>• Workpiece holding by GPE</li> </ul>	▶ <a href="#">Example 6 [ 87]</a>
A workpiece is gripped at the expected position: <ul style="list-style-type: none"> <li>• with re-gripping</li> <li>• Workpiece holding by drive control</li> </ul>	▶ <a href="#">Example 7 [ 88]</a>
A workpiece is gripped at the expected position in StrongGrip mode: <ul style="list-style-type: none"> <li>• without re-gripping</li> <li>• Workpiece holding by GPE</li> </ul>	▶ <a href="#">Example 8 [ 89]</a>

## EXAMPLE 1 Absolute positioning movement

### Application example – move to absolute position

- Movement to absolute position 80 mm
- Maximum permissible velocity of movement 60 mm/s
- Position holding by drive control prerequisites:
- Status bit 0 (ready for operation) = TRUE
- no movement of the gripper fingers

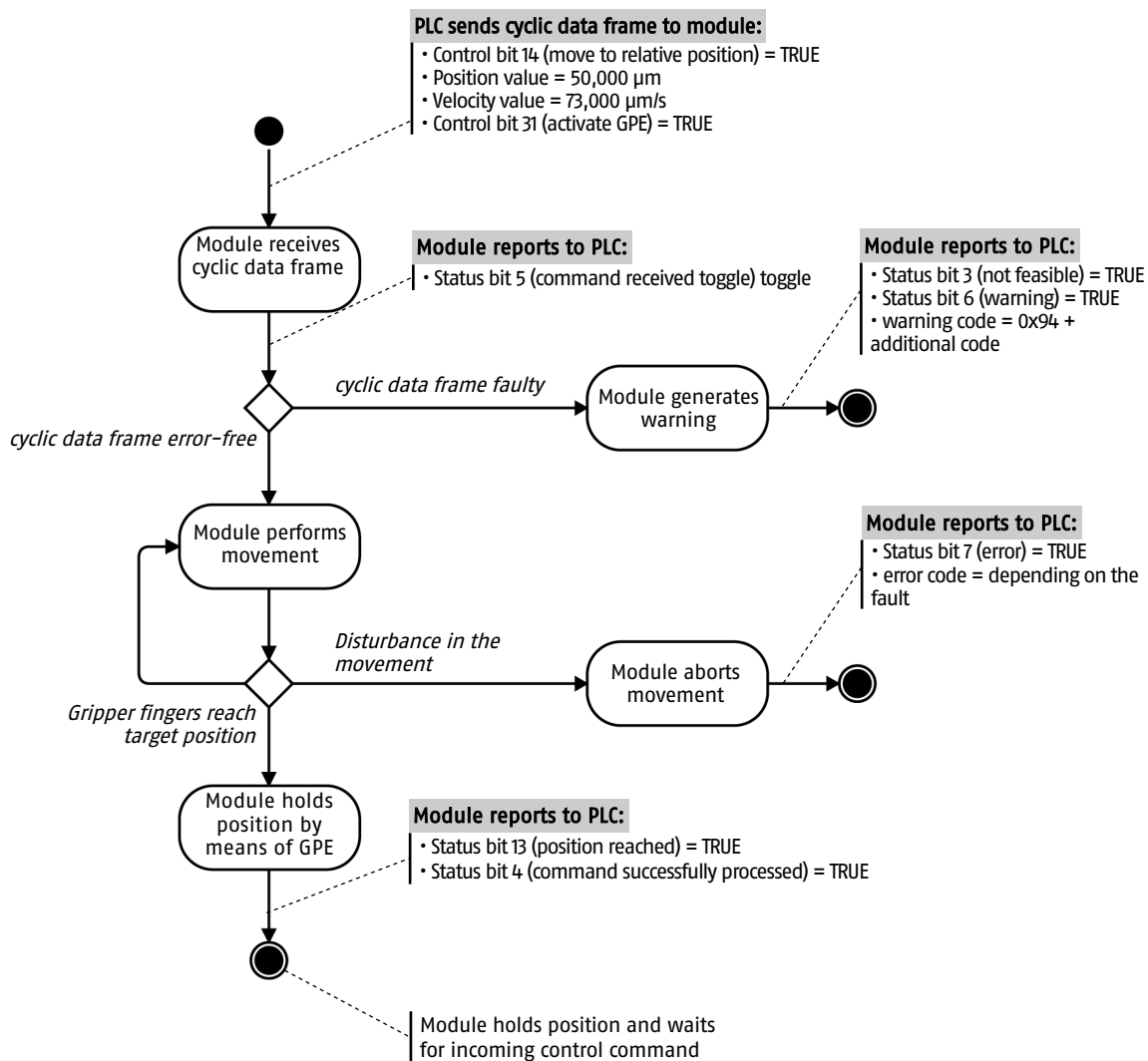


For further information, see chapter ▶ 3.2.2 [ 28].

## EXAMPLE 2 Relative positioning movement

### Application example – move to relative position

- Movement from current position -50 mm
- Maximum permissible velocity of movement 73 mm/s
- Position holding by GPE prerequisites:
- Status bit 0 (ready for operation) = TRUE
- no movement of the gripper fingers



For further information, see chapter ▶ 3.2.3 [ 30].

### EXAMPLE 3 Workpiece gripping (1)

#### Application example – grip workpiece

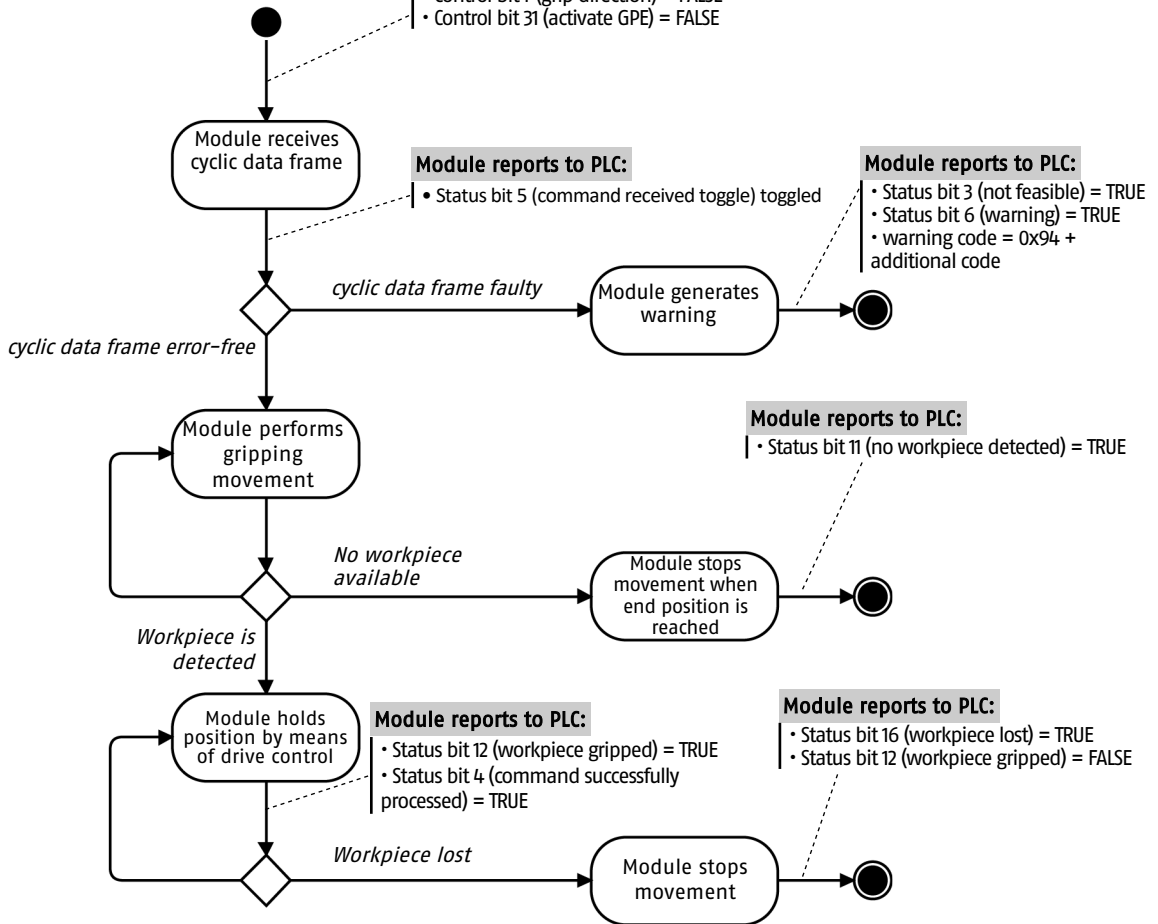
- Grip workpiece with 64% gripping force
- Grip workpiece from the outside
- without re-gripping
- Workpiece holding by drive control prerequisites:
- Status bit 0 (ready for operation) = TRUE
- no movement of the gripper fingers

**Parameterization of the module:**

- grp\_prehold\_time = 0 ms

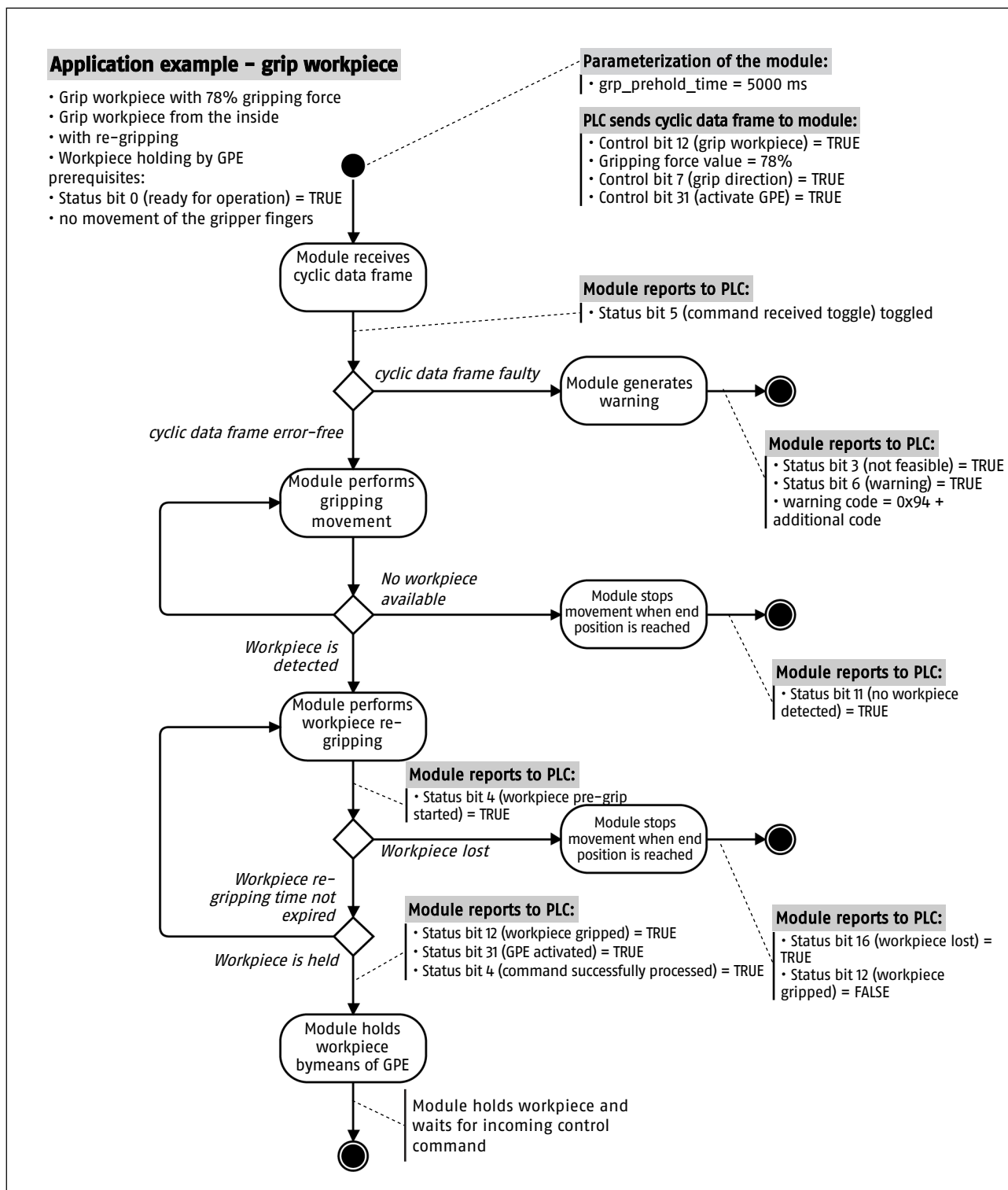
**PLC sends cyclic data frame to module:**

- Control bit 12 (grip workpiece) = TRUE
- Gripping force value = 64%
- Control bit 7 (grip direction) = FALSE
- Control bit 31 (activate GPE) = FALSE



For further information, see chapter ▶ 3.3.1 [ 33].

### EXAMPLE 4 Workpiece gripping (2)



For further information, see chapter ▶ 3.3.1 [ 33 ].

## EXAMPLE 5 Workpiece gripping in StrongGrip mode

### Application example – grip workpiece in StrongGrip mode

- Grip workpiece with 149% gripping force
- Grip workpiece from the outside
- without re-gripping
- Workpiece holding by GPE is automatically activated with StrongGrip

prerequisites:

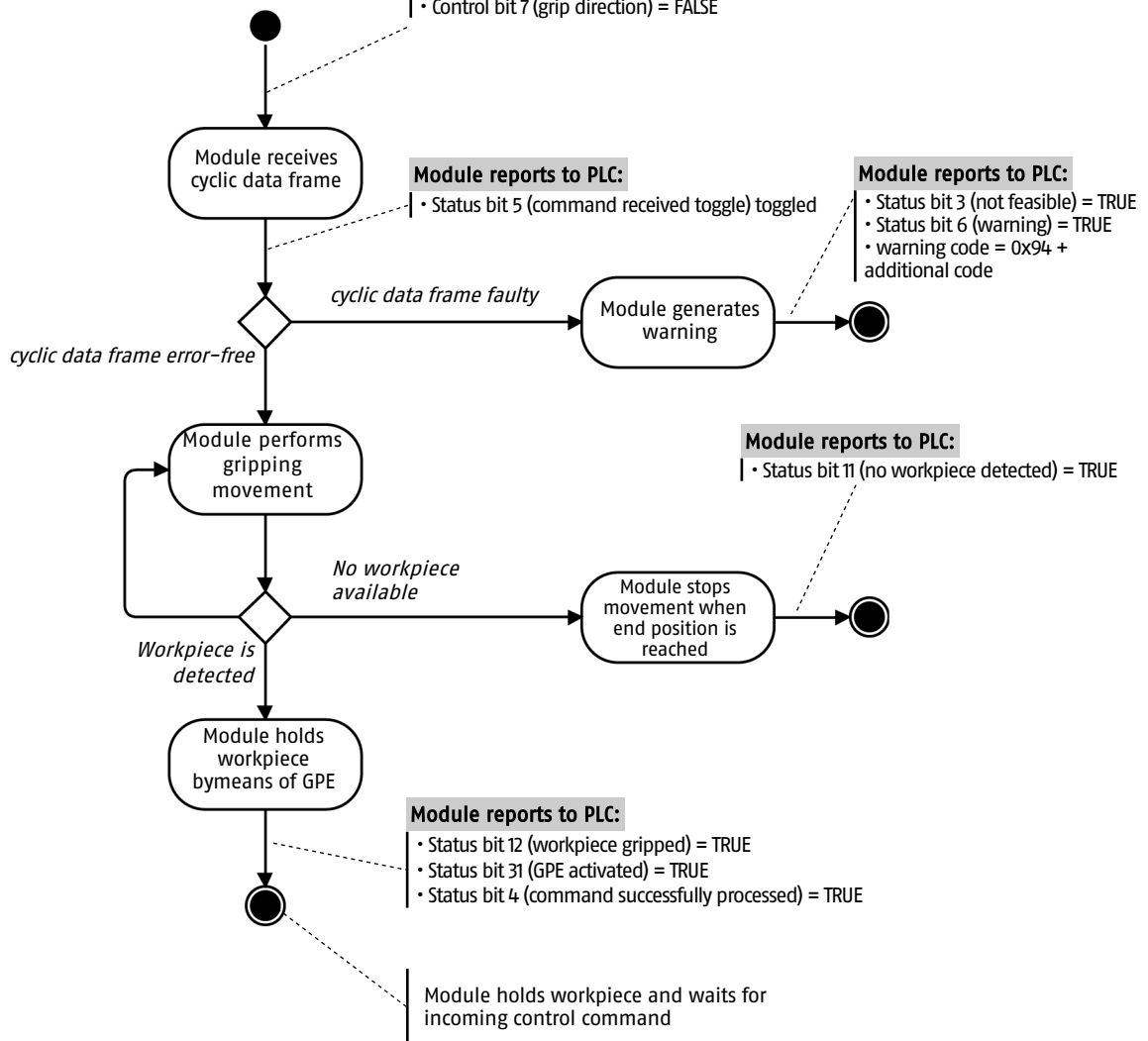
- Status bit 0 (ready for operation) = TRUE
- no movement of the gripper fingers

**Parameterization of the module:**

- `grp_prehold_time` = 0 ms

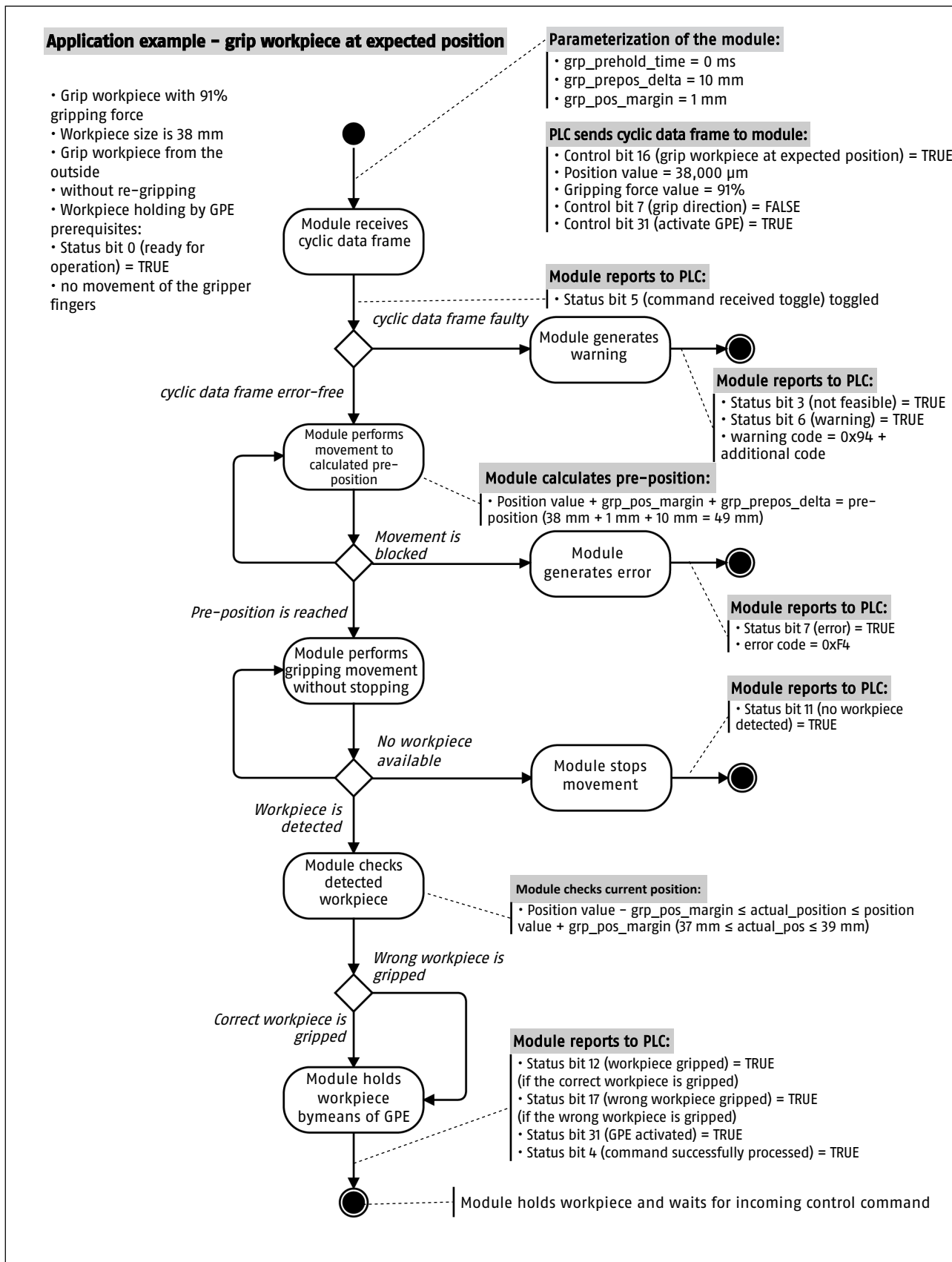
**PLC sends cyclic data frame to module:**

- Control bit 12 (grip workpiece) = TRUE
- Gripping force value = 149%
- Control bit 7 (grip direction) = FALSE



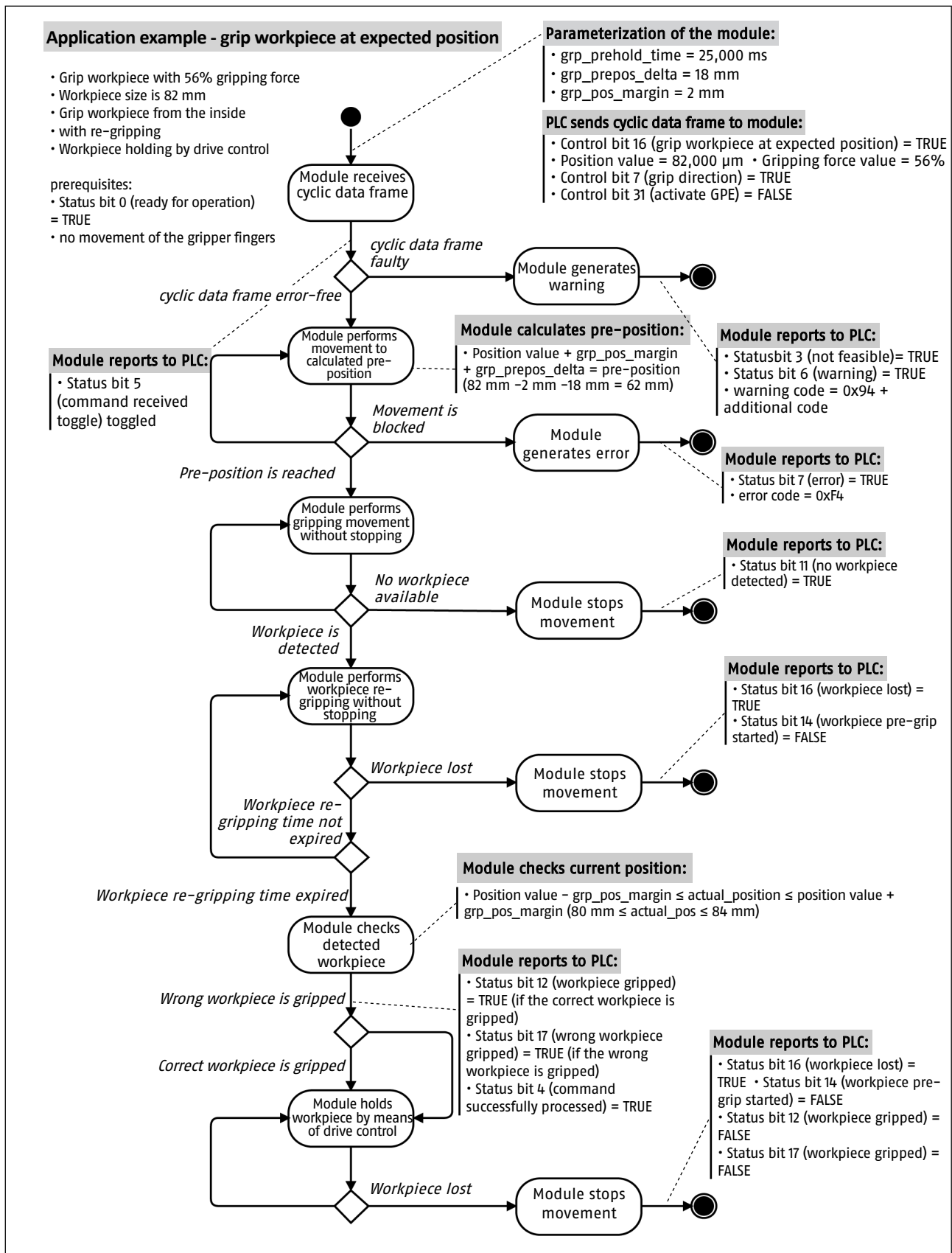
For further information, see chapter ▶ 3.3.1 [ 33 ].

### EXAMPLE 6 Workpiece gripping at expected position (1)



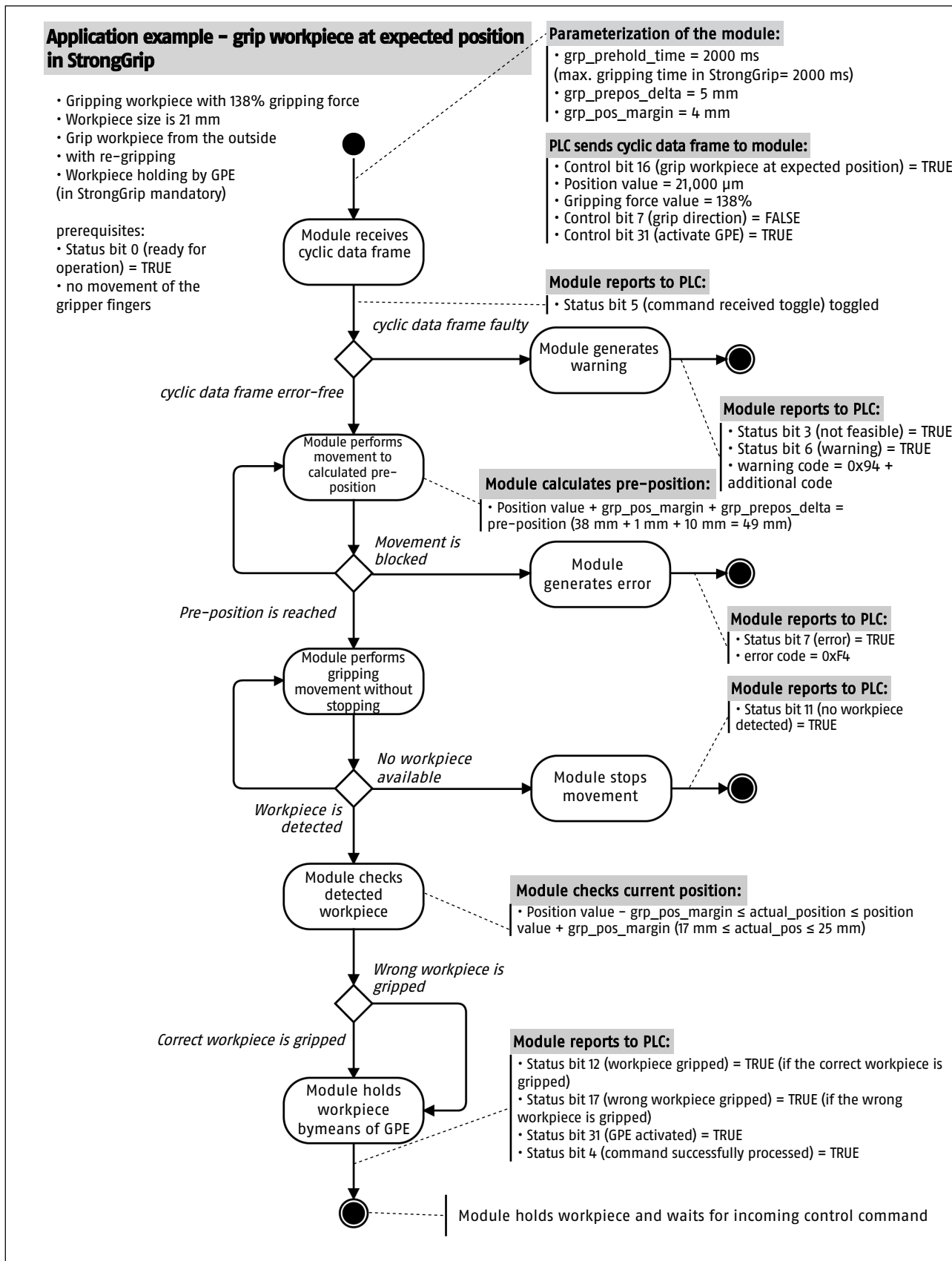
For further information, see chapter ▶ 3.3.2 [ 35].

### EXAMPLE 7 Workpiece gripping at expected position (2)



For further information, see chapter [3.3.2 \[35\]](#).

### EXAMPLE 8 Workpiece gripping at expected position in StrongGrip mode



For further information, see chapter ▶ 3.3.2 [ 35].

## 7.2 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 ▶ 2.1.1.1 [14].

### Bit 0 – fast stop

Edge change	Module reaction
0 -> 1	no reaction
1 -> 0	The module performs a quick stop, ▶ 3.2.5 [33].

### Bit 1 – stop

Edge change	Module reaction
0 -> 1	The module performs a controlled stop, ▶ 3.2.4 [32].
1 -> 0	no reaction

### Bit 2 – acknowledge

Edge change	Module reaction
0 -> 1	The module tries to acknowledge all existing warnings and errors, ▶ 6.1 [75], ▶ 6.2 [77].
1 -> 0	no reaction

### Bit 3 – prepare for shutdown

Edge change	Module reaction
0 -> 1	The module is preparing for shutdown, ▶ 3.1.2 [24].
1 -> 0	no reaction

### Bit 4 – softreset

Edge change	Module reaction
0 -> 1	The module is restarted on the software side, ▶ 3.1.3 [26].
1 -> 0	no reaction

### Bit 5 – release for manual movement

Edge change	Module reaction
0 -> 1	GPE is deactivated in order to manually remove a workpiece, ▶ 3.3.6 [44].
1 -> 0	no reaction

**Bit 6 – repeat command toggle**

Edge change	Module reaction
0 → 1	The module repeats the control command whose bit is still pending.
1 → 0	The module repeats the control command 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	As long as the bit is set, the module executes a movement in the negative direction of movement, ▶ 3.2.1 [427].
1 → 0	no reaction

**Bit 9 – jog mode positive**

Edge change	Module reaction
0 → 1	As long as the bit is set, the module executes a movement in the positive direction of movement, ▶ 3.2.1 [427].
1 → 0	no reaction

**Bit 10 – reserved**

Edge change	Module reaction
0 → 1	no reaction
1 → 0	no reaction

**Bit 11 – release workpiece**

Edge change	Module reaction
0 → 1	The module releases a workpiece, ▶ 3.3.5 [443].
1 → 0	no reaction

**Bit 12 – grip workpiece**

Edge change	Module reaction
0 -> 1	The module performs workpiece gripping, ▶ 3.3.1 [33]
1 -> 0	no reaction

**Bit 13 – move to absolute position**

Edge change	Module reaction
0 -> 1	The module performs a positioning movement to an absolute position, ▶ 3.2.2 [28].
1 -> 0	no reaction

**Bit 14 – move to relative position**

Edge change	Module reaction
0 -> 1	The module performs a positioning movement to a relative position, ▶ 3.2.3 [30].
1 -> 0	no reaction

**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 workpiece gripping at the expected position.
1 -> 0	no reaction

**Bit 17 – 29 – reserved**

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

**Bit 30 – brake test**

Edge change	Module reaction
0 -> 1	The module performs a brake test, ▶ 3.4.3 [47].
1 -> 0	no reaction

**Bit 31 – Activate grip force and position maintenance**

Status	Module reaction
0	Gripping forces and positions are held by the drive control.
1	Gripping forces and positions are held by the GPE.

**7.3 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 ▶ 2.1.1.2 [18].

**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	No information is provided in feedback.
1	The module is ready to be shut down.

**Bit 3 – not feasible**

Status	Module feedback
0	No information is reported.
1	The control command sent to the module is not feasible. ▶ 6.1 [76]

**Bit 4 – command successfully processed**

Status	Module feedback
0	No information is reported.
1	The following control commands sent to the module were successfully <i>processed</i> . <ul style="list-style-type: none"> <li>• Bit 1 – stop</li> <li>• Bit 8 – jog mode negative</li> <li>• Bit 9 – jog mode positive</li> <li>• Bit 11 – release workpiece</li> <li>• Bit 12 – grip workpiece</li> <li>• Bit 13 – move to absolute position</li> <li>• Bit 14 – move to relative position</li> <li>• Bit 16 – grip workpiece at expected position</li> </ul>

**Bit 5 – command received toggle**

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

**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 – released for manual movement**

Status	Module feedback
0	No information is provided in feedback.
1	Module is ready for manual removal of a workpiece.

**Bit 9 – software limit reached**

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

**Bit 10 – reserved**

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

**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	The module has started re-gripping.

**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	During workpiece gripping at the expected position, the wrong workpiece was gripped.

**Bit 18 – 30 – reserved**

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

**Bit 31 – Grip force and position maintenance activated**

Status	Module feedback
0	GPE is not active.
1	GPE is active.

## 7.4 Additional code when warning is present

### WRN\_NOT\_FEASIBLE

If the warning WRN\_NOT\_FEASIBLE is issued, an additional code can be read out. The following is a list of possible causes for non-execution.

<b>HEX 0x00</b>	<b>NF_NO_REASON</b> There is no warning.
<b>HEX 0x01</b>	<b>NF_IOLINK_FUNCTION_NOT_SUPPORTED</b> A customer-specific function was triggered acyclically via the IO-Link master, which is not supported by the module.
<b>HEX 0x02</b>	<b>NF_IOLINK_RESET_CONDITION_ONLY_ALLOWED_IN_ERROR_STATE</b> The module was not in an error state when one of the following functions was triggered via the IO-Link master: <ul style="list-style-type: none"> <li>• Application Reset</li> <li>• Restore Factory Settings</li> <li>• Back-To-Box</li> </ul>
<b>HEX 0x03</b>	<b>NF_SHUTDOWN_NOT_FEASIBLE_IN_CURRENT_STATE</b> The shutdown is not permitted to be triggered from the current state, ▶ 3.1.2 [□ 24].
<b>HEX 0x04</b>	<b>NF_RESET_NOT_FEASIBLE_IN_CURRENT_STATE</b> The restart is not permitted to be triggered from the current state, ▶ 3.1.3 [□ 26].
<b>HEX 0x05</b>	<b>NF_COMMAND_NOT_ALLOWED_IN_CURRENT_STATE</b> Reset to factory settings is not permitted to be triggered from the current state, ▶ 3.4.6 [□ 48].
<b>HEX 0x06</b>	<b>NF_COMMAND_NOT_ALLOWED_IN_ERROR_STATE</b> The module was in an error state when the function was triggered.

<b>HEX 0x08</b>	<b>NF_SOFT_RESET_DISABLED_BY_PARAMETER</b> The restart is not enabled by the parameter <enable_softreset>, ▶ 4.2 [□ 65].
<b>HEX 0x09</b>	<b>NF_CANNOT_TRIGGER_COMMAND_WHILE_FAST_STOP_IS_ACTIVE</b> The status bit "fast stop" (bit 0) was reset when a function was triggered.
<b>HEX 0x0A</b>	<b>NF_MULTIPLE_COMMANDS_TRIGGERED_SIMULTANEOUSLY</b> Several functions are to be triggered at the same time.
<b>HEX 0x0C</b>	<b>NF_COMMAND_NOT_ALLOWED_DURING_BRAKE_TEST</b> The module was performing a brake test when an improper function was triggered.
<b>HEX 0x0D</b>	<b>NF_RELEASE_BRAKE_ONLY_ALLOWED_IN_ERROR_STATE</b> The module was not in an error state when manual workpiece removal was triggered, ▶ 3.3.6 [□ 44].
<b>HEX 0x0E</b>	<b>NF_RELEASE_WORKPIECE_COMMAND_ONLY_ALLOWED_WHILE_GRIPPING</b> The module was not holding a workpiece when the workpiece release was triggered, ▶ 3.3.5 [□ 43].
<b>HEX 0x0F</b>	<b>NF_COMMAND_NOT_ALLOWED_WHILE_HOLDING_A_WORKPIECE</b> While the module was holding a workpiece, an impermissible function was triggered.
<b>HEX 0x10</b>	<b>NF_DESIRED_POSITION_OUT_OF_RANGE</b> The target position resulting from the cyclically transmitted position value is outside the position limits, ▶ 1.2.1 [□ 8].
<b>HEX 0x11</b>	<b>NF_CURRENT_POSITION_ALREADY_INSIDE_WORKPIECE</b> Workpiece gripping at the expected position was triggered with an invalid combination of workpiece position and gripping direction. Impermissible combinations are as follows: <ul style="list-style-type: none"> <li>• Workpiece position &gt; current position and gripping direction inwards</li> <li>• Workpiece position &lt; current position and gripping direction outwards</li> </ul>
<b>HEX 0x12</b>	<b>NF_COMMAND_NOT_ALLOWED_DURING_GRIPPING_COMMAND</b> While the module was executing a gripping movement, an impermissible function was triggered.
<b>HEX 0x13</b>	<b>NF_DESIRED_VELOCITY_OUT_OF_RANGE</b> The cyclically transmitted speed value is outside the permissible speed limits, ▶ 4.2 [□ 57] and ▶ 4.2 [□ 57].

<b>HEX 0x15</b>	<b>NF_COMMAND_NOT_ALLOWED_DURING_MOVE_TO_POSITION</b> While the module was executing a positioning movement, an impermissible function was triggered.
<b>HEX 0x1D</b>	<b>NF_COMMAND_NOT_ALLOWED_DURING_RELEASE_BRAKE_COMMAND</b> While the module was in the state in which a workpiece can be removed manually, an impermissible function was triggered.
<b>HEX 0x1E</b>	<b>NF_COMMAND_NOT_ALLOWED_DURING_RELEASE_WORKPIECE_COMMAND</b> While a workpiece was released, an impermissible function was triggered.
<b>HEX 0x22</b>	<b>NF_MINIMUM_POSITION_OUT_OF_RANGE</b> A value that is outside the permissible limits was supposed to be written to the <min_pos> parameter, ▶ 4.2 [ 55].
<b>HEX 0x23</b>	<b>NF_MINIMUM_POSITION_ABOVE_MAXIMUM_POSITION</b> A value greater than the value of the parameter <max_pos> should be written to the parameter <min_pos>.
<b>HEX 0x24</b>	<b>NF_MAXIMUM_POSITION_OUT_OF_RANGE</b> A value that is outside the permissible limits was supposed to be written to the <max_pos> parameter, ▶ 4.2 [ 56].
<b>HEX 0x25</b>	<b>NF_MAXIMUM_POSITION_BELOW_MINIMUM_POSITION</b> A value was supposed to be written into the parameter <max_pos> that is smaller than the value of the parameter <min_pos>.
<b>HEX 0x26</b>	<b>NF_RELEASE_WORKPIECE_WOULD_VIOLATE_SOFTWARE_LIMIT</b> The target position during workpiece release is outside the position limits, ▶ 3.3.5 [ 43].
<b>HEX 0x27</b>	<b>NF_MOVEMENT_INTO_WORKPIECE_NOT_ALLOWED</b> The target position when releasing a workpiece by absolute or relative position travel is located within the workpiece.
<b>HEX 0x28</b>	<b>NF_GPE_FEATURE_NOT_AVAILABLE_ON_GRIPPER_WITHOUT_BRAKE</b> The GPE should be used to hold a workpiece or position, although the module does not have a GPE.
<b>HEX 0x29</b>	<b>NF_DESIRED_FORCE_OUT_OF_RANGE</b> The cyclically transferred gripping force value is outside of the permissible gripping force limits, ▶ 4.2 [ 58] and ▶ 4.2 [ 58].

## 7.5 Software copyright notices

### Modbus RTU Stack

FreeModbus Library: a portable Modbus implementation for Modbus ASCII/RTU.

Copyright (c) 2006–2018 Christian Walter <cwalter@embedded-solutions.at>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

### Texas Instruments F2837xS Support Library, F021 Flash API

TI Release: F2837xS Support Library v3.12.00.00

Release Date: Fri Feb 12 19:06:50 IST 2021

Copyright:

Copyright (C) 2014–2021 Texas Instruments Incorporated – <http://www.ti.com/>

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Texas Instruments Incorporated nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## TMG IOL Stack

Copyright (c) 2017, TMG Technologie und Engineering GmbH  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of TMG Technologie und Engineering GmbH nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,

SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## **Loki AssocVector**

The Loki Library

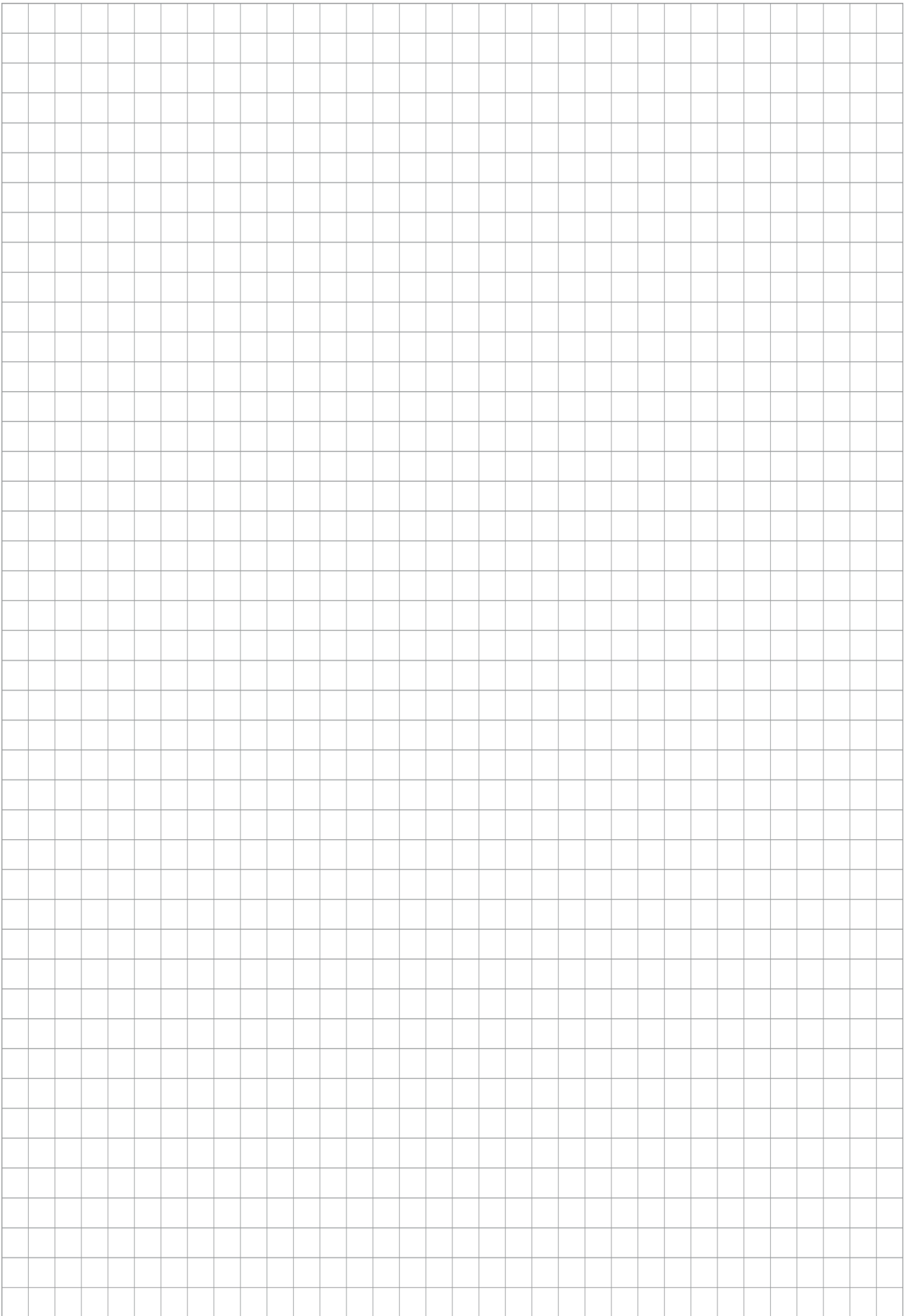
Copyright (c) 2001 by Andrei Alexandrescu

This code accompanies the book:

Alexandrescu, Andrei. "Modern C++ Design: Generic Programming and Design Patterns Applied." Copyright (c) 2001. Addison-Wesley.

Permission to use, copy, modify, distribute and sell this software for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

The author or Addison-Wesley Longman make no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.





**SCHUNK SE & Co. KG**  
Spanntechnik | Greiftechnik | Automatisierungstechnik

Bahnhofstr. 106 - 134  
D-74348 Lauffen/Neckar  
Tel. +49-7133-103-0  
info@de.schunk.com  
schunk.com

Folgen Sie uns | *Follow us*



Wir drucken nachhaltig | *We print sustainable*