WEISS ROBOTICS

GRIPLINK PLUG-IN FOR ABB

Version 2.0.0

Juli 2026



Contents

1 Introduction 3

1.1 Notation and Symbols 3

1.2 Intended Use 3

1.3 System Requirements 4

1.4 License Terms 4

1.5 Demo Programs 4

2 Installation 5

2.1 Plug-In Installation 5

2.2 Network Settings 12

2.3 Firewall Settings 13

3 Content of the Plug-In 14

3.1 RAPID 14

3.2 Wizard 15

3.3 Global Variables for Return Values 16

3.4 Device States 17

3.5 Error Handling 18

3.6 General Program Sequence 19

3.7 Demo Programs 20

4 Command Reference 21

4.1 GRIPLINK_Connect: Establish Connection 22

4.2 GRIPLINK_Disconnect: Close Connection 23

4.3 GRIPLINK_AssertDevice: Check Connected Device 24

4.4 GRIPLINK_DeviceTag: Query Device Tag 25

4.5 GRIPLINK_SetDeviceTag: Set Device Tag 26

4.6 GRIPLINK_Enable: Activate Device 27

4.7 GRIPLINK_Disable: Deactivate Device 28

4.8 GRIPLINK_Home: Reference Device 29

4.9 GRIPLINK_Grip: Grip Workpiece 30

4.10 GRIPLINK_Release: Release Workpiece 32

4.11 GRIPLINK_FlexGrip: Grip Workpiece with Flexible Parameters 34

4.12 GRIPLINK_FlexRelease: Release Workpiece with Flexible Parameters 36

4.13 GRIPLINK_LED: Select LED Visualization 38
4.14 GRIPLINK_Clamp: Control Gripping Force Retention 39
4.15 GRIPLINK_WaitForStateTransition: Wait for State Transition 40
4.16 GRIPLINK_SetValue: Set Device Value 42
4.17 GRIPLINK_WaitValue: Wait for Device Value 43
4.18 GRIPLINK_DeviceState: Query Device State 45
4.19 GRIPLINK_Value: Query Device Value 47
4.20 GRIPLINK_SetGripConfig: Configure Grip Parameters 49

1 Introduction

With GRIPLINK technology, IO-Link-compatible automation components can be connected to robot systems from leading manufacturers via a TCP/IP network connection. The GRIPLINK plug-in for ABB is the Software link between GRIPLINK controller and robot controller and enables the simple integration of GRIPLINK technology from WEISS ROBOTICS into robot systems by manufacturer ABB.



These instructions describe the content and functions of the GRIPLINK plug-in for robots from ABB.
Information on installation, commissioning, and operation of the GRIPLINK hardware can be found in the respective operating manual at **https://weiss-robotics.com.**

1.1 Notation and Symbols

The following symbols are used in these instructions to provide a better overview:



Functional or safety-relevant information.
Non-compliance may endanger the safety of personnel and the system, damage the device or impair its function.



Additional information for a better understanding of the facts described.



Reference to further information.



Specific information on gripper modules by WEISS ROBOTICS.

The GRIPLINK plug-in is compatible with all devices that support the GRIPLINK protocol via TCP/IP. This includes the GRIPLINK-ET4, as well asd all gripping modules of the WPG series. For better readability, these devices are summarized under the term *GRIPLINK controller.*

1.2 Intended Use

The software *GRIPLINK plug-in* is intended for communication between a GRIPLINK controller by WEISS ROBOTICS and a robot controller. The requirements of the applicable directives and the installation and operating manual in this manual must be observed and complied with. Any other or additional use is considered improper use. The manufacturer is not liable for any resulting damage.

1.3 System Requirements

The GRIPLINK plug-in is compatible with the GRIPLINK-ET4 (firmware version 5.1.0 or newer) and WPG gripping modules (firmware version 2.2.0. or newer).

An ABB robot controller with ABB RobotWare (version 7.1 or newer) is required for operation. The ABB RobotStudio programming environment® (version 2022.3 or newer) is also recommended for easier setup and programming.



Contact ABB or your ABB distributor to purchase these products.

1.4 License Terms

The GRIPLINK plug-in is protected by copyright. The applicable license terms are included in the software package. With installation, you accept these license terms.

1.5 Demo Programs

The demo programs included in the software package are examples of how to use the plug-in. They are intended for demonstration and testing purposes only!

2 Installation

To be able to use the GRIPLINK plug-in, the RobotWare add-in must be installed on the robot controller and configured correctly. The necessary steps are explained in the following chapter.

Detailed information on mounting the gripping modules and on connecting the GRIPLINK controller can be found in the corresponding operating manual.

2.1 Plug-In Installation

The GRIPLINK plug-in can be installed on the robot controller either via a PC with ABB RobotStudio®, or directly on the FlexPendant using a USB stick.

General information on the installation of RobotWare Add-Ins can be found in the corresponding ABB documentation.

2.1.1 Installation via ABB RobotStudio®

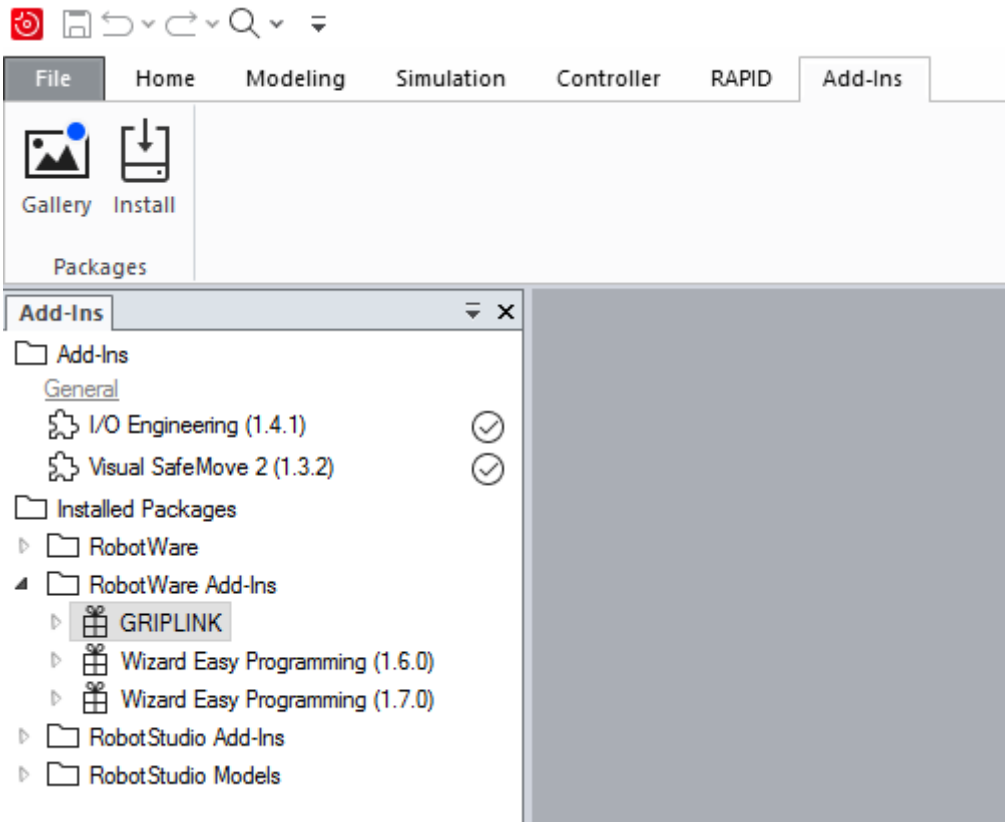
To install the GRIPLINK plug-in using RobotStudio, a PC is required on which the ABB RobotStudio® programming and simulation environment is installed and which is connected to the robot controller. Carry out the following steps:

1. Download the latest version of the GRIPLINK plug-in for ABB from **www.griplink.de** and unzip the ZIP file.
2. Start RobotStudio and the robot controller.
3. Select the *Add-Ins* tab and click on *Install*.



4. Navigate to the location of the downloaded GRIPLINK plug-in. Select the RobotStudio package, which is the file with the extension *.rspak*.

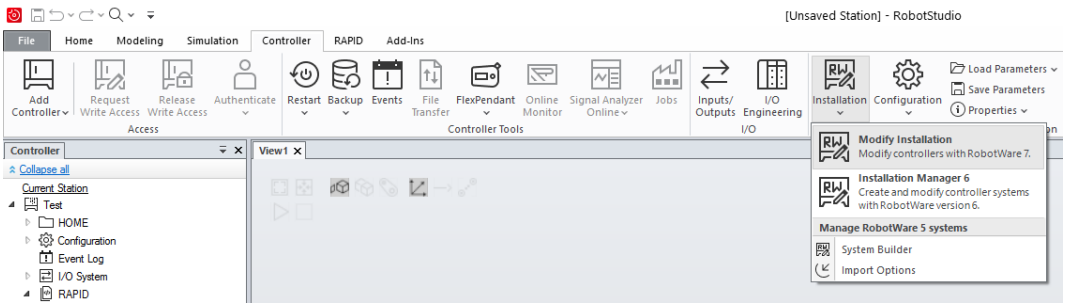
5. After installation in RobotStudio, the GRIPLINK plug-in appears in the overview of available add-ins on the left-hand side of the window.



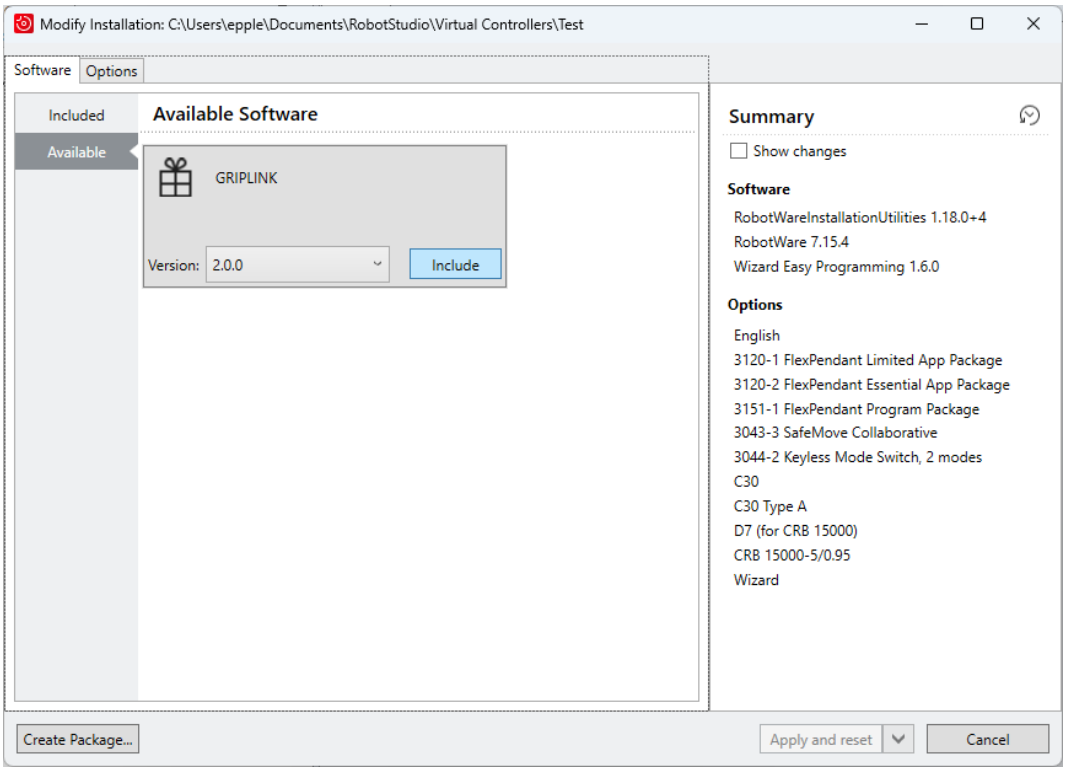
6. Open or create a project using the real or virtual robot controller on which the GRIPLINK plug-in is to be installed. You can also establish a connection to the corresponding controller directly.

Details on the operation of ABB RobotStudio® can be found in the corresponding ABB documentation.

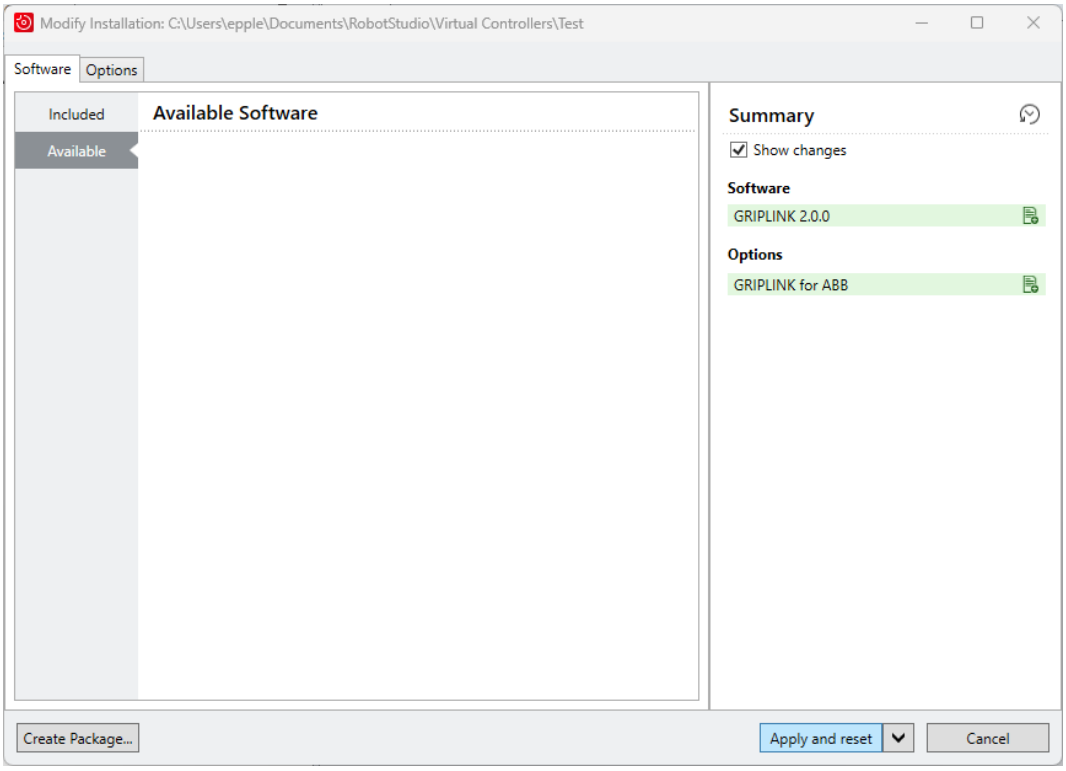
7. Select the *Controller* tab and click on *Installation* and *Modify Installation*.



8. Select the *Software* tab. The GRIPLINK plug-in is listed under *Available*. Select a version if necessary and click on *Include*.



9. The GRIPLINK plug-in is highlighted in green under *Summary* on the right-hand side of the window. Click on *Apply and Reset* and confirm the restart in the pop-up window.

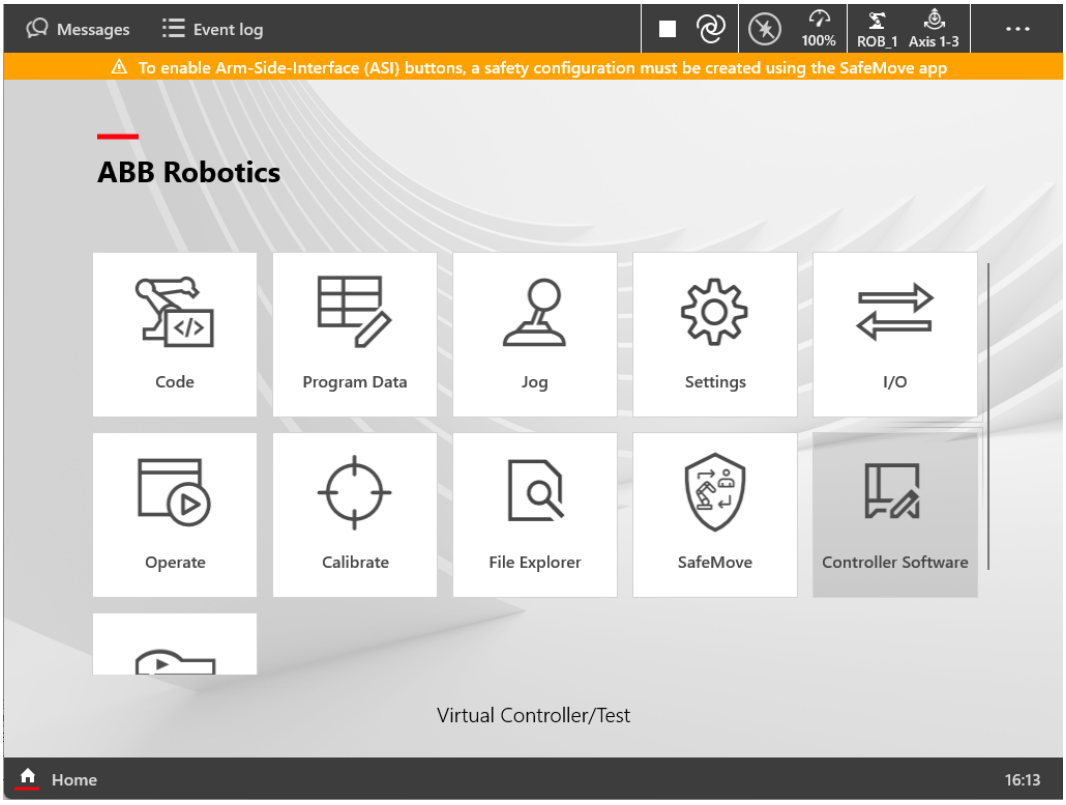


After the robot controller restart, the GRIPLINK plug-in is fully installed, and the corresponding functionality is available.

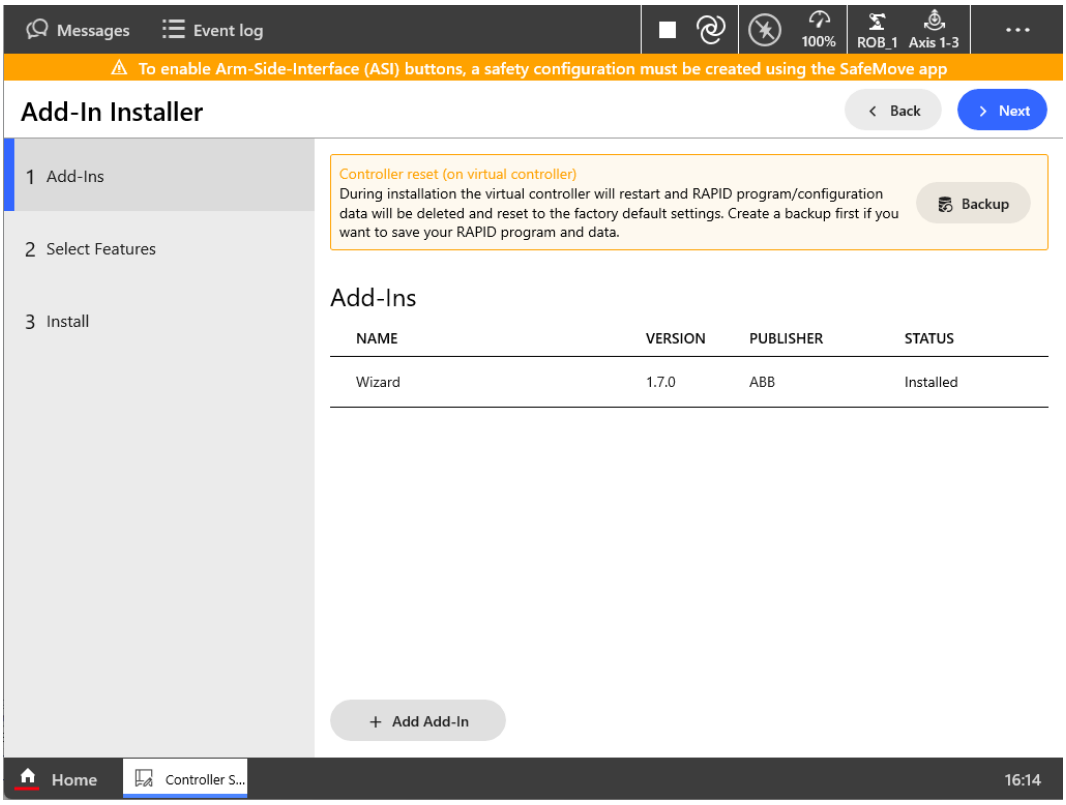
2.1.2 Installation via FlexPendant

To install the GRIPLINK plug-in using the FlexPendant, you do not need a PC connected to the robot controller. Instead, the plug-in is transferred using a USB stick that is connected to the FlexPendant. Carry out the following steps:

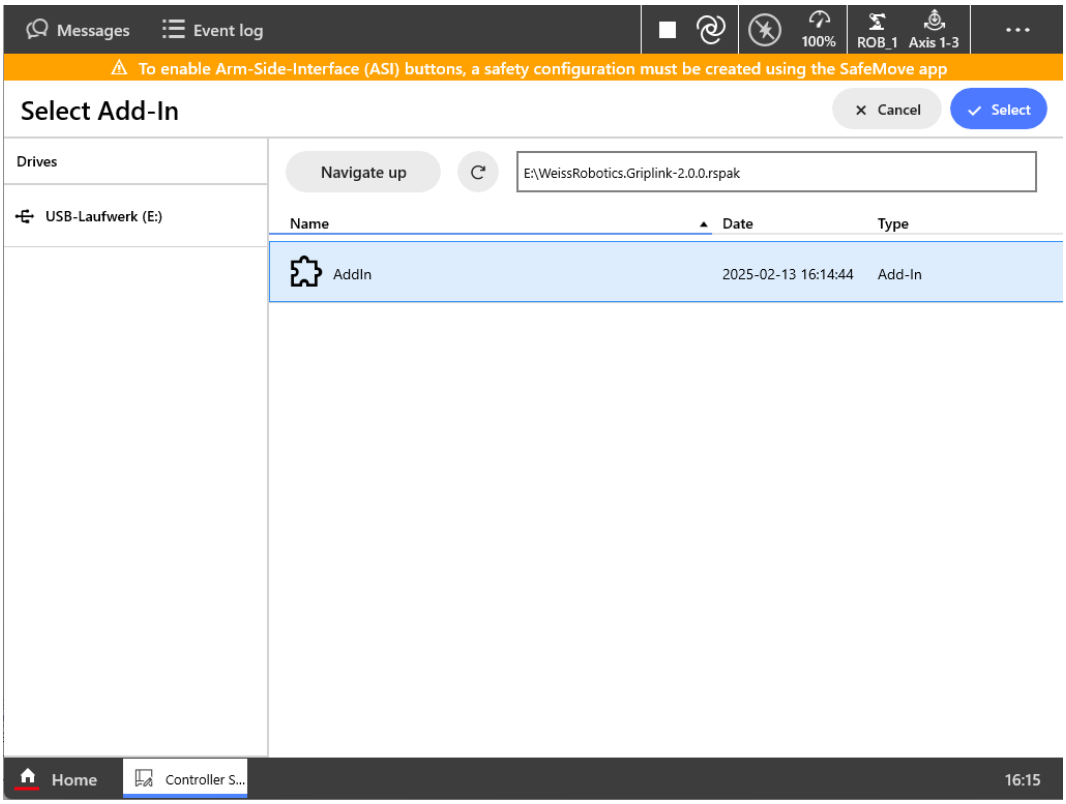
1. Download the latest version of the GRIPLINK plug-in for ABB from **www.griplink.de**, unzip the ZIP file and copy the plug-in to the USB stick.
2. Start the robot controller and connect the USB stick to the FlexPendant.
3. Start the App *Controller software* in the main menu of the FlexPendant and open the *Add-In Installer* there.



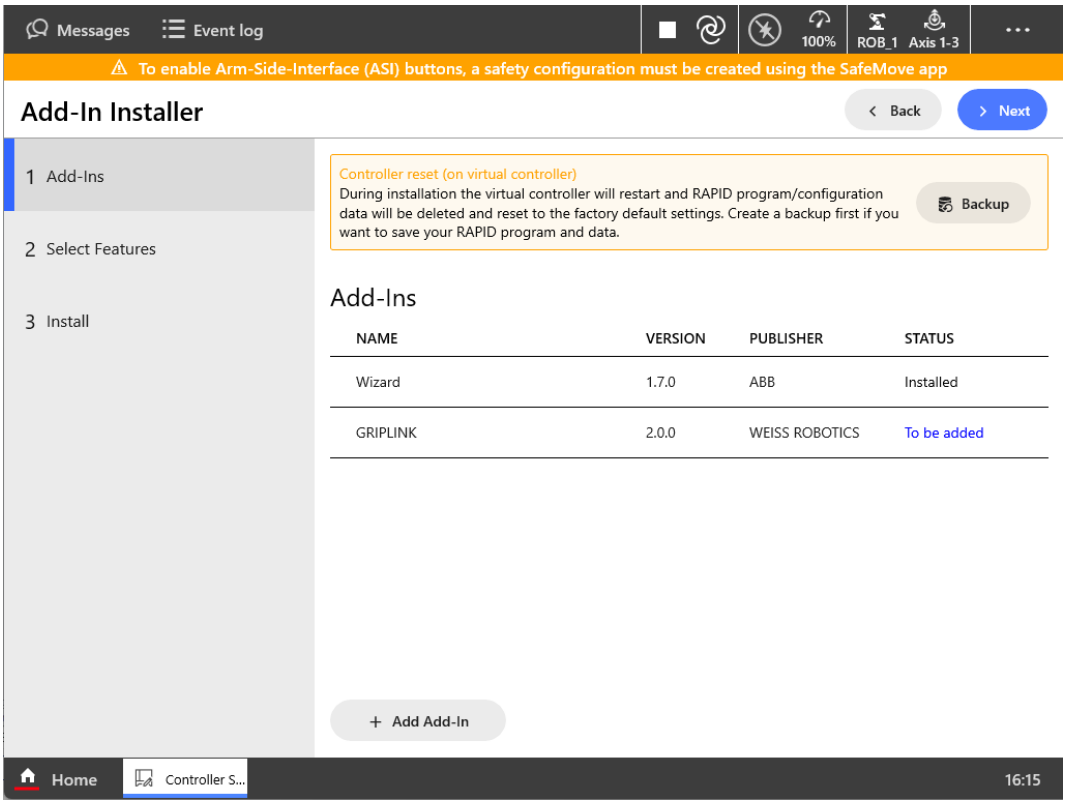
4. In the *Add-In Installer*, tap the *Add Add-In* button at the bottom.



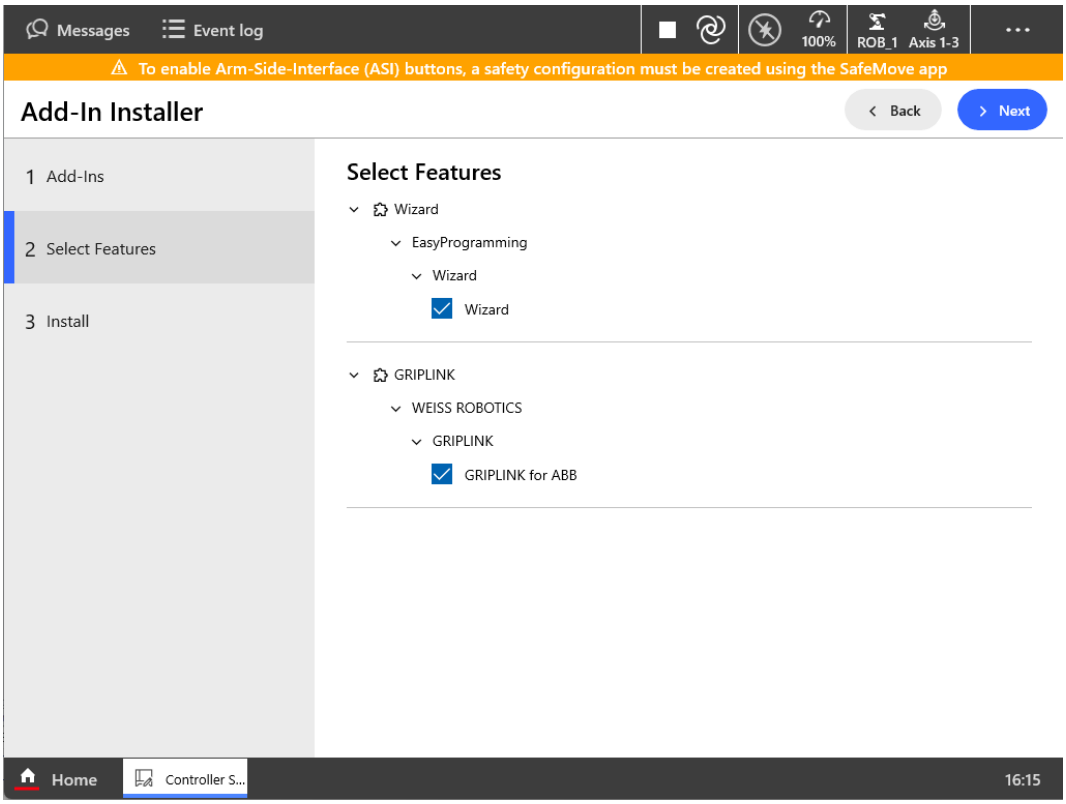
5. Navigate to the location of the plug-in on the USB stick. Select the *AddIn* contained in the file with the extension *.rspak* and confirm your selection with *Select* in the top right corner.



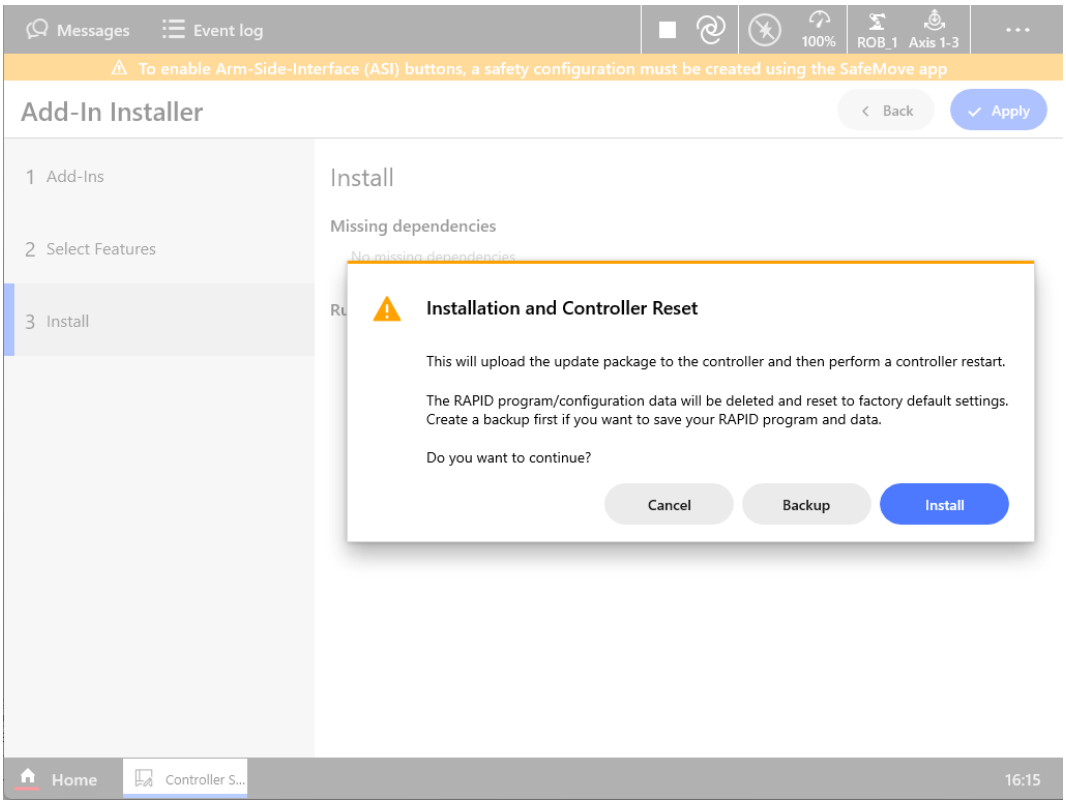
6. In the *Add-In Installer*, the GRIPLINK plug-in is displayed under *Add-Ins* with status *To be added.* Tap *Next* at the top right.



7. Under *Select Features*, *GRIPLINK for ABB* must be checked. Then, tap *Next* in the top right-hand corner.



8. Under *Install*, tap *Apply* at the top right. In the pop-up window, confirm that the installation and the associated restart should be carried out by tapping *Install*.



After the robot controller restart, the GRIPLINK plug-in is fully installed and the corresponding functionality is available.

2.2 Network Settings

The connection between the GRIPLINK controller and the robot controller is based on TCP/IP and is established via TCP port 10001. To establish a connection, it is necessary that the IP addresses of the GRIPLINK controller and the robot controller are compatible. The GRIPLINK controller has the following factory default network settings:

IP address:	192.168.1.40
Subnet mask:	255.255.255.0



The IP address and the subnet mask of the GRIPLINK controller can be configured using its web interface.



Information on configuring the IP address of the robot controller can be found in the corresponding operating manual.



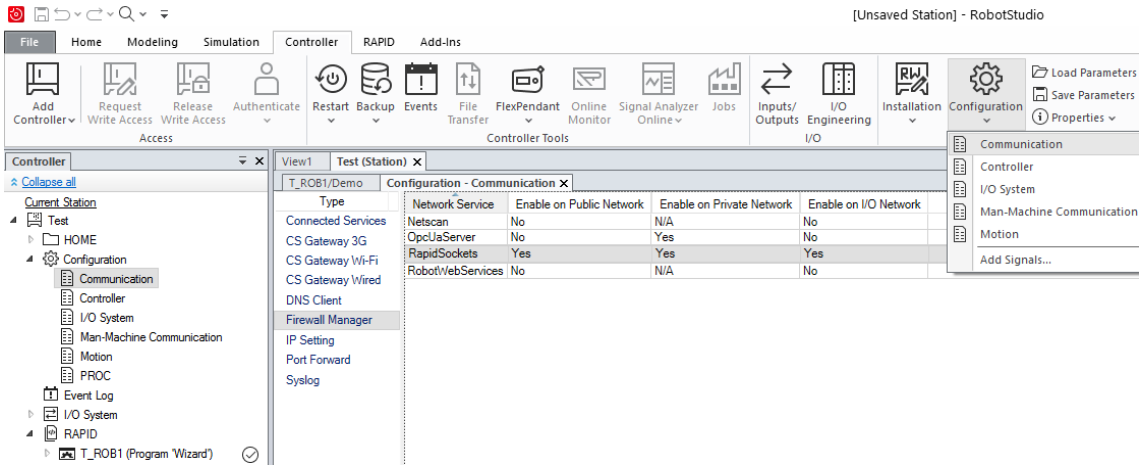
The IP address of the computer's network interface is used for virtual controllers in ABB RobotStudio®.



The IP addresses of the GRIPLINK controller and the robot controller must be in the same subnet.

2.3 Firewall Settings

RAPID sockets are used in the plug-in to establish the TCP/IP connection between GRIPLINK controller and robot controller. These RPAID sockets are therefore activated in the controller's firewall during the installation process. In some cases, however, it may be necessary to configure the firewall manually for the use of RAPID sockets. The corresponding settings can be adjusted using Robot Studio. They are located under the *Controller* tab at *Configuration* > *Communication* > *Firewall Manager*> *RapidSockets*. For the plug-in to function properly, make sure that these are activated for the appropriate type of network



3 Content of the Plug-In

The GRIPLINK plug-in provides a collection of commands to operate a GRIPLINK controller and the devices connected to it, both gripper modules from WEISS ROBOTICS and sensors and actuators from various third-party suppliers, directly via the ABB robot controller.

3.1 RAPID

Various commands of the GRIPLINK protocol are implemented as RAPID procedures in the GRIPLINK plug-in. After installing the plug-in, these procedures are available for use in user programs.



The functionality of the individual commands is explained in detail in the chapter**Command Reference** (4).



Information on programming with RAPID can be found in the corresponding ABB documentation.

```
3  PROC main()
4
5      GRIPLINK_Connect "192.168.1.40";
6      GRIPLINK_AssertDevice 0, 815, 6060;
7      GRIPLINK_SetGripConfig 0, 1, "Demo", 2.5, 100, 50, 0, 0, 0, 0, 0;
8      GRIPLINK_Home 0;
9      GRIPLINK_Release 0, 1, TRUE;
10
11     FOR nIndex FROM 1 TO 10 DO
12         WaitTime 1;
13         GRIPLINK_Grip 0, 1, TRUE;
14         WaitTime 1;
15         IF nGRIPLINK_State = GRIPLINK_DS_HOLDING THEN
16             WaitTime 3;
17             GRIPLINK_Release 0, 1, TRUE;
18         ELSEIF nGRIPLINK_State = GRIPLINK_DS_NO_PART THEN
19             WaitTime 1;
20             GRIPLINK_Release 0, 1, TRUE;
21         ELSE
22             Stop;
23         ENDIF
24     ENDFOR
25
26     GRIPLINK_Disconnect;
27
28 ENDPROC
```

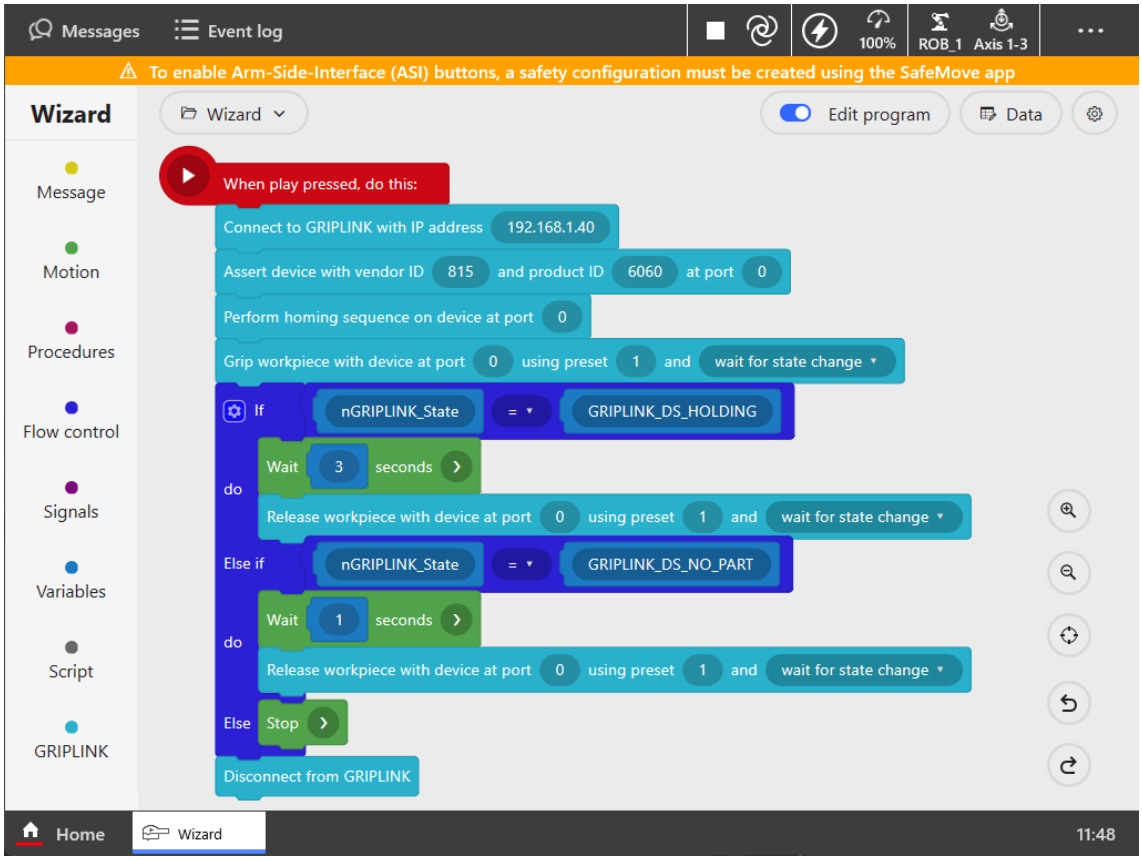
3.2 Wizard

In addition to the classic RAPID procedures, all commands are also included in the plug-in as blocks for the Wizard Easy Programming graphical programming interface. Using Wizard, simple robot applications can be programmed directly on the FlexPendant of an ABB robot.

The GRIPLINK function blocks can be found in the *GRIPLINK* category, which is available in the menu on the left-hand side of the Wizard screen after installing the plug-in. Each block corresponds to one of the RAPID procedures, so that it is also possible to switch between programming in RAPID and with Wizard to a certain extent ("Round-Tripping").

The functionality of the individual blocks is explained in detail in the chapter**Command Reference** (4).

General information on graphical programming with Wizard can be found in the corresponding ABB documentation.



3.3 Global Variables for Return Values

Any return values from commands are written to global variables by default. The following table lists all global variables defined for this purpose in the plug-in (in the module *GRIPLINK_Data*).

Global Variable	Type	Meaning
`nGRIPLINK_State`	`num`	Last device status read out
`nGRIPLINK_Value`	`num`	Last device value read out
`stGRIPLINK_Tag`	`string`	Last device tag read out



Alternatively, in RAPID, another variable can be specified for the return value when calling a command.

3.4 Device States

The following table lists all possible state values of connected devices. The constants specified in the first column are defined in the plug-in (in the module *GRIPLINK_Data*) and can be used in the robot program to compare with a queried device state.

Device State	Value	Meaning
GRIPLINK_DS_NOT_CONNECTED	0	Device not connected
GRIPLINK_DS_NOT_INITIALIZED	1	Device not initialized
GRIPLINK_DS_DISABLED	2	Device deactivated
GRIPLINK_DS_RELEASED	3	Workpiece released
GRIPLINK_DS_NO_PART	4	No workpiece found
GRIPLINK_DS_HOLDING	5	Workpiece gripped
GRIPLINK_DS_ENABLED	6	Device activated
GRIPLINK_DS_FAULT	7	Error

3.5 Error Handling

If problems occur when executing code from the GRIPLINK plug-in, an error is raised on the robot controller. If this error is not caught and processed in the robot program, execution is stopped.



It is the responsibility of the programmer to react appropriately to raised errors and to bring the system into a safe state! The tools required for this are provided by the ABB programming environment.

In order to distinguish between different types of errors, several constants for the RAPID error number (*ERRNO*) are used in the GRIPLINK plug-in. The corresponding values are reserved using the *BookErrNo* instruction when the command *GRIPLINK_Connect* is called. Thereby, the constants are automatically assigned a value from the available value range for user-defined error numbers that has not been assigned elsewhere. The following table lists the available constants and their respective meanings.

Error Number (ERRNO)	Error Description
`ERR_GRIPLINK_COMMAND`	Error during the execution of a command
`ERR_GRIPLINK_CONNECTION`	Error when connecting to the GRIPLINK controller
`ERR_GRIPLINK_DEVICE`	Error of a connected device
`ERR_GRIPLINK_PARAMETER`	Error due to invalid parameters
`ERR_GRIPLINK_TIMEOUT`	Error due to a WAITVAL wait timing out before the value reached the window



Further information about error handling in RAPID programs can be found in the instructions for your ABB robot controller and the RAPID documentation.

3.6 General Program Sequence

The basic sequence of a program using the GRIPLINK plug-in is typically as follows:

Preparation

- Establishing a connection to the GRIPLINK:
  GRIPLINK_Connect
- Checking the connected devices:
  GRIPLINK_AssertDevice
- Initialization of the connected devices:
  GRIPLINK_Enable / GRIPLINK_Home
- Configuration of the connected devices:
  GRIPLINK_SetGripConfig / ...

Main program (possibly as a loop)

- Commands for gripping and releasing:
  GRIPLINK_Grip / GRIPLINK_Release / GRIPLINK_FlexGrip / GRIPLINK_FlexRelease
- Query of states and values (and corresponding processing):
  GRIPLINK_DeviceState / GRIPLINK_Value / ...

Program end

- Deactivate the connected devices:
  GRIPLINK_Disable
- Close the connection to the GRIPLINK:
  GRIPLINK_Disconnect

3.7 Demo Programs

The *GRIPLINK_Demo* module contains several demo programs to demonstrate the typical use of the GRIPLINK plug-in and to enable quick tests of the plug-in. The respective procedures can either be called in the robot program or used as a template for custom programs.



The demo programs are intended for demonstration and test purposes only! They only contain rudimentary error handling, which might not be sufficient for production use.

4 Command Reference

The following chapter describes the commands contained in the GRIPLINK plug-in in detail. Both RAPID procedures and the corresponding blocks for graphical programming with Wizard are described.

All available commands are listed in the following table:

Command	Description
GRIPLINK_Connect	Establish connection to GRIPLINK controller
GRIPLINK_Disconnect	Close connection to GRIPLINK controller
GRIPLINK_AssertDevice	Check connected device
GRIPLINK_DeviceTag	Query device tag
GRIPLINK_SetDeviceTag	Set device tag
GRIPLINK_Enable	Activate device
GRIPLINK_Disable	Deactivate device
GRIPLINK_Home	Reference device
GRIPLINK_Grip	Grip workpiece
GRIPLINK_Release	Release workpiece
GRIPLINK_FlexGrip	Grip workpiece using specific parameters
GRIPLINK_FlexRelease	Release workpiece using specific parameters
GRIPLINK_LED	Select LED visualization
GRIPLINK_Clamp	Control mechanical gripping force retention
GRIPLINK_WaitForStateTransition	Wait for a device to change state
GRIPLINK_SetValue	Set device value
GRIPLINK_WaitValue	Wait for device value
GRIPLINK_DeviceState	Query device state
GRIPLINK_Value	Query device value
GRIPLINK_SetGripConfigGRIPLINK_Set GripConfig: Configure Grip Parameters	Configure gripping parameters

4.1 GRIPLINK_Connect: Establish Connection

This command establishes a connection between a GRIPLINK controller with the specified IPv4 address and the ABB robot controller. The protocol version supported by the GRIPLINK controller is also checked and an error is raised if it does not meet the minimum requirements.



The factory default IP address is 192.168.1.40 (subnet mask 255.255.255.0).



The IP address and the subnet mask of the GRIPLINK controller can be configured using its web interface.



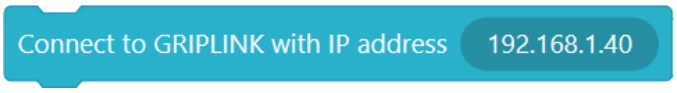
The IP addresses of the GRIPLINK controller and the robot controller must be in the same subnet.

RAPID Command

```
PROC GRIPLINK_Connect(
  string stIPAddr
)
```

Parameter	Type	Meaning
`stIPAddr`	`string`	IPv4 address of the GRIPLINK controller

Wizard Block



Example

Establishing a connection to a GRIPLINK controller with IP address 192.168.1.40:

```
GRIPLINK_Connect "192.168.1.40";
```

4.2 GRIPLINK_Disconnect: Close Connection

This command closes the current connection between a GRIPLINK controller and the ABB robot controller.

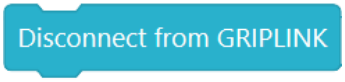


After the connection has been closed, no other commands can be sent until the command *GRIPLINK_Connect* (4.1) is called again!

RAPID Command

```
PROC GRIPLINK_Disconnect()
```

Wizard Block



Example

Closing the connection to a GRIPLINK controller:

```
GRIPLINK_Disconnect;
```

4.3 GRIPLINK_AssertDevice: Check Connected Device

This command checks whether a device with the specified IO-Link vendor ID and product ID is connected to a specific port. If this is not the case, an error is raised and the program is stopped. This check is generally recommended before a device is used in the program sequence, to ensure that the expected device is really connected.



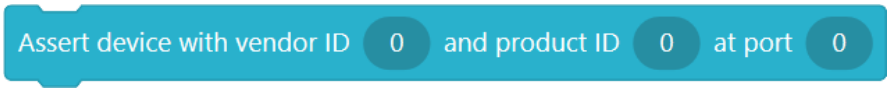
The vendor ID and product ID can be found in the operating manual for the respective device or in the table available at **www.griplink.de** under *Downloads*.

RAPID Command

```
PROC GRIPLINK_AssertDevice(
  num nPort,
  num nVID,
  num nPID
)
```

Parameter	Type	Meaning
nPort	num	GRIPLINK port
nVID	num	IO-Link vendor ID of the expected device
nPID	num	IO-Link device ID of the expected device

Wizard Block



Example

Checking for an IEG PLUS 260-030 (Product-ID 24) from WEISS ROBOTICS (Vendor-ID 815) on port 0 of a GRIPLINK controller:

```
GRIPLINK_AssertDevice 0, 815, 24;
```

4.4 GRIPLINK_DeviceTag: Query Device Tag

This command queries the tag of the device on a specific GRIPLINK port. This tag can be used, for example, to uniquely identify the device in more complex scenarios with many devices.

RAPID Command

```
PROC GRIPLINK_DeviceTag(
  num nPort,
  \VAR string stTag
)
```

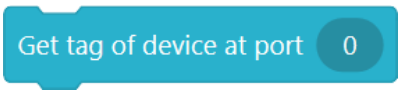
Parameter	Type	Meaning
nPort	num	GRIPLINK port
\stTag	string	optional variable for the return value

Return Value	Type	Meaning
stGRIPLINK_Tag	string	current device tag



If the optional argument \stTag is specified, the return value is only written to this variable.

Wizard Block



Example

Getting the tag of the device on port 1 of a GRIPLINK controller:

```
GRIPLINK_DeviceTag 1;
```

4.5 GRIPLINK_SetDeviceTag: Set Device Tag

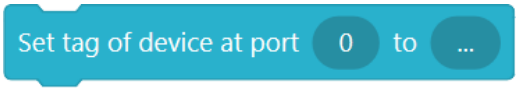
This command sets the tag of the device on a specific GRIPLINK port. This tag can be used, for example, to uniquely identify the device in more complex scenarios with many devices.

RAPID Command

```
PROC GRIPLINK_SetDeviceTag(
  num nPort,
  string stTag
)
```

Parameter	Type	Meaning
nPort	num	GRIPLINK port
stTag	string	device tag to be set

Wizard Block



Example

Setting the tag of a device on port 1 of a GRIPLINK controller:

```
GRIPLINK_SetDeviceTag 1, "Device_1";
```

4.6 GRIPLINK_Enable: Activate Device

This command activates the device connected to a specific GRIPLINK port. The exact effect of this command depends on the device in question. For some devices, errors can be acknowledged by deactivating and then activating them again.



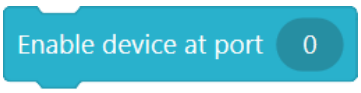
Grippers from WEISS ROBOTICS are automatically (re)activated by referencing or by executing grip or release commands.

RAPID Command

```
PROC GRIPLINK_Enable(
  num nPort
)
```

Parameter	Type	Meaning
nPort	num	GRIPLINK port

Wizard Block



Example

Activating the device on port 2 of a GRIPLINK controller:

```
GRIPLINK_Enable 2;
```

4.7 GRIPLINK_Disable: Deactivate Device

This command deactivates the device connected to a specific GRIPLINK port. The exact effect of this command depends on the device in question. For some devices, errors can be acknowledged by deactivating and then activating them again.



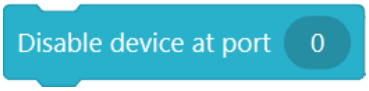
With grippers by WEISS ROBOTICS, this command deactivates the drive, so the fingers of the gripper can be moved by hand.

RAPID Command

```
PROC GRIPLINK_Disable(
  num nPort
)
```

Parameter	Type	Meaning
nPort	num	GRIPLINK port

Wizard Block



Example

Deactivating the device on port 2 of a GRIPLINK controller:

```
GRIPLINK_Disable 2;
```

4.8 GRIPLINK_Home: Reference Device

This command carries out the referencing of the device connected to a specific GRIPLINK port. Depending on the device, such an initialization may be necessary to ensure proper function. With grippers, the fingers are usually moved to the inner or outer mechanical stop to reference the position of the fingers. The command blocks further execution of the program until the referencing has been completed.



Further details on referencing can be found in the operating manual of the respective device.



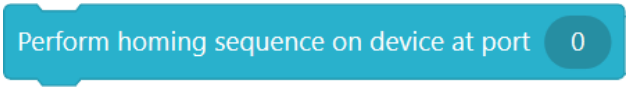
For gripping modules, the web interface of the GRIPLINK controller can be used to configure the direction of the reference run.

RAPID Command

```
PROC GRIPLINK_Home(
  num nPort
)
```

Parameter	Type	Meaning
nPort	num	GRIPLINK port

Wizard Block



Example

Referencing the device on port 3 of a GRIPLINK controller:

```
GRIPLINK_Home 3;
```

4.9 GRIPLINK_Grip: Grip Workpiece

This command grips a workpiece with the gripper connected to a specific GRIPLINK port. The grip parameters are taken from the given grip recipe. Further execution of the program can either take place immediately after the command is sent and before the movement of the fingers is completed, or it can be interrupted until the device state changes and the gripping process is completed. If the command waits for a state transition, the new device state is returned.



The grip recipes can be configured using either the web interface of the GRIPLINK controller or the *GRIPLINK_SetGripConfig* command (4.20).

RAPID Command

```
PROC GRIPLINK_Grip(
  num nPort,
  num nIndex,
  bool bBlocking,
  \VAR num nDevState
)
```

Parameter	Type	Meaning
`nPort`	`num`	GRIPLINK port
`nIndex`	`num`	index of the grip recipe
`bBlocking`	`bool`	`TRUE`: block, until either workpiece was gripped or not found `FALSE`: return immediately after motion was initiated
`\nDevState`	`num`	optional variable for the return value

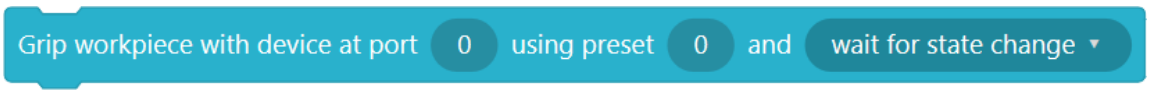
Return Value	Type	Meaning
`nGRIPLINK_State`	`num`	device state at the end of the gripping process: `GRIPLINK_DS_HOLDING`: workpiece gripped `GRIPLINK_DS_NO_PART`: no workpiece found `GRIPLINK_DS_FAULT`: error



The new device state is only written to the return variable if the command waits for a state transition (`bBlocking = TRUE`).

If the optional argument `\nDevState` is specified, the return value is only written to this variable.

Wizard Block



Example

Gripping with the gripper on port 0 of a GRIPLINK controller using gripping recipe 1. The program shall wait for the gripping process to be completed, so that further steps can depend on the new device state:

```
! Create variable for device state
VAR num nDeviceState;

! Grip with gripper at port 0 using recipe 1
GRIPLINK_Grip 0, 1, TRUE, \nDevState := nDeviceState;

! Check new device state
IF nDeviceState = GRIPLINK_DS_HOLDING THEN
  ! Workpiece gripped, move to target position
ELSEIF nDeviceState = GRIPLINK_DS_NO_PART THEN
  ! No workpiece found, wait and try again
ELSE
  ! Error, stop program
ENDIF
```

4.10 GRIPLINK_Release: Release Workpiece

This command releases a workpiece with the gripper connected to a specific GRIPLINK port. The release parameters are taken from the given grip recipe. Further execution of the program can either take place immediately after the command is sent and before the movement of the fingers is completed, or it can be interrupted until the device state changes and the release process is completed. If the command waits for a state transition, the new device state is returned.



The grip recipes can be configured using either the web interface of the GRIPLINK controller or the *GRIPLINK_SetGripConfig* command (4.20).

RAPID Command

```
PROC GRIPLINK_Release(
  num nPort,
  num nIndex,
  bool bBlocking,
  \VAR num nDevState
)
```

Parameter	Type	Meaning
nPort	num	GRIPLINK port
nIndex	num	index of the grip recipe
bBlocking	bool	TRUE: block, until the release motion was finished FALSE: do not wait for state change of the device
\nDevState	num	optional variable for the return value

Return Value	Type	Meaning
nGRIPLINK_State	num	device state at the end of the release process: GRIPLINK_DS_RELEASED: workpiece released GRIPLINK_DS_FAULT: error

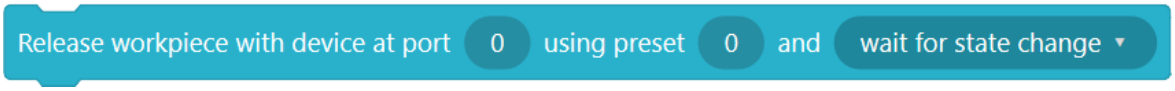


The new device state is only written to the return variable if the command waits for a state transition (`bBlocking = TRUE`).



If the optional argument `\nDevState` is specified, the return value is only written to this variable.

Wizard Block



Example

Releasing a workpiece with the gripper at port 0 of a GRIPLINK controller using gripper recipe 1. The program shall wait for the release process to be completed, to ensure that the workpiece has been released correctly:

```
! Release with gripper at port 0 using recipe 1
GRIPLINK_Release 0, 1, TRUE;

! Check new device state
IF nGRIPLINK_State = GRIPLINK_DS_RELEASED THEN
  ! Workpiece released, continue
ELSE
  ! Error, stop program
ENDIF
```

4.11 GRIPLINK_FlexGrip: Grip Workpiece with Flexible Parameters

This command grips a workpiece with the gripper connected to a specific GRIPLINK port. The grip parameters can be specified directly, independently of the stored grip recipes. Further execution of the program can either take place immediately after the command is sent and before the fingers have finished moving, or it can be interrupted until the device state changes and the gripping process is completed. If the command waits for a state transition, the new device state is returned.



Further information on grip parameters and the permissible value ranges can be found in the operating manual of the respective gripping module.



This function is not available on all devices!

RAPID Command

```
PROC GRIPLINK_FlexGrip(
  num nPort,
  num nPosition,
  num nForce,
  num nSpeed,
  num nAcceleration,
  bool bBlocking,
  \VAR num nDevState
)
```

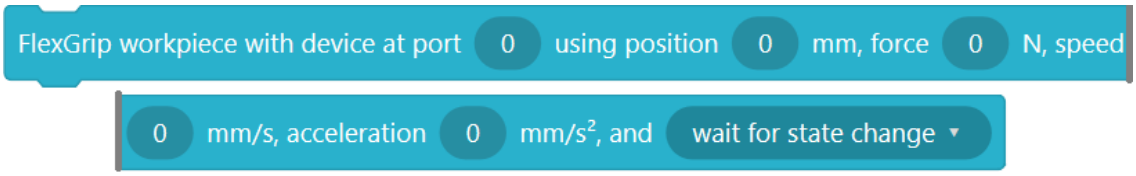
Parameter	Type	Meaning
nPort	num	GRIPLINK port
nPosition	num	target position (no-part limit) in mm
nForce	num	gripping force in N
nSpeed	num	speed in mm/s (0: calculate automatically)
nAcceleration	num	acceleration in mm/s² (0: use default value)
bBlocking	bool	TRUE: Block, until either workpiece was gripped or not found FALSE: Return immediately after motion was initiated
\nDevState	num	optional variable for the return value

Return Value	Type	Meaning
nGRIPLINK_State	num	device state at the end of the gripping process: GRIPLINK_DS_HOLDING: workpiece gripped GRIPLINK_DS_NO_PART: no workpiece found GRIPLINK_DS_FAULT: error

The new device state is only written to the return variable if the command waits for a state transition (bBlocking = TRUE).

If the optional argument \nDevState is specified, the return value is only written to this variable.

Wizard Block



Example

Gripping a workpiece larger than 12.5 mm with the gripper at port 0 of a GRIPLINK controller, using a gripping force of 60 N, a speed of 45 mm/s and standard acceleration. The program shall wait for the gripping process to be completed, so that further steps can depend on the new device state:

```
! Grip with gripper at port 0 using custom parameters
GRIPLINK_FlexGrip 0, 12.5, 60, 45, 0, TRUE;

! Check new device state
IF nGRIPLINK_State = GRIPLINK_DS_HOLDING THEN
  ! Workpiece gripped, move to target position
ELSEIF nGRIPLINK_State = GRIPLINK_DS_NO_PART THEN
  ! No workpiece found, wait and try again
ELSE
  ! Error, stop program
ENDIF
```

4.12 GRIPLINK_FlexRelease: Release Workpiece with Flexible Parameters

This command releases a workpiece with the gripper connected to a specific GRIPLINK port. The release parameters can be specified directly, independently of the stored grip recipes. This command can also be used for pre-positioning the gripper fingers. Further execution of the program can either take place immediately after the command is sent and before the fingers have finished moving, or it can be interrupted until the device state changes and the release process is completed. If the command waits for a state transition, the new device state is returned.



Further information on release parameters and the permissible value ranges can be found in the operating manual of the respective gripping module.



This function is not available on all devices!

RAPID Command

```
PROC GRIPLINK_FlexRelease(
  num nPort,
  num nPosition,
  num nSpeed,
  num nAcceleration,
  bool bBlocking,
  \VAR num nDevState
)
```

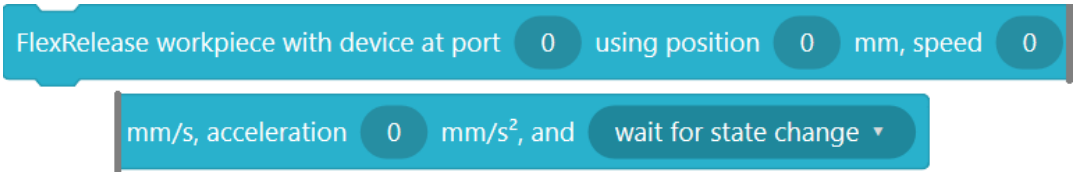
Parameter	Type	Meaning
nPort	num	GRIPLINK port
nPosition	num	target position (release limit) in mm
nSpeed	num	speed in mm/s (0: use default value)
nAcceleration	num	acceleration in mm/s² (0: use default value)
bBlocking	bool	TRUE: block, until the release motion was finished FALSE: return immediately after motion was initiated
\nDevState	num	optional variable for the return value

Return Value	Type	Meaning
`nGRIPLINK_State`	`num`	device state at the end of the release process: `GRIPLINK_DS_RELEASED`: workpiece released `GRIPLINK_DS_FAULT`: error

The new device state is only written to the return variable if the command waits for a state transition (`bBlocking = TRUE`).

If the optional argument `\nDevState` is specified, the return value is only written to this variable.

Wizard Block



Example

Carefully gripping a workpiece larger than 12.5 mm with the gripper at port 0 of a GRIPLINK controller, using gripping force 80 N, speed 25 mm/s, and acceleration 25 mm/s². The fingers shall first be pre-positioned at a larger speed to a position of 20 mm using *GRIPLINK_FlexRelease*:

```
! Pre-position fingers
GRIPLINK_FlexRelease 0, 20, 150, 300, FALSE;

! Wait until finger position is reached
GRIPLINK_WaitValue 0, 0, 20, 0.2, 2500;
WaitTime 0.1;

! Grip with custom parameters
GRIPLINK_FlexGrip 0, 12.5, 50, 25, 25, TRUE;
```

4.13 GRIPLINK_LED: Select LED Visualization

This command configures the LED status display of the device connected to a specific GRIPLINK port. Supported devices have several configurable sets of settings that can be selected with this command.



Further information on the LED visualization and its configuration options can be found in the operating manual of the respective device.



This function is not available on all devices!

RAPID Command

```
PROC GRIPLINK_LED(
  num nPort,
  num nIndex
)
```

Parameter	Type	Meaning
nPort	num	GRIPLINK port
nIndex	num	index of the LED recipe

Wizard Block



Example

Selecting the LED visualization 2 of the device at port 3 of a GRIPLINK controller:

```
GRIPLINK_LED 3, 2;
```

4.14 GRIPLINK_Clamp: Control Gripping Force Retention

This command activates or deactivates the mechanical gripping force retention of a gripper connected to a specific GRIPLINK port. This gripping force retention makes it possible to switch off the motor control during prolonged gripping of a workpiece and still hold the workpiece securely.



Further information on mechanical gripping force retention can be found in the operating manual for the respective gripper.



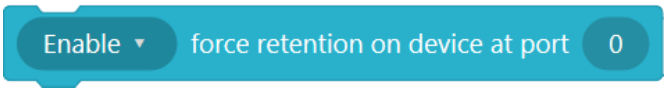
This function is not available on all devices!

RAPID Command

```
PROC GRIPLINK_Clamp(
  num nPort,
  bool bEnable
)
```

Parameter	Type	Meaning
nPort	num	GRIPLINK port
bEnable	bool	TRUE: activate mechanical gripping force retention FALSE: deactivate mechanical gripping force retention

Wizard Block



Example

Gripping a workpiece with the gripper on port 0 of a GRIPLINK controller for a longer period, using mechanical gripping force retention to prevent overheating:

```
! Activating the gripping force retention
GRIPLINK_Clamp 0, TRUE;

! Grip with gripper at port 0 using recipe 1
GRIPLINK_Grip 0, 1, TRUE;

! Wait some time, gripping force retention becomes active
WaitTime 600;
```

4.15 GRIPLINK_WaitForStateTransition: Wait for State Transition

This command waits for a change of the state of the device at a specific GRIPLINK port. It can be used, for example, to detect the end of a gripping or release process. Once the status has changed, the new device state is returned. If no status change occurs, an error is raised after a certain time.



Alternatively, waiting for a state change can also be triggered directly by calling grip and release commands with the respective parameter (`bBlocking = TRUE`).

RAPID Command

```
PROC GRIPLINK_WaitForStateTransition(
  num nPort,
  \VAR num nDevState
)
```

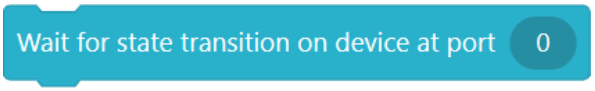
Parameter	Type	Meaning
`nPort`	`num`	GRIPLINK port
`\nDevState`	`num`	optional variable for the return value

Return Value	Type	Meaning
`nGRIPLINK_State`	`num`	device status after state change



If the optional argument `\nDevState` is specified, the return value is only written to this variable.

Wizard Block



Example

Gripping simultaneously with two grippers on ports 1 and 2 of a GRIPLINK controller. The program shall continue once both gripping processes have been completed successfully:

```
! Create variables for device states
VAR num nState1;
VAR num nState2;

! Grip simultaneously (without waiting for state change)
GRIPLINK_Grip 1, 0, FALSE;
GRIPLINK_Grip 2, 0, FALSE;

! Wait for state changes on both grippers
GRIPLINK_WaitForStateTransition 1, \nDevState := nState1;
GRIPLINK_WaitForStateTransition 2, \nDevState := nState2;

! Check new device states
IF nState1 <> GRIPLINK_DS_HOLDING THEN
  ! Workpiece 1 not gripped
ENDIF
IF nState2 <> GRIPLINK_DS_HOLDING THEN
  ! Workpiece 2 not gripped
ENDIF
```

4.16 GRIPLINK_SetValue: Set Device Value

This command sets a value on the device at a specific GRIPLINK port. The available values are selected by an associated index. The exact effect of this command depends on the respective device.



An overview of the values and their indices available for various devices can be found in the table available at **www.griplink.de** under *Downloads*.



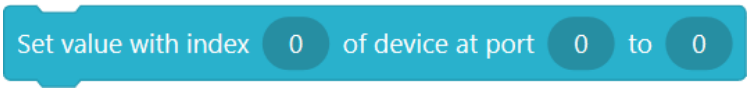
This function is not available on all devices!

RAPID Command

```
PROC GRIPLINK_SetValue(
  num nPort,
  num nIndex,
  num nValue
)
```

Parameter	Type	Meaning
nPort	num	GRIPLINK port
nIndex	num	index of the value to be set
nValue	num	value to be set

Wizard Block



Example

Setting the value with index 0 on the device at port 1 of a GRIPLINK controller to 42:

```
GRIPLINK_SetValue 1, 0, 42;
```

4.17 GRIPLINK_WaitValue: Wait for Device Value

This command waits for a given value to be reached on the device at a specific GRIPLINK port. Specifically, it waits for a certain time until the device value enters a certain range (window) around the target value (threshold value). If the time is exceeded, an error is raised.



An overview of the values and their indices available for various devices can be found in the table available at **www.griplink.de** under *Downloads*.



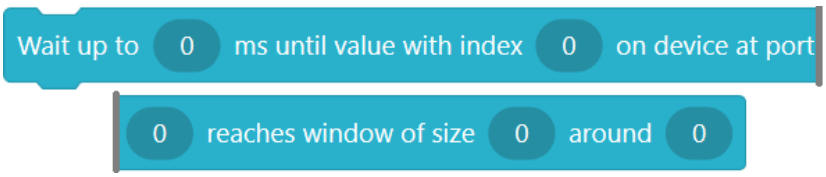
The window size should be chosen according to the rate of change of the respective device value to ensure reliable and accurate detection.

RAPID Command

```
PROC GRIPLINK_WaitValue(
  num nPort,
  num nIndex,
  num nValue,
  num nWindow,
  num nTimeout
)
```

Parameter	Type	Meaning
nPort	num	GRIPLINK port
nIndex	num	index of the value to be set
nValue	num	threshold value to be reached
nWindow	num	size of the window around the threshold value within that the value is considered to be reached
nTimeout	num	maximum duration to wait in ms

Wizard Block



Example

Waiting until the value with index 1 of the device at port 0 of a GRIPLINK controller reaches the value 10± 1.5. A timeout of 20 seconds shall be used:

```
GRIPLINK_WaitValue 0, 1, 10, 1.5, 20000;
```

4.18 GRIPLINK_DeviceState: Query Device State

This command queries the current device state of the device on a specific GRIPLINK port. This state provides information about the current status of the device.



A table with an overview of all possible device states and the corresponding predefined constants can be found in the section**Device Stat** (3.4).

RAPID Command

```
PROC GRIPLINK_DeviceState(
  num nPort,
  \VAR num nDevState
)
```

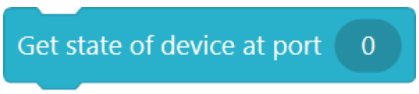
Parameter	Type	Meaning
`nPort`	`num`	GRIPLINK port
`\nDevState`	`num`	optional variable for the return value

Return Value	Type	Meaning
`nGRIPLINK_State`	`num`	current device state



If the optional argument `\nDevState` is specified, the return value is only written to this variable.

Wizard Block



Example

Querying the state of a gripper at port 3 of a GRIPLINK controller. The next steps in the program shall then depend on the device state:

```
! Query device state
GRIPLINK_DeviceState 3;

! Check device state
IF nGRIPLINK_State = GRIPLINK_DS_HOLDING THEN
  ! Workpiece gripped
ELSEIF nGRIPLINK_State = GRIPLINK_DS_NO_PART THEN
  ! No workpiece gripped
ELSEIF nGRIPLINK_State = GRIPLINK_DS_RELEASED THEN
  ! Workpiece released
ENDIF
```

4.19 GRIPLINK_Value: Query Device Value

This command queries a device value from the device on a specific GRIPLINK port. The available values are selected by an associated index.



An overview of the values and their indices available for various devices can be found in the table available at **www.griplink.de** under *Downloads*.

RAPID Command

```
PROC GRIPLINK_Value(
  num nPort,
  num nIndex,
  \VAR num nValue
)
```

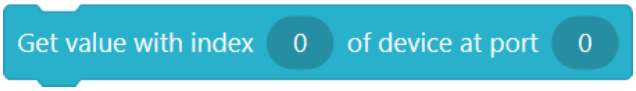
Parameter	Type	Meaning
`nPort`	`num`	GRIPLINK port
`nIndex`	`num`	index of the device value to queried
`\nValue`	`num`	optional variable for the return value

Return Value	Type	Meaning
`nGRIPLINK_Value`	`num`	current device value



If the optional argument `\nValue` is specified, the return value is only written to this variable.

Wizard Block



Example

Querying the size of a workpiece gripped with a gripper from WEISS ROBOTICS, which is connected to port 0 of a GRIPLINK controller. The device value with index 0 (opening width in mm) is used for this. The next steps in the program shall then depend on the determined workpiece size:

```
Grip workpiece with gripper at port 0 using recipe 0
GRIPLINK_Grip 0, 0, TRUE;

! Check new device state
IF nGRIPLINK_State = GRIPLINK_DS_HOLDING THEN

  ! Workpiece gripped, query finger position
  GRIPLINK_Value 0, 0;

  ! Check finger position
  IF nGRIPLINK_Value > 9.5 AND nGRIPLINK_Value < 10.5 THEN
    ! Correct workpiece gripped
  ELSE
    ! Wrong workpiece gripped
  ENDIF

ELSE
  ! No workpiece gripped
ENDIF
```

4.20 GRIPLINK_SetGripConfig: Configure Grip Parameters

This command configures one set of grip parameters for the device on a specific GRIPLINK port. Gripping modules can have several configurable sets of parameters that are selected via an index. These sets consist of up to eight parameters and a tag for identification and description.



Information on the meaning of the individual parameters for the respective device and the permissible value ranges can be found in the operating manual for the device or in the corresponding GRIPLINK device driver.



Alternatively, the web interface of the GRIPLINK controller can be used to configure the parameters of the grip recipes.



The grip parameters of WEISS ROBOTICS grippers have the following meanings:

0 target position when gripping (no-part limit) in mm
1 target position when releasing (release limit) in mm
2 force factor in %

Further optional parameters are used for grippers of the WPG series:

3 gripping speed in mm/s
4 gripping acceleration in mm/s²
5 release speed in mm/s
6 release acceleration in mm/s²

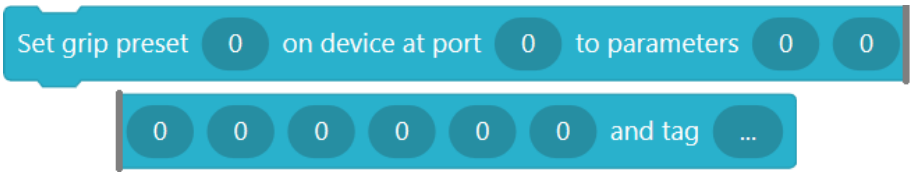
Unused parameters are simply set to 0.

RAPID Command

```
PROC GRIPLINK_SetGripConfig(
  num nPort,
  num nIndex,
  string stTag,
  num nParam0,
  num nParam1,
  num nParam2,
  num nParam3,
  num nParam4,
  num nParam5,
  num nParam6,
  num nParam7
)
```

Parameter	Type	Meaning
`nPort`	`num`	GRIPLINK port
`nIndex`	`num`	index of the gripper recipe to be configured
`stDay`	`string`	recipe tag to be set
`nParamX`	`num`	parameter value to be set (`0`: not used)

Wizard Block



Example

Configuring recipe 1 of a gripper by WEISS ROBOTICS at port 0 of a GRIPLINK controller. A no-part limit of 3.5 mm, a release limit of 12 mm, and a gripping force of 50 % are to be set:

```
GRIPLINK_SetGripConfig 0, 1, "Grip_1", 3.5, 12, 50, 0, 0, 0, 0, 0;
```

© 2025 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.

The 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 for systems whose failure could result in personal injury.