GCL Gripper Control Language
Reference Manual
WSG-GCL
Software manual - Firmware Version 4.0.x
Dear Customer,

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

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

Best regards,

Your SCHUNK team

SCHUNK GmbH & Co. KG
Spann- und Greiftechnik
Bahnhofstr. 106 – 134
D-74348 Lauffen/Neckar
Tel. +49-7133-103-0
Fax +49-7133-103-2399
info@de.schunk.com
schunk.com
Table of contents

1 Introduction ........................................................................................................................................... 5
  1.1 Connecting to the WSG .................................................................................................................. 5
  1.2 Communicating with the gripper ...................................................................................................... 7
  1.3 Error Handling ................................................................................................................................ 7
  1.4 Connecting to the command interface using PuTTY ........................................................................ 8

2 Basic Command Set ............................................................................................................................... 11
  2.1 Interface Control ............................................................................................................................. 11
    2.1.1 VERBOSE – Enable verbose mode ......................................................................................... 11
    2.1.2 BYE – Disconnect from interface .......................................................................................... 12
  2.2 Motion Control ............................................................................................................................... 13
    2.2.1 HOME – Referencing the gripper ......................................................................................... 13
    2.2.2 MOVE – Move the gripper fingers ....................................................................................... 14
    2.2.3 GRIP – Grip part ..................................................................................................................... 15
    2.2.4 RELEASE – Release part ....................................................................................................... 18
    2.2.5 PWT – Get or set part width tolerance .................................................................................. 19
    2.2.6 CLT – Get or set clamping travel ............................................................................................ 20
  2.3 Gripper State .................................................................................................................................... 21
    2.3.1 POS – Query current position of gripper jaws ..................................................................... 21
    2.3.2 SPEED – Query current speed of gripper jaws .................................................................... 21
    2.3.3 FORCE – Query current gripping force .................................................................................. 22
    2.3.4 GRIPSTATE – Query gripper state ....................................................................................... 22

3 Extended Command Set ....................................................................................................................... 23
  3.1 System Commands ........................................................................................................................... 23
    3.1.1 DEVTYPE – Query device type ............................................................................................ 23
    3.1.2 VERSION – Query firmware version ..................................................................................... 23
    3.1.3 SN – Query serial number ...................................................................................................... 24
    3.1.4 TAG – Query descriptor string (device tag) .......................................................................... 24
    3.1.5 SYSFLAGS – Query system state flags ................................................................................... 25
    3.1.6 TEMP – Query temperature ................................................................................................... 25
    3.1.7 AUTOSEND – Enable or disable auto-sending of parameters ............................................... 26
  3.2 Extended Motion Control .................................................................................................................. 28
    3.2.1 STOP – Stop motion ................................................................................................................ 28
    3.2.2 FASTSTOP – Issue fast stop ................................................................................................... 29
    3.2.3 FSACK – Acknowledge Fast Stop ............................................................................................ 30
  3.3 Extended Gripper State .................................................................................................................... 31
    3.3.1 TARE – Tare force sensor ....................................................................................................... 31
### Table of contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3.2</td>
<td>GRIPSTATS – Query gripper statistics</td>
<td>32</td>
</tr>
<tr>
<td>3.4</td>
<td>Finger Interface</td>
<td>33</td>
</tr>
<tr>
<td>3.4.1</td>
<td>FDATA – Query finger data</td>
<td>33</td>
</tr>
<tr>
<td>3.4.2</td>
<td>FTYPE – Query finger type</td>
<td>33</td>
</tr>
<tr>
<td>3.4.3</td>
<td>FFLAGS – Query finger flags</td>
<td>34</td>
</tr>
<tr>
<td>4</td>
<td>Appendix A: Status Codes</td>
<td>35</td>
</tr>
<tr>
<td>5</td>
<td>Appendix B: System State Flags</td>
<td>37</td>
</tr>
<tr>
<td>6</td>
<td>Appendix C: Finger Flags</td>
<td>41</td>
</tr>
<tr>
<td>7</td>
<td>Appendix D: Gripper States</td>
<td>42</td>
</tr>
<tr>
<td>8</td>
<td>Appendix E: Data Types</td>
<td>44</td>
</tr>
</tbody>
</table>
1 Introduction

This manual describes the Gripper Control Language (GCL). GCL is a text based protocol that can be used to control the gripper.

The WSG family of gripping modules can be controlled by different standard interfaces, each of which supports one or more different communication protocols (TCP/IP or UDP/IP).

The following chapters provide a detailed explanation of the protocol itself as well as of the product’s command set.

To get started with the communication protocol, SCHUNK recommends using a common Telnet client like the free PuTTY1 for Microsoft Windows, (☞ 1.4, Page 8)

1.1 Connecting to the WSG

Before connecting to the gripper, an appropriate interface must be selected using the gripper’s web interface.

✔ Network settings of the computer are configured.
✔ A suitable IP address for the local network is set on the gripper.

Further information see Assembly and Operating Manual of the gripper

1 Connect the gripper with the local network or directly to your computer’s network interface.

2 Start the web browser and enter the default address http://192.168.1.20 into the address bar.

⇒ Web interface is displayed.

3 Choose tab "Settings" -> "Command Interface".

4 Choose interface "TCP" or "UDP".

Note: SCHUNK recommends that you use the TCP protocol.

Use "UDP" only then, if you know the basic command format.

5 Enable setting "Use text based interface".

6 Click the "Apply" button to save your settings.
Enabling GCL on the web interface
1.2 Communicating with the gripper

Regardless of the interface "TCP" or "UDP" chosen above, the gripper now waits for incoming connections using the text-based protocol "GCL".

**Text format**
The gripper expects commands being submitted as plain ASCII strings. Each command must be terminated by a line feed character ('\n' or ASCII code 0x0d). Return messages are submitted in the same format and are terminated by a line feed character, too.

**NOTE**
The commands of the GCL Gripper Control Language are not case sensitive, i.e. sending "move(50)" is the same as "MOVE(50)" or "mOVe(50)". Response messages will, however, always be sent in upper case notation.

1.3 Error Handling

In case of an error, for example, if a command was not understood or a command could not be executed as desired, the gripper returns a message string of the following format:

**Standard mode**

```
ERR <command_name> <error_code>
```

- `<command_name>` Represents the command that caused the error
- `<error_code>` Number that describes the error. Available error codes, (☞ 4, Page 35).

**Extended error mode**

If verbose mode is active, the gripper submits extended error messages containing an additional text string that describes the type of error:

```
ERR <command_name> <error_code> <description_string>
```

- `<command_name>` see above
- `<error_code>` see above
- `<description_string>` Extended description of the error in the text format

- Description of the error codes, (☞ 4, Page 35)
- Description to enable extended error mode ("Verbose Mode"), (☞ 2.1.1, Page 11)
1.4 Connecting to the command interface using PuTTY

PuTTY is a free Telnet and SSH client that can be used to connect to the gripper’s command interface.

The following chapter shows how to use PuTTY with the gripper.

- The gripper is configured to use the "TCP" interface and the text based interface ("Use text based interface") is selected ([1.1, Page 5]).

   Note: On Unix-like systems, an equivalent command line Telnet client should be available which can be used, too.

2 Start PuTTY.

   ![PuTTY connection settings](image)

3 Insert IP address of the gripper and the port number of the command interface.
   Note: The default IP address of the module is 192.168.1.20, the default TCP/IP listening port is 1000.

4 Set the connection type to "raw".
NOTE
As the gripper does not send a carriage return character (\r' or ASCII code 0x0d) in its response messages, PuTTY must be configured to explicitly do a carriage return on each line feed character.

5 Select the "Terminal" tab.

6 Select "Implicit CR in every LF". This setting may be different when using other Telnet clients.

7 Click "Open" button.
   ⇒ Connection to gripper is established.
   ⇒ A new and empty terminal window appears.
8 Type the command and confirm with "Enter". For example, to reference the gripper, type "HOME ()", (\textit{\textbf{\textcolor{red}{2.2.1, Page 13}}}).

⇒ The response messages received by the gripper are displayed in the command window.
2 Basic Command Set

This chapter describes the basic command set that can be used to grip parts. For an extended set of commands providing more functionality, (☞ 3, Page 23).

2.1 Interface Control

2.1.1 VERBOSE – Enable verbose mode

Enables the interface’s verbose mode. By default, verbose mode is turned off, meaning that the module only returns a numeral error code with its error messages. By enabling verbose mode, the module additionally returns a text string describing the error.

Syntax
VERBOSE=<integer>

Parameter
Integer value telling whether verbose mode should be enabled (1) or disabled (0).

Response message
VERBOSE=<integer>  e.g. VERBOSE=0
2.1.2 BYE – Disconnect from interface

To disconnect safely from the module, the termination of the connection must be announced before closing. Otherwise the termination of the connection is considered as an error.

**NOTE**
If the disconnect is not announced before closing the connection, a FAST STOP will be raised blocking all further motion commands.

**Syntax**
BYE()

**Parameter**
No parameters

**Response message**
ACK BYE to acknowledge the command
2.2 Motion Control

2.2.1 HOME – Referencing the gripper

Execute a homing sequence to reference the gripper fingers. If no parameter is given, referencing will be done in default direction. This command has to be executed prior to any other motion-related command. The direction of homing can be either explicitly specified or can be obtained from the gripper’s configuration.

During homing, the gripper moves its fingers into the specified direction until it reaches its mechanical end stop. The blocking position is used as new origin for all motion-related commands.

NOTE
The best positioning performance will be achieved if homing is done into the direction you require the better positioning accuracy.

During homing soft limits are disabled!

Obstacles in the movement range of the fingers and collision with these during homing may result in a wrong reference point for the finger position!

Syntax
HOME()
HOME( <bool> )

Parameter
<bool> (optional)   Direction of referencing.
   If "1", referencing will be done in positive direction (to outside)
   If "0", referencing will be done in negative direction (to inside)

Response message
ACK HOME   to immediately acknowledge receipt of the message
FIN HOME   after successful completion of the command
2.2.2 MOVE – Move the gripper fingers

The MOVE command is intended to position the gripper jaws between the gripping cycles, e.g. to move the jaws quickly before softly gripping sensitive parts.

The command expects one or two parameters of which the first one indicates the target position in millimeters to which the gripper jaws should be moved and the second parameter indicates a speed limit in millimeters per second.

**NOTE**
Do not use the MOVE command to grip or release parts. Any blocking of the gripper fingers causes an error message. Use the GRIP and RELEASE command instead.

**Syntax**
MOVE( <float> )
MOVE( <float>, <float> )

**Parameter**

- <float> Position in mm
- <float> Speed limit in mm/s (optional)

**Response message**

- ACK MOVE to immediately acknowledge receipt of the message
- FIN MOVE after successful completion of the command
2.2.3 GRIP – Grip part

Grip a part. The command’s behavior depends on the number of given parameters:

1. No Parameter.
   Grip inside until a part is detected. Use default force and speed.

2. One parameter: Force
   Grip inside until a part is detected. Use the given force (in N).

3. Two parameters: Force, Part width
   Grip inside or outside (depending on the current position and the target position). Expect a part at the given position. Use the given force (in N). If the gripper detects a contact outside the part width tolerance, it is interpreted as a collision and an error is returned.

   Like 3 but use the additional speed limit (in mm/s).

NOTE
Case 1 and 2: As no part width is given, the gripper will grip itself if there is no part between the fingers and the part detection will always set the gripper state to HOLDING. You might check the position of the gripper jaws after gripping to make sure a part has been gripped.

If a part width is passed to the GRIP command (i.e. case 3 or 4 in the list above) and the gripper can establish the desired force within the defined clamping travel, the gripper state is set to HOLDING (Gripping part recognition see figure "Part width tolerance and clamping travel", (ref 2.2.3, Page 16)) and Grip Monitoring will be enabled, i.e. force and position will be continuously checked.

If there was no part between the gripper fingers so they can fall through the clamping travel without establishing the full force, the gripper reports that no part was found. When gripping, the gripper state is updated with the result (either HOLDING or NO PART) as well as the gripper statistics. If no part was found, the command returns an E_CMD_FAILED error, (ref 4, Page 35).

If the gripper detects a contact before reaching the part width tolerance area, this is interpreted as a collision and an E_AXIS_BLOCKED error is returned, (ref 4, Page 35). On each new call the command the gripping statistics are updated, (ref 3.3.2, Page 32).
NOTE
The clamping travel and the part width tolerance can be set using the gripper’s web interface. Please see the assembly and operating manual for a detailed description of these parameters.
NOTE

• You may reduce the grasping speed with sensitive parts to limit the impact due to the mass of the gripper fingers and the internal mechanics. Observe the instructions in the assembly and operating instructions.

• The gripper state reflects the current state of the process. You can read it using the, (☞ 2.3.4, Page 22).

• It is not possible to send a grip commands while holding a part. In general, a grip command should be always followed by a release command, (☞ 2.2.4, Page 18), before the next grip command is issued.

Syntax

GRIP()
GRIP( <float> )
GRIP( <float>, <float> )
GRIP( <float>, <float>, <float> )

Parameter

<float> Force in N (optional)
<float> Part width in mm (optional)
<float> Speed limit in mm/s (optional)

Response message

ACK GRIP to immediately acknowledge receipt of the message
FIN GRIP after successful completion of the command
2.2.4 RELEASE – Release part

Release a previously gripped part. The command’s behavior depends on the number of given parameters:

1. No parameter.
   Open the gripper fingers relative to the current position by the predefined default pull back distance relative to the current position. The default pull back distance can be set via the product’s web interface by choosing "Settings" -> "Motion Configuration" from the menu.

2. One parameter: Pull back distance.
   Open the gripper fingers by the given pull back distance relative to the current position.

3. Two parameters: Pull back distance, speed limit
   Open the gripper fingers by the given pull back distance relative to the current position using the given speed limit.

**NOTE**
Release commands are only allowed if the gripper has gripped a part before using the GRIP() command.

**Syntax**

RELEASE()

RELEASE( <float> )

RELEASE( <float>, <float> )
Basic Command Set

Parameter

\[
\begin{align*}
\text{<float>} & \quad \text{Pull back distance in mm, relative to the actual finger position (optional)} \\
\text{<float>} & \quad \text{Speed limit in mm/s (optional)}
\end{align*}
\]

Response message

\[
\begin{align*}
\text{ACK RELEASE} & \quad \text{to immediately acknowledge receipt of the message} \\
\text{FIN RELEASE} & \quad \text{after successful completion of the command}
\end{align*}
\]

2.2.5 PWT – Get or set part width tolerance

During the execution of a grip command (2.2.3, Page 15) the part width tolerance indicates the distance before reaching the nominal part width, within which a part is considered to be gripped correctly (cf. Figure "Part width tolerance and clamping travel" (2.2.3, Page 16)). If the fingers are blocked outside this distance, the grip command returns an error.

The part width tolerance can be set globally using the gripper’s web interface. The command described here can be used to override the preconfigured value, for example to dynamically adjust the part width tolerance to different parts.

NOTE

The part width tolerance is changed only for the time the connection is active and the changed value only takes effect for grip commands sent via GCL. As soon as the connection is closed, the part width tolerance will be reset to the preconfigured value.

Syntax

\[
\begin{align*}
PWT? \\
PWT=<\text{float}>
\end{align*}
\]

Parameter

\[
\begin{align*}
\text{<float>} & \quad \text{Part width tolerance in mm, referring to fingers' opening width.}
\end{align*}
\]

Response message

\[
\begin{align*}
PWT=<\text{float}>
\end{align*}
\]
2.2.6 CLT – Get or set clamping travel

During the execution of a grip command (§ 2.2.3, Page 15) the clamping travel indicates the distance the fingers are allowed to move further after touching a part to apply the desired gripping force (cf. figure "Part width tolerance and clamping travel", (§ 2.2.3, Page 16)). If the gripping force can’t be applied within this distance, the grip command returns an error. At the same time, the clamping travel indicates how far the fingers are allowed to move beyond the nominal part width to detect a part.

If a part is detected before reaching the nominal part width, the clamping travel is measured from that point. If the nominal part width is reached without detecting a part, the clamping travel is measured from the nominal part width.

The clamping travel can be set globally using the gripper’s web interface. The command described here can be used to override the preconfigured value, for example to dynamically adjust the clamping travel to different parts.

NOTE
The clamping travel is changed only for the time the connection is active and the changed value only takes effect for grip commands sent via GCL. As soon as the connection is closed, the clamping travel will be reset to the preconfigured value.

Syntax
CLT?
CLT=<float>

Parameter
<float> Clamping travel in mm, referring to fingers' opening width.

Response message
CLT=<float>
2.3 Gripper State

2.3.1 POS – Query current position of gripper jaws

Returns the current position of the gripper jaws (open width).

**Syntax**
POS?

**Response message**
POS=<float>
e.g. POS=20.0

2.3.2 SPEED – Query current speed of gripper jaws

Get current speed
Returns the current gripper jaw's speed of movement, relative to each other in mm/s

**Syntax**
SPEED?

**Response message**
SPEED=<float>
e.g. POS=142.0
2.3.3  FORCE – Query current gripping force

Returns current force value in Newton.

**Syntax**

FORCE?

**Response message**

FORCE=<float>

e.g. FORCE=23.0

2.3.4  GRIPSTATE – Query gripper state

Returns a numeral value indicating the gripper state. Overview of the available gripper states, (☞ 7, Page 42).

**Syntax**

GRIPSTATE?

**Response message**

GRIPSTATE=<integer>

e.g. GRIPSTATE=4 to indicate HOLDING.
3 Extended Command Set

The following chapter describes the extended command set of the WSG in detail.

3.1 System Commands

3.1.1 DEVTYPE – Query device type

This command returns the type of the system and can be used for example to distinguish between different devices manufactured by SCHUNK that support this protocol.

Syntax
DEVTYPE?

Response message
DEVTYPE=<string>
The command returns the system type string, e.g. DEVTYPE="WSG 32-068".

3.1.2 VERSION – Query firmware version

Returns the version number of the firmware installed on the gripper.

Syntax
VERSION?

Response message
VERSION=<string>
The command returns the firmware version string, e.g. VERSION="1.0.0".
3.1.3 **SN – Query serial number**

Returns the gripper’s serial number.

**Syntax**

SN?

**Response message**

SN=<integer>

The command returns the gripper’s serial number, e.g. 
SN=12345678.

3.1.4 **TAG – Query descriptor string (device tag)**

This command can be used to read the gripper’s descriptor string. The descriptor string, also known as device tag, can be used to identify the gripper.

**Syntax**

TAG?

**Response message**

TAG=<string>

The command returns the gripper’s descriptor string, e.g. DEV-TAG=”My Descriptor”.
3.1.5  SYSFLAGS – Query system state flags

Get system state flags. These flags describe the current state of the gripper.
Overview of available flags and their meaning, (☞ 5, Page 37).

Syntax
SYSFLAGS?
SYSFLAGS[<index>]?

Response message
SYSFLAGS=[<bool>,<bool>,...,<bool>]
SYSFLAGS[<index>]=<bool>
<bool>="1" if the corresponding system flag is set.
<bool>="0" if the corresponding system flag is not set.

If an index value is given in square brackets, the command returns only the value for the flag with the given index.

3.1.6  TEMP – Query temperature

Returns the current temperature value of the integrated temperature sensor. The temperature sensor is located on the control board inside the housing and protect the gripper from overheating

NOTE
For further information on temperature management, refer to the instructions in the assembly and operating instructions.

Syntax
TEMP?

Response message
TEMP=<float>
in degrees Celsius, e.g. TEMP=34.2
### 3.1.7 AUTOSEND – Enable or disable auto-sending of parameters

A number of values can be sent automatically by the gripper at fixed time intervals or when changing by a certain value. With the AUTOSEND command it is possible to switch this feature on or off, and to configure the behavior for each value. The following values are available:

<table>
<thead>
<tr>
<th>Designation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>POS</td>
<td>Position of the gripper fingers (open width) in mm</td>
</tr>
<tr>
<td>SPEED</td>
<td>Current speed of the gripper fingers relative to each other in mm/s</td>
</tr>
<tr>
<td>FORCE</td>
<td>Current gripping force</td>
</tr>
<tr>
<td>GRIPSTATE</td>
<td>Current gripper state</td>
</tr>
<tr>
<td>SYSFLAGS</td>
<td>Current system flags</td>
</tr>
<tr>
<td>TEMP</td>
<td>Current temperature of the controller board</td>
</tr>
</tbody>
</table>

**Syntax**

AUTOSEND( <string>, <integer> )
AUTOSEND( <string>, <integer>, <float> )
AUTOSEND( <string>, <integer>, <bool> )

**Parameter**

- `<string>` Indicates the name of the value that should be submitted automatically.
- `<integer>` Send interval in millisecond (ms). Indicates the send interval in ms with a minimum value of 10 ms. Set this value to "0" to disable auto-sending.
- `<float>` Optional delta value. For numeric values, a delta value can be set. In this case, the value will only be auto-submitted if it has changed by that value since the last submission.
- `<bool>` Automatic transmission only on change since the last transmission. For non-numeric values, this parameter specifies that a value is sent only if it has changed since the last transmission.
Response message

`ACK AUTOSEND` to immediately acknowledge command execution

The periodically submitted values all begin with an ‘@’ sign:

- `@POS=<float>` Auto submitted position value in mm
- `@FORCE=<float>` Auto submitted force value in N
- `@SPEED=<float>` Auto submitted speed value in mm/s
- `@GRIPSTATE=<integer>` Auto submitted gripper state
- `@SYSFLAGS=[<bool>,...,<bool>]` Auto submitted system flags vector
- `@TEMP=<float>` Auto submitted temperature value in °C.

Examples

`AUTOSEND("POS",10)` will send the finger position every 10 ms.

`AUTOSEND("POS",10, 2)` will send the current opening width of the gripper fingers every 10 ms, but only if the value has changed by at least 2 mm since the last transmission.
3.2 Extended Motion Control

3.2.1 STOP – Stop motion

Immediately stops any finger movement. The command sets the system state flag SF_AXIS_STOPPED and the gripper state changes to IDLE. A running motion command (e.g. MOVE, (2.2.2, Page 14)) will return ERR 19 (E_AXIS_STOPPED).

NOTE
If you would like to stop the gripper in case of an error, use the FASTSTOP command instead, (3.2.2, Page 29).

Syntax
STOP()

Parameter
No parameters

Response message
ACK STOP to immediately acknowledge command execution
3.2.2 FASTSTOP – Issue fast stop

Issue fast stop and disable drive. This command is intended to react to error conditions within the application. This function is similar to an "Emergency Stop". It immediately stops any finger movement the fastest way and prevents further motion-related commands from being executed. While a Fast Stop is active, all incoming motion commands are ignored and return the error code 16 (E_ACCESS_DENIED). This state can only be canceled by acknowledgment with the command, (☞ 3.2.3, Page 30).

The active faststop state is indicated in the system status flags and is logged in the gripper log file.

NOTE
To stop the current finger movement without raising an error condition, use the STOP command instead (☞ 3.2.1, Page 28).

Syntax
FASTSTOP()

Parameter
No parameters

Response message
ACK FASTSTOP
3.2.3 FSACK – Acknowledge Fast Stop

Acknowledge fast stop. A previously issued FAST STOP or a severe error condition must be acknowledged using this command to bring the gripper back into normal operating mode.

Syntax
FSACK(

Parameter
No parameters

Response message
ACK FSACK to immediately acknowledge command execution
3.3 Extended Gripper State

3.3.1 TARE – Tare force sensor

This command tares the force measurement if the gripper is equipped with force measurement fingers (type WSG-FMF). If no parameter is given, all connected force measuring fingers will be tared, otherwise only the finger will be tared with the given finger index (0 or 1).

Syntax

TARE()
TARE( <integer> )

Index

<integer>  Index of force measurement finger to be tared. (optional)

Response message

ACK TARE to immediately acknowledge command execution
3.3.2 GRIPSTATS – Query gripper statistics

This command returns statistics of the performed gripping operations. This includes
– Total number grips,
– Number of grips with no part detected and
– Number of grips with part lost.

Syntax
GRIPSTATS?
GRIPSTATS[ <integer> ]?

Response message
GRIPSTATS=[<integer>,<integer>,<integer>]
GRIPSTATS[<integer>]=<integer>

Returns an integer vector representing the gripper statistics. The first value indicates the total number of executed gripping commands. The second value indicates the number of gripping commands for which no gripping part has been found. The third value indicates the number of gripping commands during which the gripping part was lost.

If the command returns an index value in square brackets, only a value corresponding to the desired array index is returned.
3.4 Finger Interface

3.4.1 FDATA – Query finger data

Get finger data. Return value depends on finger type.

Syntax
FDATA?
FDATA[<integer>]?

Index
<integer> Finger-Index (optional)

Response message
FDATA=[<data>,<data>]
FDATA[<integer>]=<data>
The return value depends heavily on the finger type:
Force measuring finger (WSG-FMF) returns the measured force value.
Tactile Sensor finger (WSG-DSA) returns a vector holding measured tactile sensing values.

3.4.2 FTYPE – Query finger type

Get finger type string.

Syntax
FTYPE?
FTYPE[<integer>]?

Index
<integer> Finger-Index (optional)

Response message
FTYPE=[<string>,<string>]
FTYPE[<integer>]=<string>
### 3.4.3 FFLAGS – Query finger flags

Get finger flags. The finger condition flags describe the operating state of the connected fingers (e.g. WSG-FMF or WSG-DSA). Overview of available flags, ([6, Page 41]).

**Syntax**

- FFLAGS?
- FFLAGS[<integer>]?

**Index**

- `<integer>` Finger-Index (optional)

**Response message**

- FFLAGS=[[<bool>, ..., <bool>], [<bool>, ..., <bool>]]
- FFLAGS[<integer>]=[<bool>, ..., <bool>]
4 Appendix A: Status Codes

In case of an error, the gripper returns a numeric status code, which describes the error. The following table lists the available status codes.

<table>
<thead>
<tr>
<th>Statuscode</th>
<th>Designation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>E_SUCCESS</td>
<td>No error occurred, operation was successful</td>
</tr>
<tr>
<td>1</td>
<td>E_NOT_AVAILABLE</td>
<td>Function or data is not available</td>
</tr>
<tr>
<td>2</td>
<td>E_NO_SENSOR</td>
<td>No measurement converter is connected</td>
</tr>
<tr>
<td>3</td>
<td>E_NOT_INITIALIZED</td>
<td>Device was not initialized</td>
</tr>
<tr>
<td>4</td>
<td>E_ALREADY_RUNNING</td>
<td>The data acquisition is already running</td>
</tr>
<tr>
<td>5</td>
<td>E_FEATURE_NOT_SUPPORTED</td>
<td>The requested feature is currently not available</td>
</tr>
<tr>
<td>6</td>
<td>E_INCONSISTENT_DATA</td>
<td>One or more parameters are inconsistent</td>
</tr>
<tr>
<td>7</td>
<td>E_TIMEOUT</td>
<td>Timeout error</td>
</tr>
<tr>
<td>8</td>
<td>E_READ_ERROR</td>
<td>Error while reading data</td>
</tr>
<tr>
<td>9</td>
<td>E_WRITE_ERROR</td>
<td>Error while writing data</td>
</tr>
<tr>
<td>10</td>
<td>E_INSUFFICIENT_RESOURCES</td>
<td>No more memory available</td>
</tr>
<tr>
<td>11</td>
<td>E_CHECKSUM_ERROR</td>
<td>Checksum error</td>
</tr>
<tr>
<td>12</td>
<td>E_NO_PARAM_EXPECTED</td>
<td>A Parameter was given, but none expected</td>
</tr>
<tr>
<td>13</td>
<td>E_NOT_ENOUGH_PARAMS</td>
<td>Not enough parameters to execute the command</td>
</tr>
<tr>
<td>14</td>
<td>E_CMD_UNKNOWN</td>
<td>Unknown command</td>
</tr>
<tr>
<td>15</td>
<td>E_CMD_FORMAT_ERROR</td>
<td>Command format error</td>
</tr>
<tr>
<td>16</td>
<td>E_ACCESS_DENIED</td>
<td>Access denied</td>
</tr>
<tr>
<td>17</td>
<td>E_ALREADY_OPEN</td>
<td>Interface is already open</td>
</tr>
<tr>
<td>18</td>
<td>E_CMD_FAILED</td>
<td>Error while executing a command</td>
</tr>
<tr>
<td>19</td>
<td>E_CMD_ABORTED</td>
<td>Command execution was aborted by the user</td>
</tr>
<tr>
<td>20</td>
<td>E_INVALID_HANDLE</td>
<td>Invalid handle</td>
</tr>
<tr>
<td>21</td>
<td>E_NOT_FOUND</td>
<td>Device or file not found</td>
</tr>
<tr>
<td>22</td>
<td>E_NOT_OPEN</td>
<td>Device or file not open</td>
</tr>
<tr>
<td>23</td>
<td>E_IO_ERROR</td>
<td>Input/Output Error</td>
</tr>
<tr>
<td>24</td>
<td>E_INVALID_PARAMETER</td>
<td>Wrong parameter</td>
</tr>
<tr>
<td>25</td>
<td>E_INDEX_OUT_OF_BOUNDS</td>
<td>Index out of bounds</td>
</tr>
<tr>
<td>26</td>
<td>E_CMD_PENDING</td>
<td>The command has not yet been executed completely. A response with status code follows after execution of the command</td>
</tr>
</tbody>
</table>
## Appendix A: Status Codes

<table>
<thead>
<tr>
<th>Statuscode</th>
<th>Designation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>27</td>
<td>E_OVERRUN</td>
<td>Data overrun</td>
</tr>
<tr>
<td>28</td>
<td>E_RANGE_ERROR</td>
<td>Range error</td>
</tr>
<tr>
<td>29</td>
<td>E_AXIS_BLOCKED</td>
<td>Axis blocked</td>
</tr>
<tr>
<td>30</td>
<td>E_FILE_EXISTS</td>
<td>File already exists</td>
</tr>
</tbody>
</table>
## Appendix B: System State Flags

The WSG provides a total of 32 system state flags that can be read using the SYSFLAGS query (§ 3.1.5, Page 25). The following table lists the available flags and explains their meaning.

<table>
<thead>
<tr>
<th>Index</th>
<th>Designation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31..21</td>
<td>reserved</td>
<td>These flags are currently not used.</td>
</tr>
<tr>
<td>20</td>
<td>SF_SCRIPT_FAILURE</td>
<td><strong>Script Error</strong>&lt;br&gt;An error occurred while executing a script and the script has been aborted. The flag is reset whenever a script is started.</td>
</tr>
<tr>
<td>19</td>
<td>SF_SCRIPT_RUNNING</td>
<td><strong>A script is currently running</strong>&lt;br&gt;The flag is reset if the script either terminated normally, a script error occurred or the script has been terminated manually by the user.</td>
</tr>
<tr>
<td>18</td>
<td>SF_CMD_FAILURE</td>
<td><strong>Command Error</strong>&lt;br&gt;The last command returned an error.</td>
</tr>
<tr>
<td>17</td>
<td>SF_FINGER_FAULT</td>
<td><strong>Finger Fault</strong>&lt;br&gt;The status of at least one finger is different from &quot;operating&quot; and &quot;not connected&quot;. Please check the finger flags for a more detailed error description.</td>
</tr>
<tr>
<td>16</td>
<td>SF_CURR_FAULT</td>
<td><strong>Engine Current Error</strong>&lt;br&gt;The flag is set if the engine has reached its maximum thermal power consumption. The flag will be reset automatically as soon as the engine has recovered. Then the corresponding Fast Stop can be committed.</td>
</tr>
</tbody>
</table>
| 15    | SF_POWER_FAULT            | **Power Error**<br>The flag is set if the power supply is out-
## Appendix B: System State Flags

<table>
<thead>
<tr>
<th>Index</th>
<th>Designation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Side the valid range. The power supply has to be checked and adapted if necessary.</td>
</tr>
<tr>
<td>14</td>
<td>SF_TEMP_FAULT</td>
<td><strong>Temperature Error</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The flag is set if the gripper hardware has reached a critical temperature. All motion-related commands are disabled until the temperature falls below the critical level.</td>
</tr>
<tr>
<td>13</td>
<td>SF_TEMP_WARNING</td>
<td><strong>Temperature Warning</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The flag is set if the gripper hardware will soon reach a critical temperature level.</td>
</tr>
<tr>
<td>D12</td>
<td>SF_FAST_STOP</td>
<td><strong>Fast Stop</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The flag is set if the gripper has been stopped due to an error condition. You have to acknowledge the error in order to reset this flag and to re-enable motion-related commands.</td>
</tr>
<tr>
<td>11..10</td>
<td>reserved</td>
<td>These flags are currently not used.</td>
</tr>
<tr>
<td>9</td>
<td>SF_FORCECNTL_MODE</td>
<td><strong>Force Control Mode</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The flag is set if the true force control is currently enabled by using the installed force measurement finger (WSG-FMF). If this flag is not set, the grasping force is controlled by approximation based on the motor current.</td>
</tr>
<tr>
<td>8</td>
<td>SF_OVERDRIVE_MODE</td>
<td><strong>Overdrive Modus</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The flag is set if the gripper is in overdrive mode. The grasping force can be set to a value up to the overdrive force limit. If this bit is not set, the grasping</td>
</tr>
</tbody>
</table>
### Appendix B: System State Flags

<table>
<thead>
<tr>
<th>Index</th>
<th>Designation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>01.00</td>
<td>WSG-GCL</td>
<td>en</td>
</tr>
<tr>
<td>7</td>
<td>SF_TARGET_POS_REACHED</td>
<td><strong>Target position reached</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The flag is set if the target position was reached. This flag is not synchronized with SF_MOVING, so it is possible that there is a delay between SF_MOVING being reset and SF_TARGET_POS becoming active.</td>
</tr>
<tr>
<td>6</td>
<td>SF_AXIS_STOPPED</td>
<td><strong>Axis stopped</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The flag is set if a previous motion command has been aborted using the stop command. This flag is reset on the next motion command.</td>
</tr>
<tr>
<td>5</td>
<td>SF_SOFT_LIMIT_PLUS</td>
<td><strong>Positive direction soft limit reached</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The flag is set if the fingers reached the defined soft limit in positive moving direction. A further movement into this direction is not allowed any more. This flag is cleared if the fingers are moved away from the soft limit position.</td>
</tr>
<tr>
<td>4</td>
<td>SF_SOFT_LIMIT_MINUS</td>
<td><strong>Negative direction soft limit reached</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The flag is set if the fingers reached the defined soft limit in negative moving direction. A further movement into this direction is not allowed any more. This flag is cleared if the fingers are moved away from the soft limit position.</td>
</tr>
</tbody>
</table>
## Appendix B: System State Flags

<table>
<thead>
<tr>
<th>Index</th>
<th>Designation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>SF_BLOCKED_PLUS</td>
<td><strong>Axis is blocked in positive moving direction</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The flag is set if the axis is blocked in positive moving direction. The flag will be reset if either the blocking condition is resolved or a stop command is issued.</td>
</tr>
<tr>
<td>2</td>
<td>SF_BLOCKED_MINUS</td>
<td><strong>Axis is blocked in negative moving direction</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The flag is set if the axis is blocked in negative moving direction. The flag will be reset if either the blocking condition is resolved or a stop command is issued.</td>
</tr>
<tr>
<td>1</td>
<td>SF_MOVING</td>
<td><strong>Fingers are currently moving</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The flag is set if whenever a movement is started (e.g. MOVE command) and reset automatically as soon as the movement stops.</td>
</tr>
<tr>
<td>0</td>
<td>SF REFERENCED</td>
<td><strong>Fingers Referenced</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The flag is set if the gripper is referenced and accepts motion-related commands.</td>
</tr>
</tbody>
</table>
### Appendix C: Finger Flags

<table>
<thead>
<tr>
<th>Index</th>
<th>Designation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>15..11</td>
<td>reserved</td>
<td>These flags are currently not used.</td>
</tr>
<tr>
<td>10</td>
<td>FF_INIT_FAULT</td>
<td><strong>Finger initialization error</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>An error occurred during finger initialization.</td>
</tr>
<tr>
<td>9</td>
<td>FF_COMM_FAULT</td>
<td><strong>Communication fault</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>A Communication fault occurred during runtime.</td>
</tr>
<tr>
<td>8</td>
<td>FF_POWER_FAULT</td>
<td><strong>Power fault</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Over-Current fault detected.</td>
</tr>
<tr>
<td>7..3</td>
<td>reserved</td>
<td>These flags are currently not used.</td>
</tr>
<tr>
<td>2</td>
<td>FF_COMM_OPEN</td>
<td><strong>Communication open</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Finger communication interface is open.</td>
</tr>
<tr>
<td>1</td>
<td>FF_CONFIG_AVAIL</td>
<td><strong>Finger configuration available</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>A finger configuration descriptor could be read from the finger’s memory.</td>
</tr>
<tr>
<td>0</td>
<td>FF_POWER_ON</td>
<td><strong>Finger powered</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Finger is powered on.</td>
</tr>
</tbody>
</table>
## Appendix D: Gripper States

Stop command: Issuing a Stop Command in any state (except ERROR) will abort the current action and immediately return to IDLE state.

<table>
<thead>
<tr>
<th>Zustand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDLE</td>
<td>The gripper is in its idle state</td>
</tr>
<tr>
<td></td>
<td>No gripping process is currently in progress.</td>
</tr>
<tr>
<td>GRIPPING</td>
<td>The gripper is closing</td>
</tr>
<tr>
<td></td>
<td>A part is being gripped, the fingers are moving towards the part.</td>
</tr>
<tr>
<td></td>
<td>The detection of gripped items is active.</td>
</tr>
<tr>
<td>HOLDING</td>
<td>A part is being held</td>
</tr>
<tr>
<td></td>
<td>A part is being held with the configured force.</td>
</tr>
<tr>
<td></td>
<td>The monitor of the gripped part has been activated.</td>
</tr>
<tr>
<td>PART LOST</td>
<td>A part has been lost</td>
</tr>
<tr>
<td></td>
<td>The pre-configured gripping force can no longer be applied, presumably because the part to be gripped has been lost.</td>
</tr>
<tr>
<td>NO PART FOUND</td>
<td>No part has been lost</td>
</tr>
<tr>
<td></td>
<td>When closing the gripper at the configured position, no part was found to be gripped.</td>
</tr>
<tr>
<td>Zustand</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>RELEASING</td>
<td><strong>The gripper is opening</strong></td>
</tr>
<tr>
<td></td>
<td>The gripped part is being released, the fingers are moving.</td>
</tr>
<tr>
<td>POSITIONING</td>
<td><strong>Pre-positioning</strong></td>
</tr>
<tr>
<td></td>
<td>The grippers are being pre-positioned and the fingers are moving.</td>
</tr>
<tr>
<td>ERROR</td>
<td><strong>Error</strong></td>
</tr>
<tr>
<td></td>
<td>An error occurred carrying out the last command. For errors which require</td>
</tr>
<tr>
<td></td>
<td>acknowledgment, the SF_FAST_STOP flag is also set in the system status word.</td>
</tr>
<tr>
<td></td>
<td>After any acknowledgment required, the movement can be restarted.</td>
</tr>
</tbody>
</table>
8 Appendix E: Data Types

The following data types are used throughout this document:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;integer&gt;</td>
<td>An integer value</td>
</tr>
<tr>
<td>&lt;bool&gt;</td>
<td>An integer value which can be &quot;0&quot; or &quot;1&quot;</td>
</tr>
<tr>
<td>&lt;float&gt;</td>
<td>A floating point value</td>
</tr>
<tr>
<td>&lt;string&gt;</td>
<td>A string literal, always enclosed into quotes</td>
</tr>
<tr>
<td>&lt;vector&gt;</td>
<td>A vector of multiple values of the same type</td>
</tr>
</tbody>
</table>