



GRILINK-PLUGIN FOR STÄUBLI ROBOT

Version 2.0.0



Content

1	Introduction	2
1.1	Notation and symbols	2
1.2	Intended use.....	2
1.3	System requirements	3
1.4	License terms.....	3
2	Installing the GRIPLINK plugin	4
2.1	Create SIO socket connection for the GRIPLINK.....	4
2.2	Behavior in case of error.....	5
3	Command Reference.....	6
3.1	Establish connection – connect().....	7
3.2	Activate device - enable()	8
3.3	Query device state - devstate()	9
3.4	Simultaneous polling of device states - mdevstate()	10
3.5	Disable device - disable()	11
3.6	Reference gripping module - home()	12
3.7	Grip workpiece - grip().....	13
3.8	Simultaneous gripping of workpieces - mgrip().....	14
3.9	Release workpiece - release().....	15
3.10	Simultaneous release of workpieces - mrelease().....	16
3.11	Control gripping force hold - hold()	17
3.12	Control of the LED display - LED()	18
3.13	Query position and sensor values - value().....	19
	Appendix A. Device state.....	20

1 Introduction

With GRIPLINK technology, IO-Link compatible automation components can be connected to robot systems from leading manufacturers via a simple network connection. The GRIPLINK plug-in for STÄUBLI is the control-side link and enables an easy integration of GRIPLINK technology from WEISS ROBOTICS into robot systems from the manufacturer STÄUBLI.



These instructions describe the functions of the GRIPLINK plug-in. For information on mounting, commissioning and operation of the GRIPLINK controller, refer to the operating instructions of the respective module. These can be found online at www.griplink.de/manuals

1.1 Notation and symbols

For a better overview, the following symbols are used in these instructions:



Function or safety-relevant note. Non-observance may endanger the safety of personnel and plant, damage the device or impair the function of the device.



Additional information for a better understanding of the described facts.



Reference to further information.

1.2 Intended use

The "GRIPLINK Plugin" software is intended for communication between the GRIPLINK Controller from WEISS ROBOTICS and a robot controller. The requirements of the applicable directives and the installation and operating instructions in these instructions must be observed and complied with. Any other use or use beyond the scope of this manual is considered improper use. The manufacturer is not liable for any damage resulting from this.

1.3 System requirements

One of the following STÄUBLI robot controllers is required for operation:

- CS9 (s8.10.6)

The following robot options are required to run the software:

- Stäubli SRS



Contact your STÄUBLI dealer to obtain the robot options.



The IP address of the GRIPLINK controller must be in the same subnet as that of the robot controller. The GRIPLINK controller manual describes the exact procedure for changing the IP address.

1.4 License terms

The GRIPLINK plugin is protected by copyright. The respective valid license terms are included in the software package. With the installation you accept these license terms.

2 Installing the GRIPLINK plugin

To operate the GRIPLINK controller, the GRIPLINK plugin is required on the robot controller. To install the GRIPLINK plugin, follow points 1 to 4.



Make sure that you are using the latest version of the GRIPLINK plugin. The current version can be downloaded from www.griplink.de/plugins.

1. Unpack the previously downloaded ZIP archive with the GRIPLINK plugin into the SRS directory of the cell you are using: Staubli > SRS > CellName > ControllerName > usr > usrapp.
2. Open the STAUBLI Robotics Suite:
 - Open the cell
 - Go to the Cell Explorer
 - Select the Val3 application that should use the GRIPLINK plugin, then "right click" > Add > New Library > ... > GRIPLINK > GRIPLINK.pjx > Open > Specify Alias > Load Automatically > Ok



In this manual GRIPLINK is used as alias for all examples. Commands are called via "call Alias:Command()".

2.1 Create SIO socket connection for the GRIPLINK

In STAUBLI Robotics Suite, under the Home > I/O Management tab, in the new window, click on Sockets > "Right Click" > Edit I/O Map > + > Tcp client > Port. Enter the IP address of the GRIPLINK controller as "Server IP" and the timeout (see Figure 1 for default values).



The IP address of the GRIPLINK controller can be changed via the GRIPLINK-ET4 website. By default, the port 10001 and the IP address 192.168.1.40 are set.



If the Val3 applications from the STAUBLI Robotics Suite are loaded onto the controller, the SIO variable may have to be linked again via the Application Manager. To do this, create an SIO socket connection via the teach pendant: I/O > Sockets > TCP Client > + > Port and specify the server IP and the timeout.

Now link the global SIO variable in the program via: ≡ > Application Manager > Val3 Application > Hard Disk > Program > Global Variables > sio > SIO Variable > Link (F2) > Edit (F6) > Socket > TCP Clients > Socket Name > Ok (F8) > Ok (F8) > Save (F8)

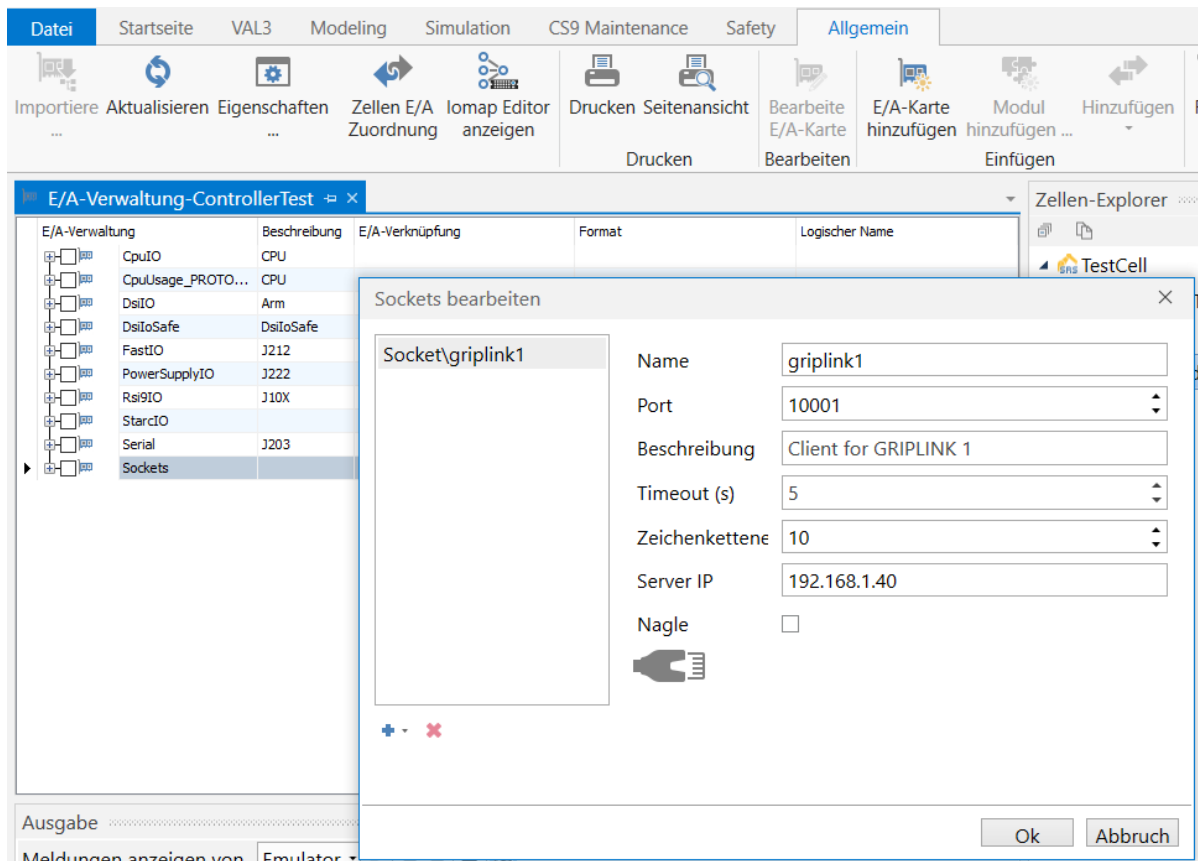


Figure 1: Socket settings

2.2 Behavior in case of error

If an error occurs within the GRIPLINK plug-in or during communication with the GRIPLINK controller, an error message is always written to the logger. As a rule, the running movements of the robot are also aborted. The same applies if the addressed device is in the FAULT state or changes there due to a command.

3 Command Reference

The GRIPLINK plugin provides the user with a collection of generic movement and gripping functions as well as functions for reading connected sensors. The commands are implemented as Val3 programs, which receive their parameters as transfer values. The return values of the commands are stored in the passed reference variables. To execute a command, the corresponding Val3 program must be called. Some commands are available as single as well as multiple commands.

The basic program flow with the GRIPLINK plugin is always as follows:

1. Establish connection with GRIPLINK:connect()
2. Activate gripping module or sensor with GRIPLINK:enable()
3. For servo gripping modules without absolute encoder: Reference gripping module with GRIPLINK:home()
4. Grasp/release with GRIPLINK:grip() or GRIPLINK:release()
5. Query position or measured value with GRIPLINK:value()

The available commands of the GRIPLINK plug-in are described below.

3.1 Establish connection – connect()

This command establishes the connection between GRIPLINK plugin and an Ethernet socket port. The reference of the GRIPLINK Ethernet socket connection is passed as a sio type as a transfer parameter. The command stores the port in the library so that following commands can access it. If the GRIPLINK cannot be reached via the specified port, a log message is written and the program execution is stopped.

If GRIPLINK-ET4 commands are executed before a GRIPLINK:connect(), a log message is written and program execution is halted.

Syntax

call GRIPLINK:connect(< sio&x_sioSocket >)

Parameter

< sio&x_sioSocket > Reference of the GRIPLINK Ethernet Socket connector.

Example

Establish connection between GRIPLINK-plugin and the GRIPLINK Ethernet Socket port via the variable "sioGRIPLINK":

```
call GRIPLINK:connect(sioGRIPLINK)
```


3.2 Activate device - enable()

This command enables the device connected to the specified device port. GRIPLINK:enable() must be executed after GRIPLINK:connect() for all devices. If GRIPLINK commands are executed before a GRIPLINK:enable(), this triggers a log message and program execution is halted.

Syntax

```
call GRIPLINK:enable(<num x_nPort> , < num&x_nDevstate > )
```

Parameter

< x_nPort > Index of the device port (0 to 3)

< num&x_nDevstate > Reference for the return value

Return value

x_nDevstate Current device state (see Appendix A)

Example

Activate drive and gripping module at port 0:

```
call GRIPLINK:enable(0,x_nDevstate)
```

3.3 Query device state - devstate()

This command returns the state of the selected device.

Syntax

```
call GRIPLINK:devstate(<num x_nPort>,< num&x_nDevstate >)
```

Parameter

< num x_nPort > Index of the device port (0 to 3)

< num&x_nDevstate > Reference for the return value

Return value

x_nDevstate Current device state (see Appendix A)

Example

Wait until the device state of the gripping module at port 2 is "HOLDING" (4):

```
while x_nDevstate!=4
  call GRIPLINK:devstate(2,x_nDevstate)
  delay(0.01)
endWhile
```

3.4 Simultaneous polling of device states - mdevstate()

This command returns the states of the selected gripping modules.

Syntax

```
call GRIPLINK:mdevstate(< num x_nDevstate[1] >)
```

Parameter

< num&x_nDevstate[1] > Reference array for the return values

Return values

x_nDevstate[0]	Current device state of gripping module at port 0 (see Appendix A)
x_nDevstate[1]	Current device state of gripping module at port 1 (see Appendix A)
x_nDevstate[2]	Current device state of gripping module at port 2 (see Appendix A)
x_nDevstate[3]	Current device state of gripping module at port 3 (see Appendix A)

Examples

Wait until the device state of the gripping modules at port 1 and 2 is "HOLDING" (4):

```
while x_nDevstate[1]!=4 or x_nDevstate[2]!=4 call  
GRIPLINK:mrelease(x_nDevstates)  
    delay(0.01)  
endWhile
```

3.5 Disable device - disable()

Deactivates the device connected to the selected device port. This command can be used, for example, to change tools. The device can be reactivated via GRIPLINK:enable().

Syntax

call GRIPLINK:disable(< num x_nPort >, < num&x_nDevstate >)

Parameter

< num x_nPort > Index of the device port (0 to 3)

< num&x_nDevstate > Reference for the return value

Return value

x_nDevstate Current device state (see Appendix A)

Example

Change the gripping module at port 1:

```
//Connect to GRIPLINK
  call GRIPLINK:connect(sioGRIPLINK)
//Activate gripper 1
  call GRIPLINK:enable(1,x_nDevstate)
//DO something
//...
//Prepare tool change:
//Disable gripper 1:
  call GRIPLINK:disable(1,x_nDevstate)
//Operate the tool changer here
//Activate new gripper:
  call GRIPLINK:enable(1,x_nDevstate)
```

3.6 Reference gripping module - home()

References the selected servo gripper. The function executes a reference run of the gripping module and waits until this is completed. After the GRIPLINK:home() command has been executed, the fingers of the gripping module are powerless and must be moved to a defined position with a GRIPLINK:grip() or GRIPLINK:mgrip() or GRIPLINK:release() or GRIPLINK:mrelease().



The reference run can be configured via the web interface of the GRIPLINK controller.

Syntax

call GRIPLINK:home(<num_x_nPort>)

Parameter

< num_x_nPort > Index of the device port (0 to 3)

Example

Reference the gripping module to port 2:

```
call GRIPLINK:home(2)
```

3.7 Grip workpiece - grip()

Grips a workpiece with the selected gripping module and the selected handle. The command waits until the device state changes to either "HOLDING" or "NO PART".



The gripping parameters can be configured via the web interface of the GRIPLINK interface converter.

Syntax

```
call GRIPLINK:grip(< num x_nPort > , < num x_nIndex > , < num& x_nDevstate > )
```

Parameter

< num x_nPort >	Index of the device port (0 to 3)
< num x_nIndex >	Selected handle (range depends on gripper model)
< num& x_nDevstate >	Reference for the return value

Return value

x_nDevstate	Current device state (see Appendix A)
-------------	---------------------------------------

Examples

Gripping module at port 0 is to grip with grip 2. If no workpiece was found, the gripper should open again and try again:

```
while bEnd==false
  call GRIPLINK:grip(0,2,x_nDevstate)
  if x_nDevstate!=5
    //No part, open and try again:
    call GRIPLINK:release(0,2,x_nDevstate)
  else
    //Part gripped!
    bEnd=true
  endif
endwhile
```

3.8 Simultaneous gripping of workpieces - mgrip()

This command executes a grip with the selected gripping modules. The command waits until all gripping modules have each reached one of the states "HOLDING" or "NO PART".



The gripping parameters can be configured via the web interface of the GRIPLINK interface converter.

Syntax

call GRIPLINK:mgrip(< num&x_nPorts[1] >, < num x_nIndex >, < num& x_nDevstate[1] >)

Parameter

< num&x_nPorts[1] >	Selected gripping modules as num vector: x_nPorts[0]: 1 = gripping module selected at port 0, 0 = not selected x_nPorts[1]: 1 = gripping module selected at port 1, 0 = not selected x_nPorts[2]: 1 = gripping module selected at port 2, 0 = not selected x_nPorts[3]: 1 = gripping module selected at port 3, 0 = not selected
< num x_nIndex >	Selected handle (range depends on gripper model)
< num&x_nDevstate[1] >	Reference array for the return values

Return values

x_nDevstate[0]	Current device state of gripping module at port 0 (see Appendix A)
x_nDevstate[1]	Current device state of gripping module at port 1 (see Appendix A)
x_nDevstate[2]	Current device state of gripping module at port 2 (see Appendix A)
x_nDevstate[3]	Current device state of gripping module at port 3 (see Appendix A)

Examples

Gripper modules at port 1 and 2 grip workpiece with handle 2:

```
L_nPorts[1]=1  
L_nPorts[2]=1  
call GRIPLINK:mgrip(L_nPorts,2,x_nDevstates)
```

3.9 Release workpiece - release()

Releases the workpiece gripped with the selected gripping module. The command waits until the workpiece has been released.



The gripping parameters can be configured via the web interface of the GRIPLINK interface converter.

Syntax

```
call GRIPLINK:release(< num x_nPort >, < num x_nIndex >, < num& x_nDevstate > )
```

Parameter

< num x_nPort >	Index of the device port (0 to 3)
< num x_nIndex >	Selected handle (range depends on gripper model)
< num& x_nDevstate >	Reference for the return value

Return value

x_nDevstate	Current device state (see Appendix A)
-------------	---------------------------------------

Examples

Release the workpiece gripped with the gripping module at port 0 and handle 2:

```
call GRIPLINK:release(0,2,x_nDevstate)
```


3.10 Simultaneous release of workpieces - mrelease()

Simultaneously releases the workpiece gripped with the selected gripping modules. The command waits until all gripping modules have each reached the "RELEASED" state.

Syntax

```
call GRIPLINK:mrelease(< num x_nPorts[1] >, < num x_nIndex >, < num x_nDevstate[1] > )
```

Parameter

< num&x_nPorts[1] >	Selected gripping modules as num vector: x_nPorts[0]: 1= gripping module selected at port 0, 0 = not selected x_nPorts[1]: 1= gripping module selected at port 1, 0 = not selected x_nPorts[2]: 1= gripping module selected at port 2, 0 = not selected x_nPorts[3]: 1= gripping module selected at port 3, 0 = not selected
< num x_nIndex >	Selected handle (range depends on gripper model)
< num&x_nDevstate[1] >	Reference array for the return values

Return values

x_nDevstate[0]	Current device state of gripping module at port 0 (see Appendix A)
x_nDevstate[1]	Current device state of gripping module at port 1 (see Appendix A)
x_nDevstate[2]	Current device state of gripping module at port 2 (see Appendix A)
x_nDevstate[3]	Current device state of gripping module at port 3 (see Appendix A)

Examples

Gripping module at port 0,1 and 2 release workpiece with handle 3:

```
L_nPorts[0]=1  
L_nPorts[1]=1  
L_nPorts[2]=1  
call GRIPLINK:mrelease(L_nPorts,3,x_nDevstates)
```

3.11 Control gripping force hold - hold()

The innovative gripping force safety device developed by WEISS ROBOTICS maintains the gripping force on the workpiece even if the power supply to the gripping module is unexpectedly interrupted. Thanks to the integrated absolute sensor technology, production can continue immediately when the power supply is restored, even without referencing. Furthermore, hold() enables permanent gripping without the gripping module becoming hot.

This command activates or deactivates the gripping force safety device for the selected gripping module.



HOLD is not available with all gripping modules.

Syntax

```
call GRIPLINK:hold(<numx_nPort>,< numx_nEnable >)
```

Parameter

< numx_nPort > Index of the device port (0 to 3)

< numx_nEnable > Gripping force retention: 1 = on, 0 = off

Example

Activates gripping force retention for gripping module at port 2 if the gripping status is "HOLDING":

```
call GRIPLINK:grip(2,0,x_nDevstate)
if x_nDevstate=5
    call GRIPLINK:hold(2,1)
endif
```

3.12 Control of the LED display - LED()

This command changes the color and pattern of the light ring of a selected gripping module. This function is only available for gripping modules of the CRG series from WEISS ROBOTICS.



Light patterns can be configured via the web interface of the GRIPLINK controller.

Syntax

```
call GRIPLINK:LED(< num x_nPort >, < num x_nIndex >)
```

Parameter

< num x_nPort > Index of the device port (0 to 3)

< num x_nIndex > Index of the predefined light pattern

Example

Grasp with the gripping module at port 3 and change the color of the light ring to light pattern 0, if the finger position (Value Index 0) is greater than or equal to 8.1 mm afterwards and to light pattern 1, if smaller:

```
call GRIPLINK:grip(3,0,x_nDevstate)
call GRIPLINK:value(3,0,x_nValue)
if x_nValue>=8100
    call GRIPLINK:LED(3,0)
else
    call GRIPLINK:LED(3,1)
endif
```

3.13 Query position and sensor values - value()

Reads a measured value from the connected device. Depending on the device, one or more measured values are available, which can be selected via the index to be specified.



For gripping modules and similar actuators, index 0 always corresponds to the finger position/gripping distance in micrometers.

Syntax

```
call GRIPLINK:value(< num x_nPort>, < num x_nIndex>, < num x_nValue >)
```

Parameter

< num x_nPort >	Index of the device port (0 to 3)
< num x_nIndex >	Index of the measured value to be queried
< num x_nValue >	Reference for the return of the measured value

Return value

x_nValue	Returns the measured value
----------	----------------------------

Example 1

Query the distance to the workpiece (measured value index 0) of the laser distance sensor connected to port 1 and grip with the gripper at port 3 if the distance is smaller than 42 mm. Otherwise, call the "skip" subprogram:

```
call GRIPLINK:value(1,0,x_nValue)
if x_nValue<42000
  call GRIPLINK:GRIP(3,0,x_nDevstate)
else
  call skip()
endif
```

Example 2

Execute the subroutine "prog" if the finger position of the gripping module at port 2 is greater than 10.5 mm:

```
call GRIPLINK:value(2,0,x_nValue)
if x_nValue>10500
  call prog()
endif
```

Appendix A. Device state

Device status	Code	Description
NOT CONNECTED	0	No device connected
NOT INITIALIZED	1	Not initialized
DISABLED	2	Ready for operation, but not activated
RELEASED	3	Workpiece released
NO PART	4	No workpiece found
HOLDING	5	Workpiece is held
OPERATING	6	Ready for operation
FAULT	7	Error condition

© 2020 WEISS ROBOTICS GmbH & Co. KG. All rights reserved.

GRIPLINK and PERMAGRIP are registered trademarks of WEISS ROBOTICS GmbH & Co. KG. All other trademarks are the property of their respective owners.

Specifications given in this document are subject to change without notice for the purpose of product improvement. Trademarks are the property of their respective owners. Our products are not intended for use in life support systems or systems where failure could result in personal injury.

