WEISS ROBOTICS

GRIPLINK
FOR UNIVERSAL ROBOTS

Version 2.1.0

August 2024

Contents

1 Introduction

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

❗ These instructions describe the functions of the FLEXGRIP plug-in for robots from Universal Robots. For information on installing, commissioning and operating the GRIPKIT-Easy gripping module, please refer to the operating instructions. These can be found online at **www.weiss-robotics.com/gripkit-easy/.**

1.1 Notation and symbols

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

⚠ Functional or safety-relevant information. Failure to observe this may endanger the safety of personnel and the system, damage the appliance or impair the function of the appliance.

❗ Additional information for a better understanding of the facts described.

☞ Reference to further information.

The GRIPLINK plug-in is compatible with all devices that support the GRIPLINK protocol via TCP/IP. This includes the GRIPLINK-ET4 and the gripping modules of the WPG series.

For ease of reading, these devices are summarized under the term „GRIPLINK Controller“.

1.2 Intended use

The „GRIPLINK-Plugin“ software is intended for communication between a GRIPLINK controller from WEISS ROBOTICS and a robot controller. The requirements of the applicable directives and the installation and operating instructions 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

This plugin is compatible with GRIPLINK-ET4 from firmware version 5.0.0 and WPG gripping modules from firmware version 2.0.1.

One of the following Universal Robots robot controllers is required for operation:

- UR CB3.1 with software version 3.11 or higher
- UR e-Series with software version 5.5 or higher

❗ The IP address of the GRIPLINK controller must be in the same subnet as that of the robot controller.

1.4 License terms

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

1.5 Demo programs

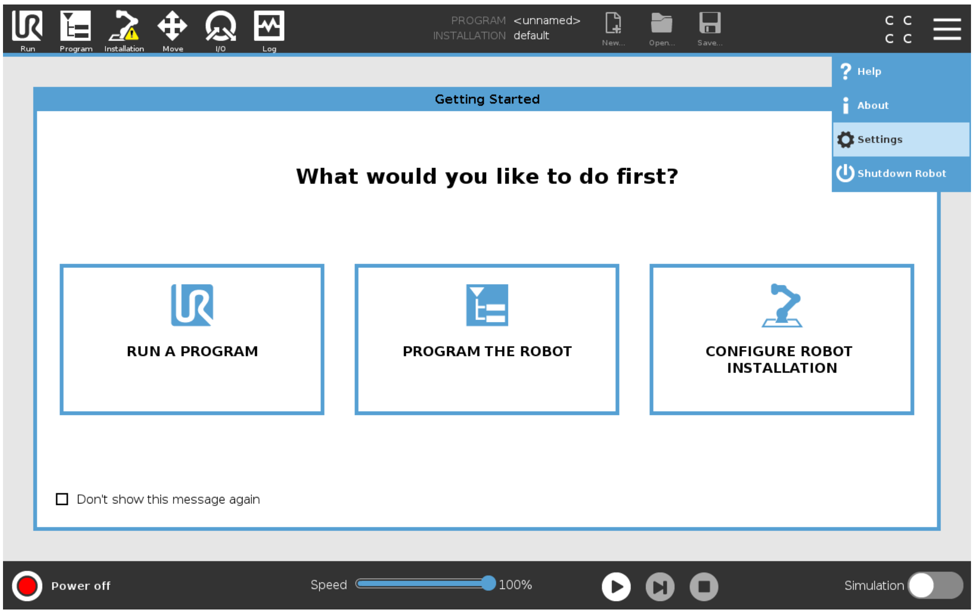
The demo programs included in the software package show how to use the plug-in. They are intended for test purposes only!

2 Installation

2.1 Installing the software

☞ Make sure that you are using the latest version of the GRIPLINK plugin. The latest version can be downloaded at www.griplink.de.

1. Download the plugin file „griplink_plugin_universalrobots_<version>.zip“.
2. Unpack the previously downloaded ZIP archive with the GRIPLINK plug-in into the root directory of a USB stick and insert it into the USB slot of the Teach Pendant.
3. Open the settings and navigate to the „System/URCaps“ menu



Getting Started

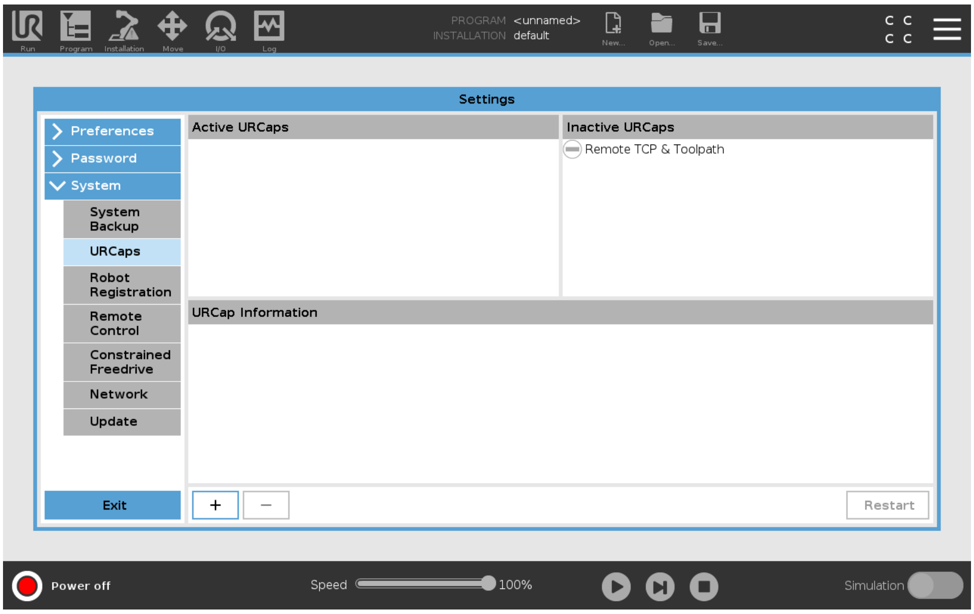
What would you like to do first?

RUN A PROGRAM

PROGRAM THE ROBOT

CONFIGURE ROBOT INSTALLATION

□ Don't show this message again

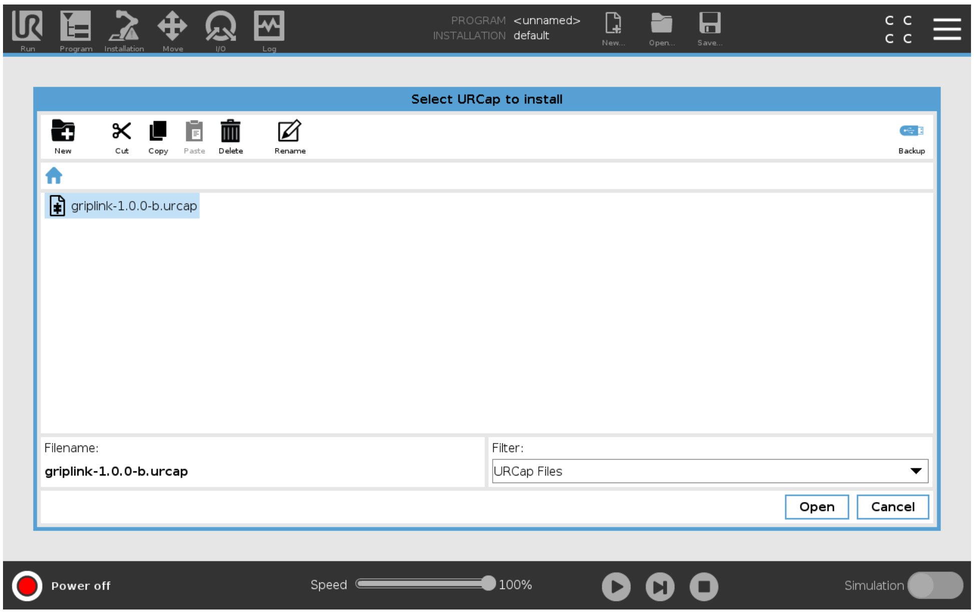


Settings

Active URCaps	Inactive URCaps
	Remote TCP & Toolpath

URCap Information

4. Press the „+“ button and select the previously unzipped .urcap file



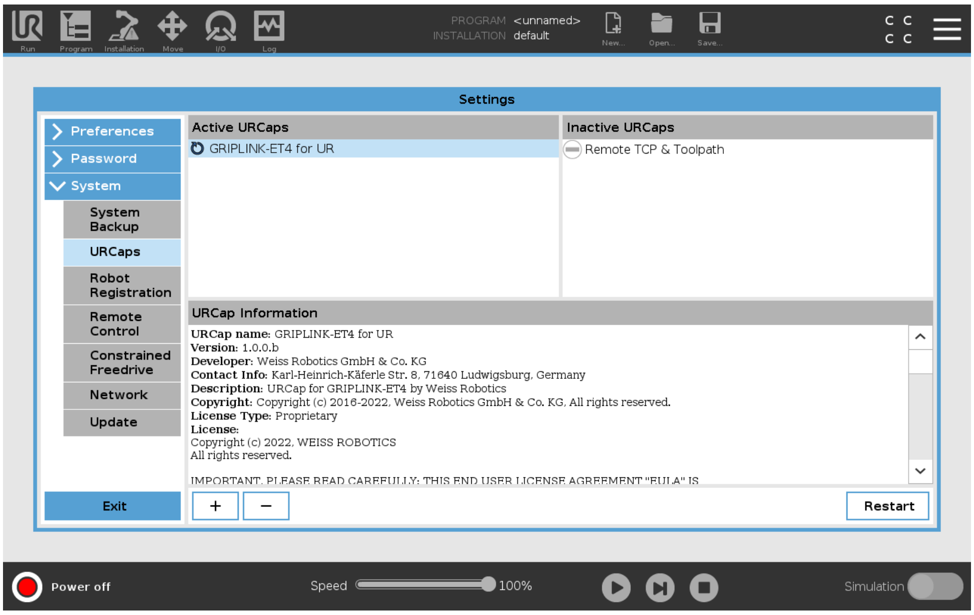
Select URCap to install

griplink-1.0.0-b.urcap

Filename: griplink-1.0.0-b.urcap

Filter: URCap Files

5. Restart the robot by pressing the “Restart“ button



Settings

Active URCaps: GRIPLINK-ET4 for UR

Inactive URCaps: Remote TCP & Toolpath

URCap Information

URCap name: GRIPLINK-ET4 for UR
Version: 1.0.0.b
Developer: Weiss Robotics GmbH & Co. KG
Contact Info: Karl-Heinrich-Käferle Str. 8, 71640 Ludwigsburg, Germany
Description: URCap for GRIPLINK-ET4 by Weiss Robotics
Copyright: Copyright (c) 2016-2022, Weiss Robotics GmbH & Co. KG, All rights reserved.
License Type: Proprietary
License:
Copyright (c) 2022, WEISS ROBOTICS
All rights reserved.

IMPORTANT. PLEASE READ CAREFULLY: THIS END USER LICENSE AGREEMENT "EULA" IS

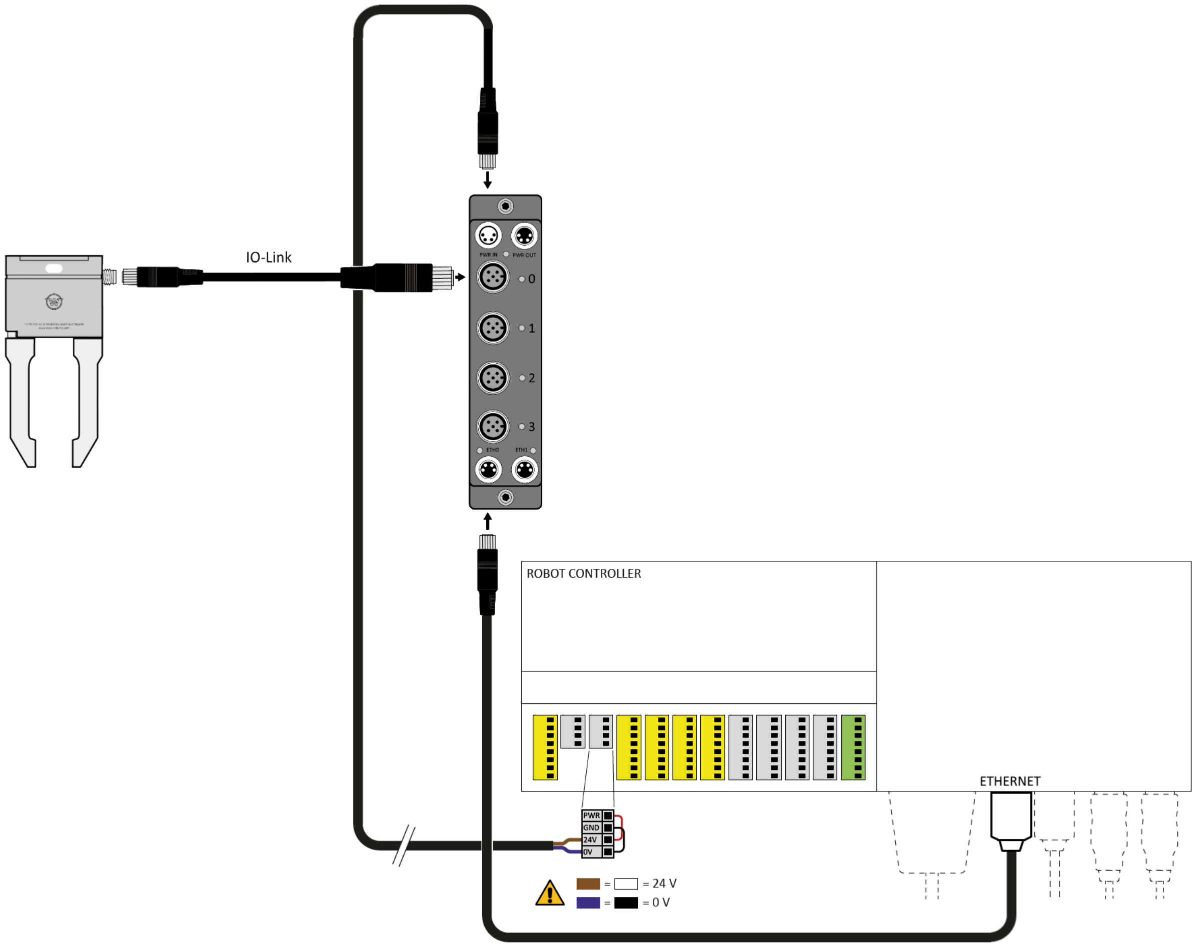
2.1.1 Checking the installation

After you have completed the installation process, the “GRIPLINK“ entry appears in the „Installation“ main menu, as well as various GRIPLINK nodes under the “URCaps“ menu item in the „Program“ main menu.

2.2 Uninstalling the software

To remove the GRIPLINK plug-in from your robot, follow the instructions in the robot controller manual.

3 Hardware setup



If you are not sure whether the power supply via the robot controller connection is sufficient to operate all devices connected to the GRIPLINK, use an external power supply!

The power supply must be adequately dimensioned in conjunction with the connected devices!

4 Functionality of the plugin

4.1 Operation via program node

After installation, the plugin provides various function nodes that can be inserted into a robot program like other robot commands.

4.2 Operation via URScript

The plugin includes a set of basic functions in URScript, which can also be called from within the robot program using the corresponding script program node. These basic functions are automatically integrated into the robot program when the plugin is installed.

In section 6 describes the available functions in more detail.

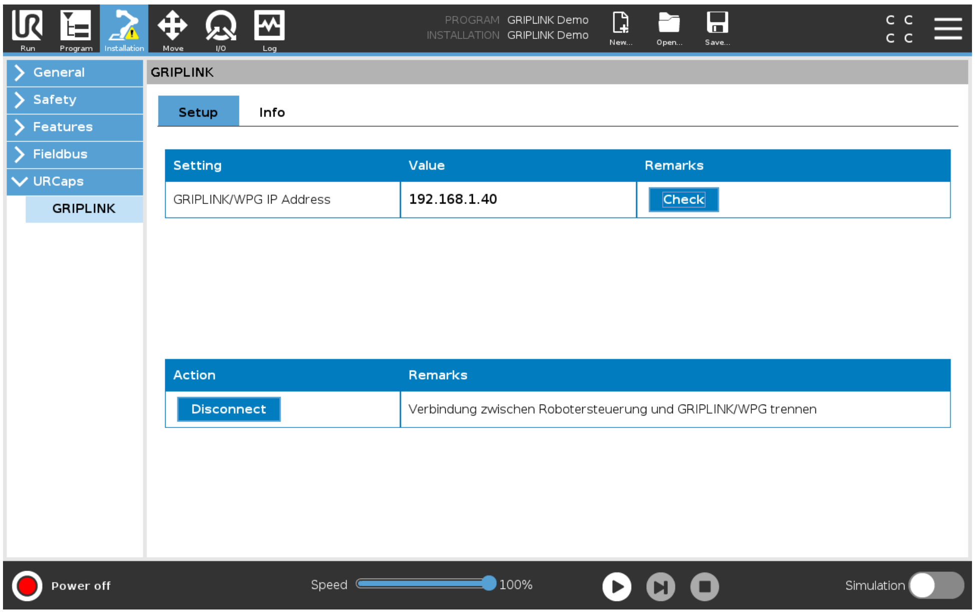
5 Preparing the robot

The basic settings of the GRIPLINK-URCap must be made in the corresponding installation.

The GRIPLINK controller is configured via the respective web interface, which can be accessed via the set IP address (default: 192.168.1.40).

5.1 Setup

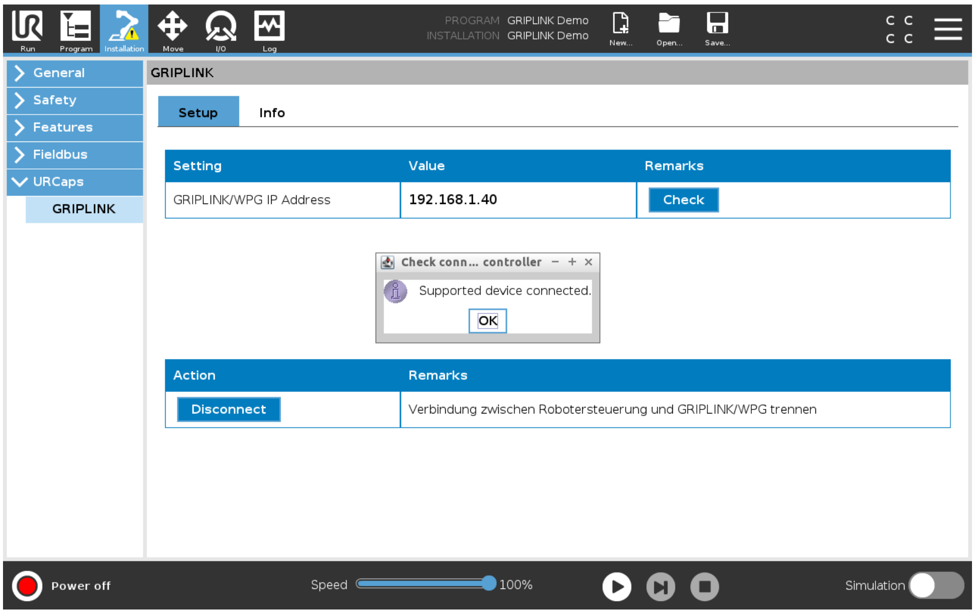
The IP address of the GRIPLINK controller connected to the robot is entered in the setup.



Setting	Value	Remarks
GRIPLINK/WPG IP Address	192.168.1.40	Check

Action	Remarks
Disconnect	Verbindung zwischen Robotersteuerung und GRIPLINK/WPG trennen

The „Check“ button can be used to check whether the configured GRIPLINK controller is connected.



GRIPLINK

Setting	Value	Remarks
GRIPLINK/WPG IP Address	192.168.1.40	Check

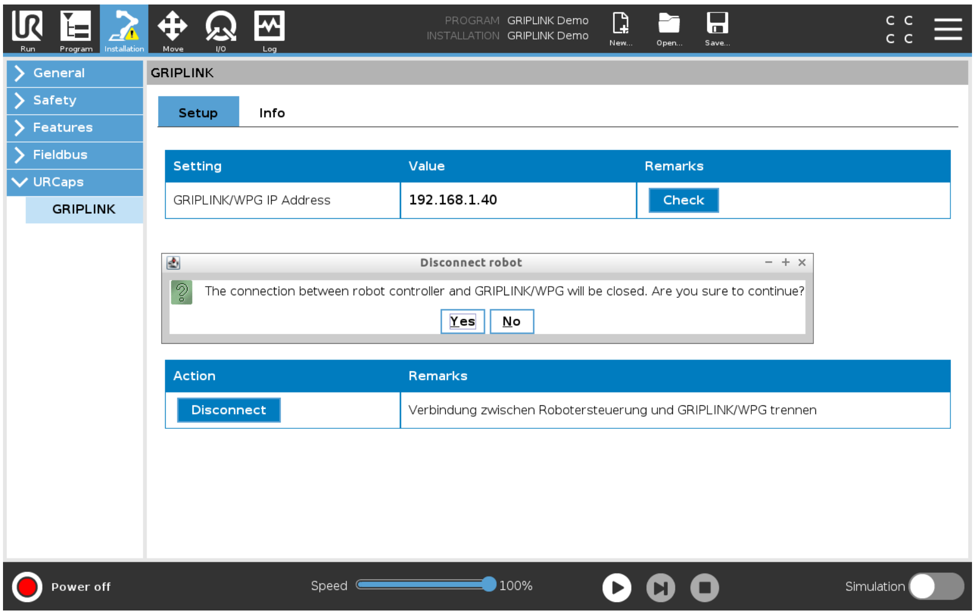
Check conn... controller

Supported device connected.

OK

Action	Remarks
Disconnect	Verbindung zwischen Robotersteuerung und GRIPLINK/WPG trennen

If a connection between the robot controller and GRIPLINK Controller is active, no gripping commands can be executed via the buttons of the program nodes. The „Disconnect“ button can be used to disconnect the connection:



GRIPLINK

Setting	Value	Remarks
GRIPLINK/WPG IP Address	192.168.1.40	Check

Disconnect robot

The connection between robot controller and GRIPLINK/WPG will be closed. Are you sure to continue?

Yes No

Action	Remarks
Disconnect	Verbindung zwischen Robotersteuerung und GRIPLINK/WPG trennen

Do not disconnect the connection while a robot program is running! Risk of damage and injury due to falling parts!

6 Program nodes (Nodes)

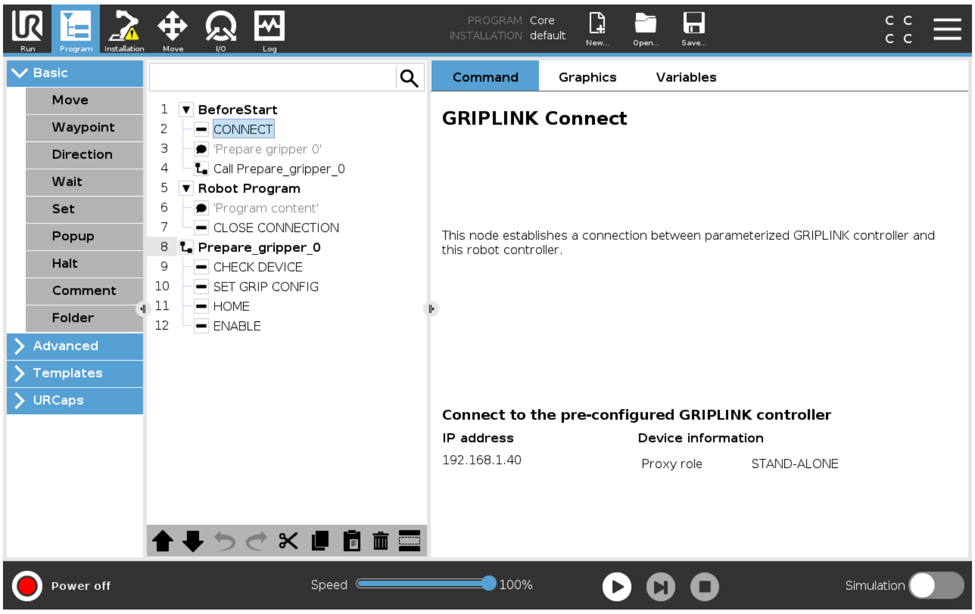
6.1 Basic program sequence

Every robot program that controls devices such as grippers and sensors via the GRIPLINK should follow the following design guidelines.

6.1.1 Program

<u>The BeforeStart section</u>

1. initially contains the „GRIPLINK Connect“ node
2. then contains the call of a subroutine for gripper initialization for each device connected to GRIPLINK.
   The correct device is ensured here using „GRIPLINK Check Device“ and then configured, referenced and activated for the specific device if necessary



```
▼ BeforeStart
   CONNECT
   'Prepare gripper 0'
   Call Prepare_gripper_0
▼ Robot Program
   'Program content'
   CLOSE CONNECTION
Prepare_gripper_0
   CHECK DEVICE
   SET GRIP CONFIG
   HOME
   ENABLE
```

GRIPLINK Connect

This node establishes a connection between parameterized GRIPLINK controller and this robot controller.

Connect to the pre-configured GRIPLINK controller

IP address	Device information	
192.168.1.40	Proxy role	STAND-ALONE

<u>The Robot Program section</u>

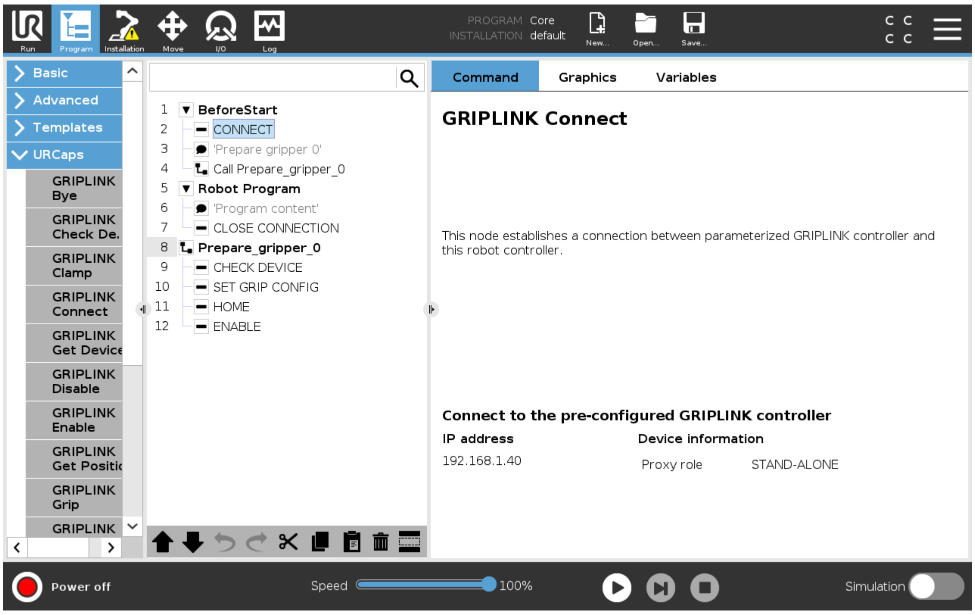
1. contains the commands for gripping and releasing
2. processes variables that are described by nodes, e.g. for status and value queries
3. of programs that do not run continuously, the „GRIPLINK Bye“ node should be called at the end of the program to ensure a clean exit from the network interface

! Correct functioning can only be guaranteed if this basic procedure is followed.

The available commands of the plugin are described in the following sections. Each command can be used both as a graphical node and as a URScript implementation.

6.2 Establishing a connection – GRIPLINK Connect

This command establishes the connection between the GRIPLINK controller and the robot controller. The IP address set on the installation page of the plug-in is displayed.



```
▼ BeforeStart
    CONNECT
    'Prepare gripper 0'
    Call Prepare_gripper_0
▼ Robot Program
    'Program content'
    CLOSE CONNECTION
Prepare_gripper_0
    CHECK DEVICE
    SET GRIP CONFIG
    HOME
    ENABLE
```

GRIPLINK Connect

This node establishes a connection between parameterized GRIPLINK controller and this robot controller.

Connect to the pre-configured GRIPLINK controller

IP address	Device information	
192.168.1.40	Proxy role	STAND-ALONE

If the GRIPLINK Controller can be reached at the IP address set during installation, additional information such as the proxy role is displayed.
If not, check the connection.

6.2.1 Command call with script code

```
GL_CONNECT(
      <GRIPLINK_IP>,
      <SOCKET_NAME>
)
```

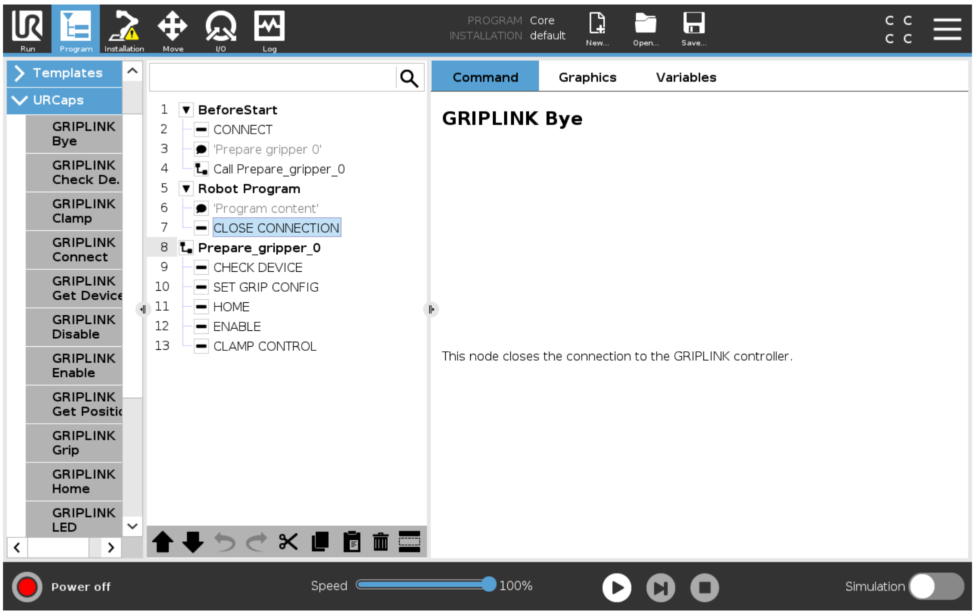
Parameters	Type	Meaning
<GRIPLINK_IP>	String	IP address of the GRIPLINK controller E.g. „192.168.1.40“
<SOCKET_NAME>	String	Name of the socket to be used, e.g. „sock_griplink“ The socket name is used in all subsequent nodes in the robot program!

Example:

```
      GL_CONNECT(„192.168.1.40“,“sock_griplink“)
```

6.3 Close connection – GRIPLINK Bye

This command disconnects the connection between the GRIPLINK Controller and the robot controller. This can be useful if the GRIPLINK controller is installed behind a changer.



```
▼ BeforeStart
    CONNECT
    'Prepare gripper 0'
    Call Prepare_gripper_0
▼ Robot Program
    'Program content'
    CLOSE CONNECTION
Prepare_gripper_0
    CHECK DEVICE
    SET GRIP CONFIG
    HOME
    ENABLE
    CLAMP CONTROL
```

GRIPLINK Bye

This node closes the connection to the GRIPLINK controller.

After disconnecting, no commands can be sent until GRIPLINK Connect is called up again!

6.3.1 Command call with script code

```
GL_BYE(
      <SOCKET_NAME>
)
```

Parameters	Type	Meaning
<SOCKET_NAME>	String	Name of the socket to be used, e.g. „sock_griplink“ The socket name is used in all subsequent nodes in the robot program!

Example:

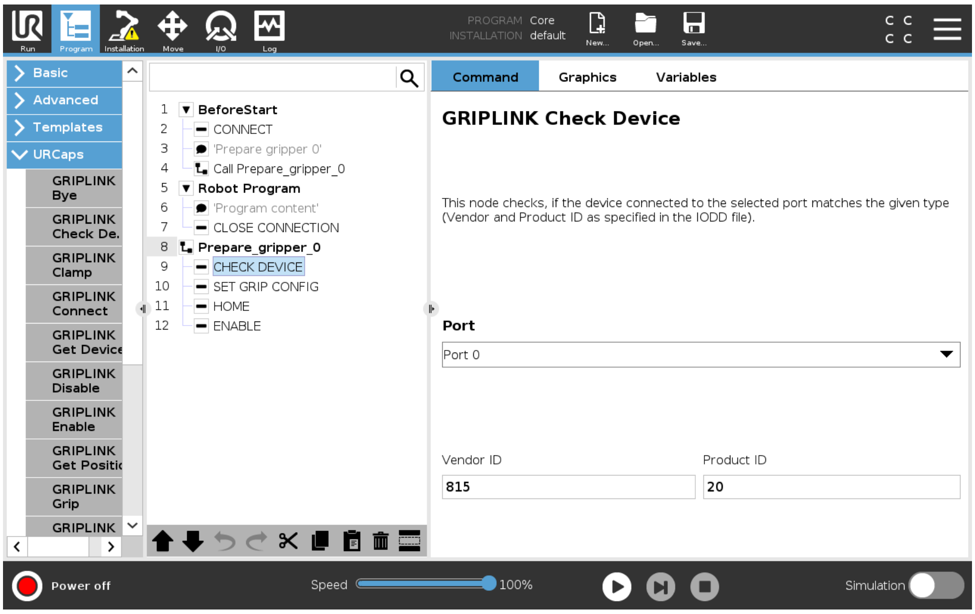
```
      GL_BYE(„sock_griplink“)
```

6.4 Checking a connected device – GRIPLINK Check Device

Before a device is used in the program, it is advisable to check whether the expected device is also connected. The „GRIPLINK Check Device“ node checks at the selected port whether the vendor and product ID (VID/PID) of the connected device match the expected values. The robot program is stopped immediately if they do not match.

Further information on the vendor and product ID can be found in the operating instructions for the respective device.

The port, vendor and product ID can be set in the settings of the „GRIPLINK Check Device“ node. In the example, an IEG 55-020 from WEISS ROBOTICS (VID 815, PID 20) is expected at port 0.



```
▼ BeforeStart
    CONNECT
    'Prepare gripper 0'
    Call Prepare_gripper_0
▼ Robot Program
    'Program content'
    CLOSE CONNECTION
Prepare_gripper_0
    CHECK DEVICE
    SET GRIP CONFIG
    HOME
    ENABLE
```

GRIPLINK Check Device

This node checks, if the device connected to the selected port matches the given type (Vendor and Product ID as specified in the IODD file).

Port: Port 0

Vendor ID: 815

Product ID: 20

6.4.1 Command call with script code

```
GL_DEVASSERT(
     <PORT>,
     <VENDOR_ID>,
     <PRODUCT_ID>,
     <SOCKET_NAME>
)
```

Parameters	Type	Meaning
<PORT>	Integer	Port index (0..31)
<VENDOR_ID>	Integer	Vendor ID of the expected device
<PRODUCT_ID>	Integer	Product ID of the expected device
<SOCKET_NAME>	String	Name of the socket to be used, e.g. „sock_griplink“ The socket name is used in all subsequent nodes in the robot program!

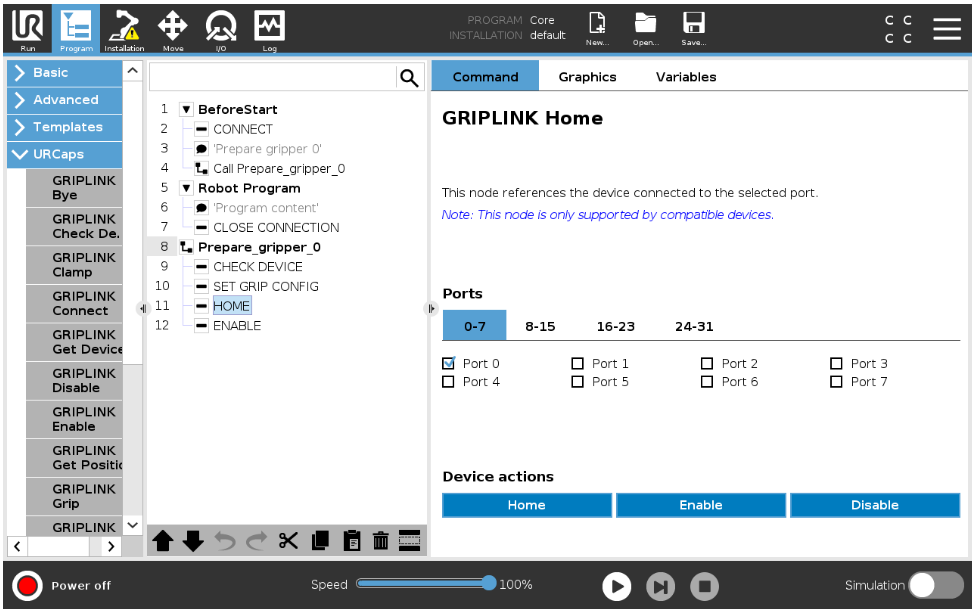
Example:

```
    # Ensure, a WEISS ROBOTICS IEG 55-020 (VID 815, PID 20) is connected
    # to port 0
    GL_DEVASSERT(0,815,22, „sock_griplink“)
```

6.5 Referencing a device – GRIPLINK Home

If necessary, a device must be referenced at the beginning. The „GRIPLINK Home“ node is used for this.

Further details can be found in the operating instructions for the respective device.



```
1  ▼ BeforeStart
2      CONNECT
3      'Prepare gripper 0'
4      Call Prepare_gripper_0
5  ▼ Robot Program
6      'Program content'
7      CLOSE CONNECTION
8    Prepare_gripper_0
9      CHECK DEVICE
10     SET GRIP CONFIG
11     HOME
12     ENABLE
```

GRIPLINK Home

This node references the device connected to the selected port.

Note: This node is only supported by compatible devices.

Ports

0-7 | 8-15 | 16-23 | 24-31

☑ Port 0 □ Port 1 □ Port 2 □ Port 3
□ Port 4 □ Port 5 □ Port 6 □ Port 7

Device actions

Home | Enable | Disable

Several gripping modules can be referenced at the same time.

At least one port must be selected!

6.5.1 Command call with script code

```
GL_HOME(
     <PORT>,
     <SOCKET_NAME>
)
```

Parameters	Type	Meaning
<PORT>	Integer	Port index (0..31)
<SOCKET_NAME>	String	Name of the socket to be used, e.g. „sock_griplink“ The socket name is used in all subsequent nodes in the robot program!

Example:

```
     # Home device connected to port 0
     GL_HOME(0, „sock_griplink")
```

6.6 Activating and deactivating the device – GRIPLINK Enable/Disable

Devices can be activated and deactivated during operation.

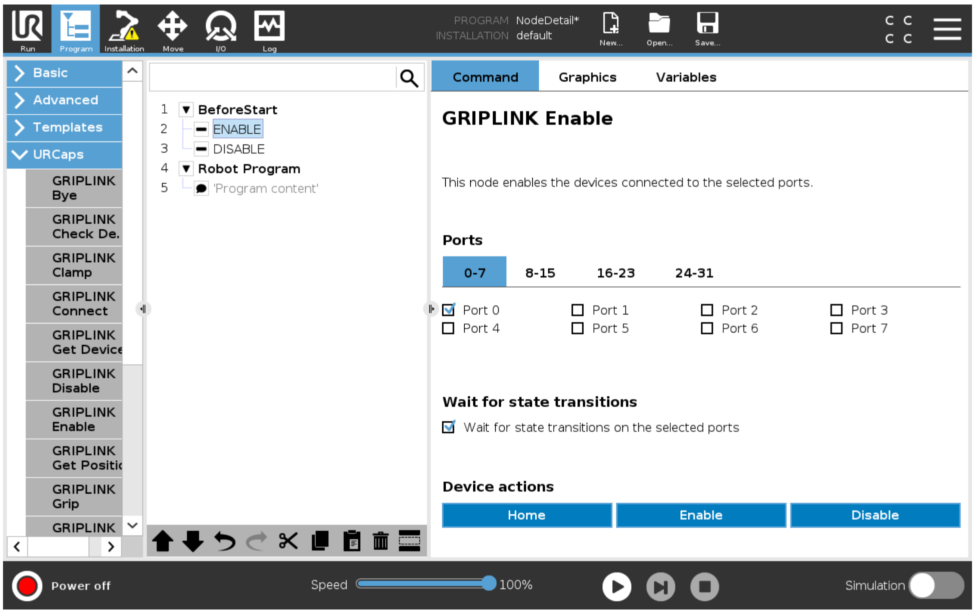
The „Wait for status transitions“ checkbox can be used to control whether the device status of the devices on the selected ports should actually change after the command has been executed.
If a device is already in the DISABLED state when the „GRIPLINK Disable“ program node is called up, this leads to a timeout and the program is aborted.

Some devices must first be activated using the Enable command before they can be used! Further details can be found in the operating instructions for the respective device driver.

Several devices can be activated/deactivated at the same time.

At least one port must be selected!

Make sure that the devices are in a permissible state if the „Wait for state transitions“ checkbox is checked!



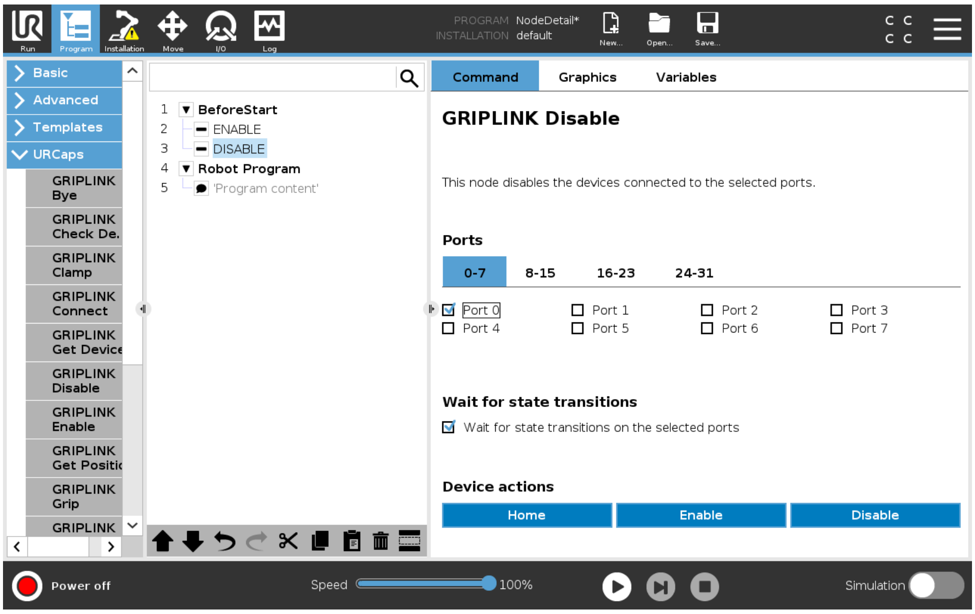
1 BeforeStart
2 ENABLE
3 DISABLE
4 Robot Program
5 'Program content'

GRIPLINK Enable

This node enables the devices connected to the selected ports.

☑ Port 0 ☐ Port 1 ☐ Port 2 ☐ Port 3
☐ Port 4 ☐ Port 5 ☐ Port 6 ☐ Port 7

Wait for state transitions
☑ Wait for state transitions on the selected ports



```
1  ▾ BeforeStart
2     ENABLE
3     DISABLE
4  ▾ Robot Program
5     'Program content'
```

GRIPLINK Disable

This node disables the devices connected to the selected ports.

Ports

☑ Port 0 □ Port 1 □ Port 2 □ Port 3
□ Port 4 □ Port 5 □ Port 6 □ Port 7

Wait for state transitions

☑ Wait for state transitions on the selected ports

Device actions

Several gripping modules can be activated/deactivated at the same time.

At least one port must be selected!

6.6.1 Command call with script code

```
GL_ENABLE(
      <PORT>,
      <WSTR_ENABLED>,
      <SOCKET_NAME>
)

GL_DISABLE(
      <PORT>,
      <WSTR_ENABLED>,
      <SOCKET_NAME>
)
```

Parameters	Type	Meaning
<PORT>	Integer	Port index (0..31)
<WSTR_ENABLED>	Boolean	„True“ to wait for the status transition of the device „False“ so as not to wait for the device to change state
<SOCKET_NAME>	String	Name of the socket to be used, e.g. „sock_griplink“ The socket name is used in all subsequent nodes in the robot program!

Example:

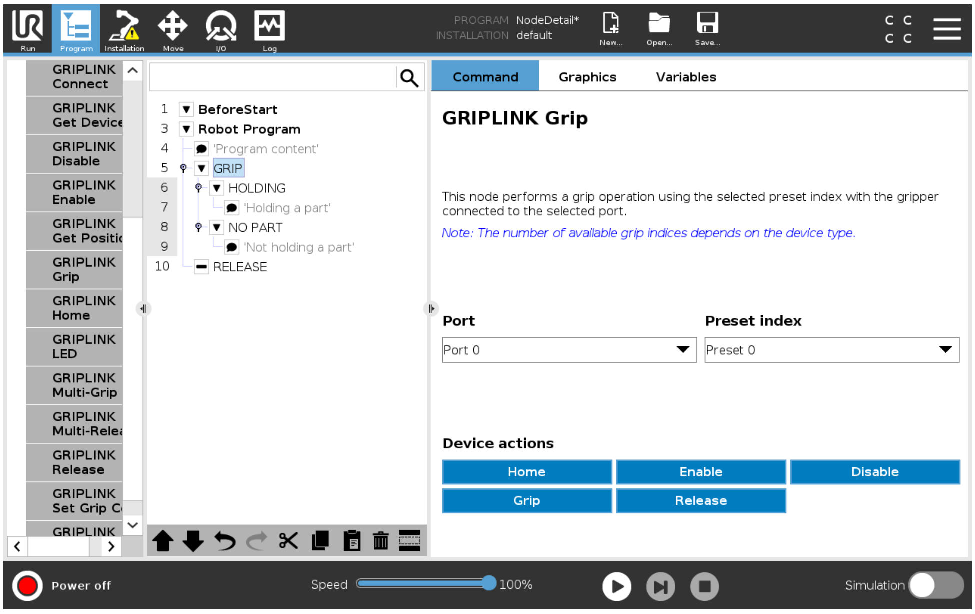
```
# Disable device connected to port 0, wait until disabled
GL_DISABLE(0,True, „sock_griplink‟)

# Enable device connected to port 0 again, wait until enabled
GL_ENABLE(0,True, „sock_griplink‟)
```

6.7 Grip and release – GRIPLINK Grip/Release

Each gripper module knows the basic commands „Grip“ and „Release“. Depending on the gripper module, up to eight freely configurable grip presets can be executed.

Further information on the grips and their parameterization can be found in the operating instructions for the respective gripper module.

In the „GRIPLINK Grip“ and „GRIPLINK Release“ nodes, the port of the desired gripper and the index of the grip can be selected



```
▼ BeforeStart
▼ Robot Program
    'Program content'
  ▼ GRIP
    ▼ HOLDING
        'Holding a part'
    ▼ NO PART
        'Not holding a part'
  RELEASE
```

GRIPLINK Grip

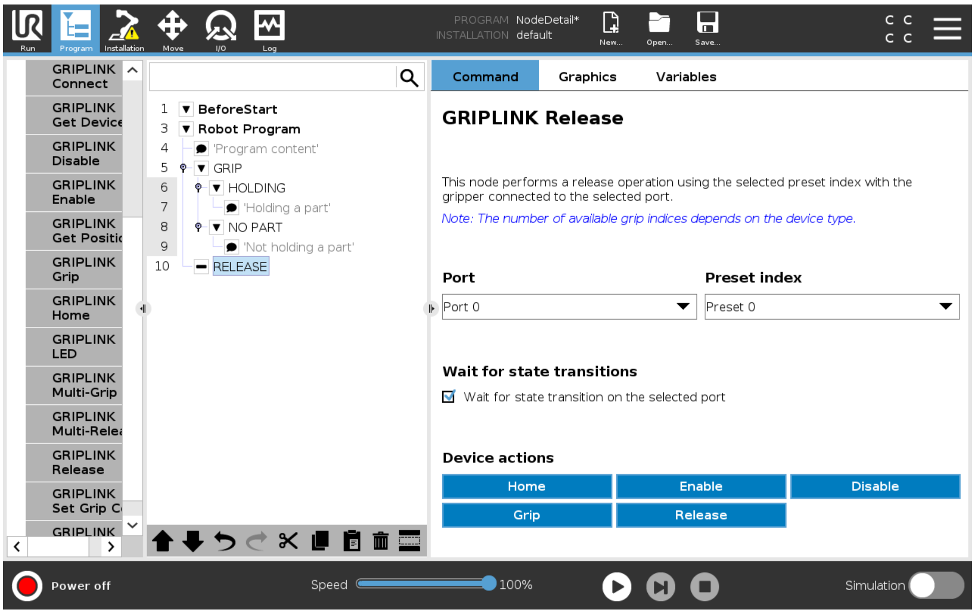
This node performs a grip operation using the selected preset index with the gripper connected to the selected port.

Note: The number of available grip indices depends on the device type.

Port: Port 0

Preset index: Preset 0

Device actions



```
▼ BeforeStart
▼ Robot Program
    'Program content'
  ▼ GRIP
    ▼ HOLDING
        'Holding a part'
    ▼ NO PART
        'Not holding a part'
  RELEASE
```

GRIPLINK Release

This node performs a release operation using the selected preset index with the gripper connected to the selected port.

Note: The number of available grip indices depends on the device type.

Port: Port 0

Preset index: Preset 0

Wait for state transitions

☑ Wait for state transition on the selected port

Device actions

The „Wait for status transitions“ checkbox in the „GRIPLINK Release“ program node can be used to control whether the system should wait until the device status of the device on the selected port has changed after the command has been executed.
If a device is already in the RELEASED state when the „GRIPLINK Release“ program node is called up, this leads to a timeout and the program is aborted.

Make sure that the devices are in a permissible state if the „Wait for state transitions“ checkbox is checked.

6.7.1 Command call with script code

```
GL_GRIP(
     <PORT>,
     <PRESET_INDEX>,
     <WSTR_ENABLED>,
     <SOCKET_NAME>
)
```

Parameters	Type	Meaning
<PORT>	Integer	Port index (0..31)
<PRESET_INDEX>	Integer	Preset index (0..7)
<WSTR_ENABLED>	Boolean	„True“ to wait for the status transition of the device „False“ so as not to wait for the device to change state
<SOCKET_NAME>	String	Name of the socket to be used, e.g. „sock_griplink“ The socket name is used in all subsequent nodes in the robot program!

Example:

```
# Grip with gripper connected to port 0 and preset index 3
# Wait until gripper state changed
GL_GRIP(0,3,True, „sock_griplink“)

# ...
# Move part to place position
# ...

# Release part again
GL_RELEASE(0,3,True, „sock_griplink“)
```

6.7.2 Evaluation of the gripping state

After gripping, the „GRIPLINK Grip“ node recognizes whether the gripper has gripped a component or not. Depending on this, the child nodes are then executed under „HOLDING“ (component was gripped) or under „NO PART“ (component was not gripped).

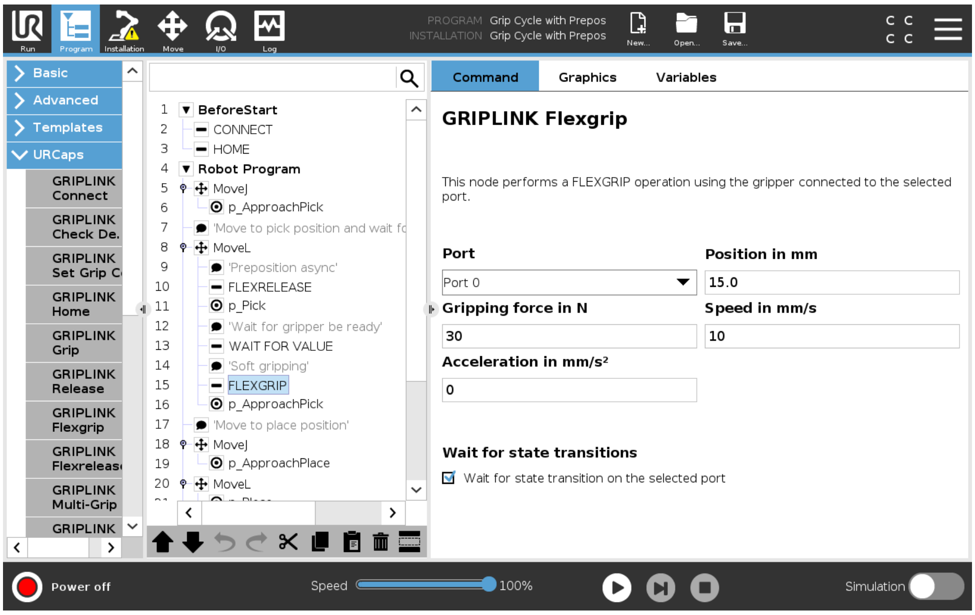
6.8 Flexible gripping and releasing – GRIPLINK Flexgrip/Flexrelease

The gripper modules in the WPG series support flexible gripping and releasing. The position, gripping force and movement parameters can be specified independently of presets. This enables fast pre-positioning and gentle gripping, reducing cycle times and protecting gripped parts.

☞ Further information on the movement parameters can be found in the operating instructions for the respective gripping module.

The port of the desired gripper can be selected in the „GRIPLINK Flexgrip“ program node. The target position is specified in millimetres. The gripping force is set in Newtons. The speed in mm/s and the acceleration in mm/s² can be specified as movement parameters.

The „Wait for state transitions“ checkbox can be used to control whether the system should wait after executing the command until the device state of the gripper at the selected port has changed.



```
▼ BeforeStart
    CONNECT
    HOME
▼ Robot Program
    MoveJ
        p_ApproachPick
    'Move to pick position and wait fo
    MoveL
        'Preposition async'
        FLEXRELEASE
        p_Pick
        'Wait for gripper be ready'
        WAIT FOR VALUE
        'Soft gripping'
        FLEXGRIP
        p_ApproachPick
    'Move to place position'
    MoveJ
        p_ApproachPlace
    MoveL
```

GRIPLINK Flexgrip

This node performs a FLEXGRIP operation using the gripper connected to the selected port.

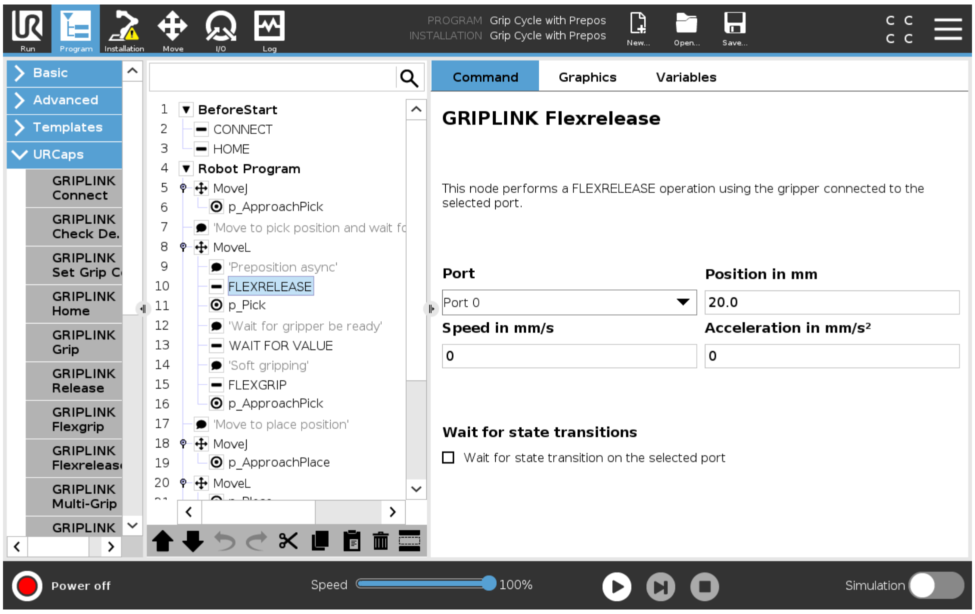
Port	Port 0
Position in mm	15.0
Gripping force in N	30
Speed in mm/s	10
Acceleration in mm/s²	0

Wait for state transitions

☑ Wait for state transition on the selected port

The port of the desired gripper can be selected in the „GRIPLINK Flexrelease“ program node. The target position is specified in millimeters. The speed in mm/s and the acceleration in mm/s² can be specified as movement parameters.

The „Wait for state transitions“ checkbox in the „GRIPLINK Release“ program node can be used to control whether the system should wait until the gripper has reached the target position at the selected port after the command has been executed.



PROGRAM Grip Cycle with Prepos
INSTALLATION Grip Cycle with Prepos

```
BeforeStart
  CONNECT
  HOME
Robot Program
  MoveJ
    p_ApproachPick
  'Move to pick position and wait fo
  MoveL
    'Preposition async'
    FLEXRELEASE
    p_Pick
    'Wait for gripper be ready'
    WAIT FOR VALUE
    'Soft gripping'
    FLEXGRIP
    p_ApproachPick
  'Move to place position'
  MoveJ
    p_ApproachPlace
  MoveL
```

GRIPLINK Flexrelease

This node performs a FLEXRELEASE operation using the gripper connected to the selected port.

Port: Port 0
Position in mm: 20.0
Speed in mm/s: 0
Acceleration in mm/s²: 0

Wait for state transitions

☐ Wait for state transition on the selected port

6.8.1 Command call with script code

```
GL_FLEXGRIP(
     <PORT>,
     <POSITION>,
     <GRIPPING_FORCE>,
     <SPEED>,
     <ACCELERATION>,
     <WSTR_ENABLED>,
     <SOCKET_NAME>
)

GL_FLEXRELEASE(
     <PORT>,
     <POSITION>,
     <SPEED>,
     <ACCELERATION>,
     <WSTR_ENABLED>,
     <SOCKET_NAME>
)
```

Parameters	Type	Meaning
<PORT>	Integer	Port index (0..31)
<POSITION>	Double	Target position in 1/1000 mm
<GRIPPING_FORCE>	Integer	Gripping force in 1/1000 N (only for GL_FLEXGRIP)
<SPEED>	Integer	Speed in 1/1000 mm/s
<ACCELERATION>	Integer	Acceleration in 1/1000 mm/s^2

<WSTR_ENABLED>	Boolean	GL_FLEXGRIP „True“ to wait for the status transition of the device „False“ to continue directly to the next node GL_FLEXRELEASE „True“ to wait for the target position to be reached „False“ to continue directly to the next node
<SOCKET_NAME>	String	Name of the socket to be used, e.g. „sock_griplink“ The socket name is used in all subsequent nodes in the robot program!

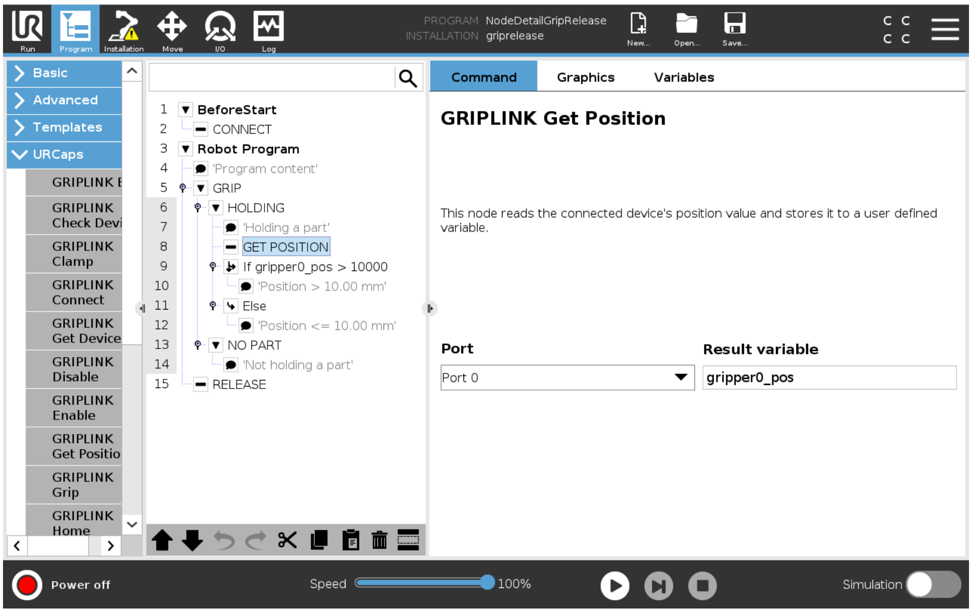
Example:

```
# Preposition with WPG at port 0 to 50.0 mm with optimum speed
# and acceleration
# Wait for target position reached
GL_FLEXRELEASE(0,50000,0,0,True, „sock_griplink“)

# Grip part with No Part Limit 45.0 mm with 100 N and optimum speed
# and acceleration
# Wait for state transition
GL_FLEXGRIP(0,45000,100000,0,0,True,“sock_griplink“)
```

6.9 Evaluation of the finger position – GRIPLINK Get Position

The „GRIPLINK Get Position“ node can be used to read out the position of the finger jaws, for example to check the gripped component based on its size. This writes the current position of the finger jaws of the selected gripping module in thousandths of a millimeter to a variable that can be freely defined by the user.

When defining the variable, make sure that it is only used for evaluating the gripping width of the gripping module!



```
BeforeStart
    CONNECT
Robot Program
    'Program content'
    GRIP
        HOLDING
            'Holding a part'
            GET POSITION
            If gripper0_pos > 10000
                'Position > 10.00 mm'
            Else
                'Position <= 10.00 mm'
        NO PART
            'Not holding a part'
    RELEASE
```

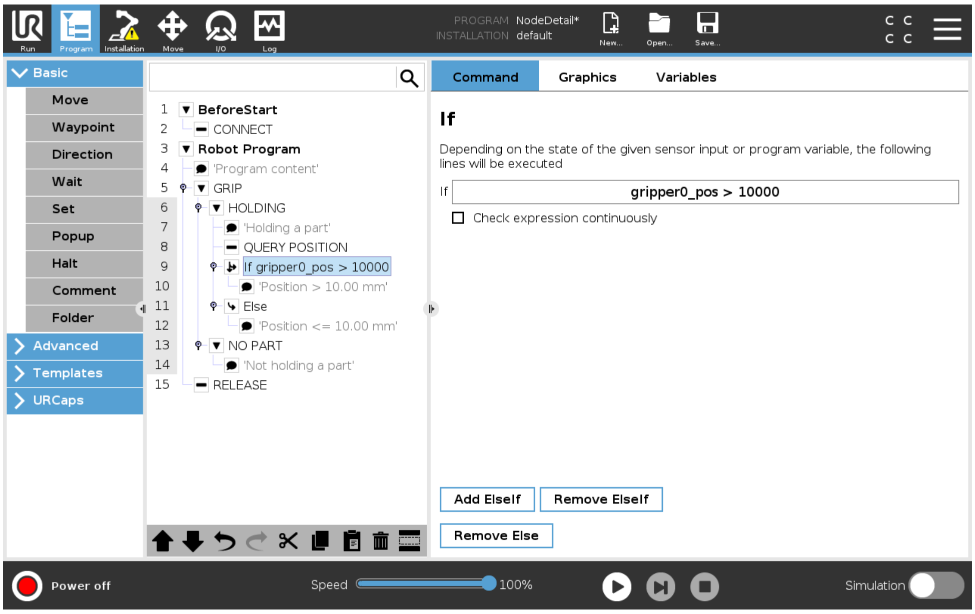
GRIPLINK Get Position

This node reads the connected device's position value and stores it to a user defined variable.

Port: Port 0

Result variable: gripper0_pos

The finger position can be processed using simple if-else queries:



```
BeforeStart
    CONNECT
Robot Program
    'Program content'
    GRIP
        HOLDING
            'Holding a part'
            QUERY POSITION
            If gripper0_pos > 10000
                'Position > 10.00 mm'
            Else
                'Position <= 10.00 mm'
        NO PART
            'Not holding a part'
    RELEASE
```

If

Depending on the state of the given sensor input or program variable, the following lines will be executed

If gripper0_pos > 10000

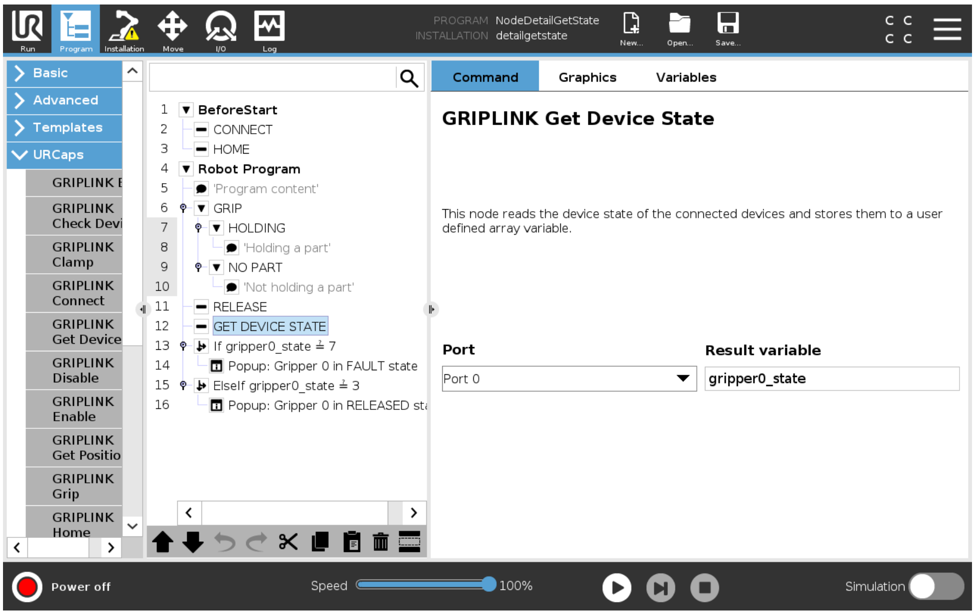
□ Check expression continuously

6.9.1 Command call with script code

The GL_VALUE function is used to query the position. The value 0 must be used as the value index (see section 6.12.1).

6.10 Status query – GRIPLINK Devstate

The „GRIPLINK Devstate“ node is used to query the status of a device. This writes the read status as a numerical value to the variable defined by the user.


PROGRAM NodeDetailGetState
INSTALLATION detailgetstate

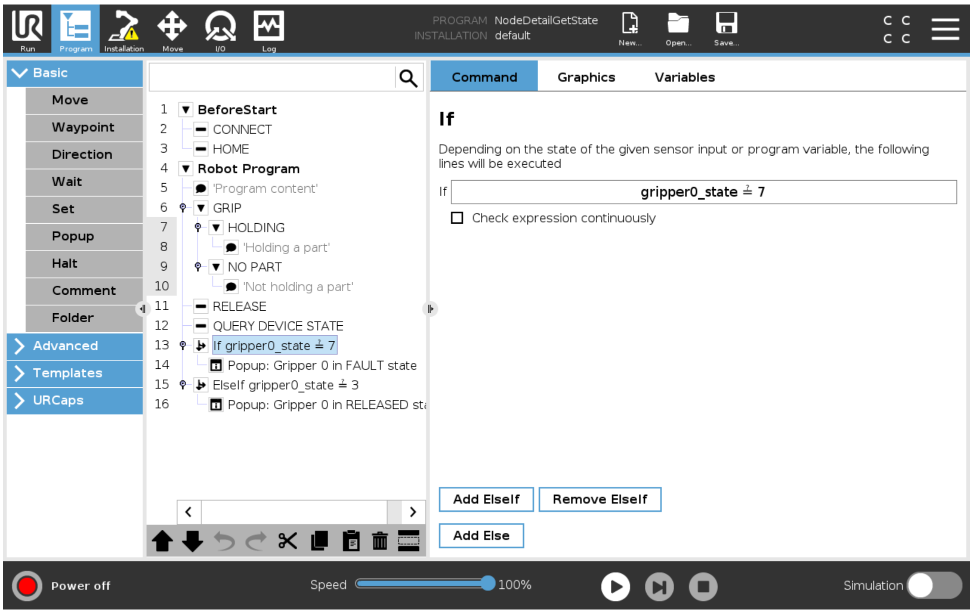
```
BeforeStart
  CONNECT
  HOME
Robot Program
  'Program content'
  GRIP
    HOLDING
      'Holding a part'
    NO PART
      'Not holding a part'
  RELEASE
  GET DEVICE STATE
  If gripper0_state ≟ 7
    Popup: Gripper 0 in FAULT state
  ElseIf gripper0_state ≟ 3
    Popup: Gripper 0 in RELEASED sta
```

GRIPLINK Get Device State

This node reads the device state of the connected devices and stores them to a user defined array variable.

Port: Port 0
Result variable: gripper0_state

The state can now be processed using if-else constructs. In the following example, the state of a gripper at port 0 is processed:


PROGRAM NodeDetailGetState
INSTALLATION default

```
BeforeStart
  CONNECT
  HOME
Robot Program
  'Program content'
  GRIP
    HOLDING
      'Holding a part'
    NO PART
      'Not holding a part'
  RELEASE
  QUERY DEVICE STATE
  If gripper0_state ≟ 7
    Popup: Gripper 0 in FAULT state
  ElseIf gripper0_state ≟ 3
    Popup: Gripper 0 in RELEASED sta
```

If

Depending on the state of the given sensor input or program variable, the following lines will be executed

If gripper0_state ≟ 7

□ Check expression continuously

The possible device states are listed in Appendix A listed!

6.10.1 Command call with script code

```
<RETURN_VARIABLE> = GL_DEVSTATE(
     <PORT>,
     <SOCKET_NAME>
)
```

Parameters	Type	Meaning
<PORT>	Integer	Port index (0..31)
<SOCKET_NAME>	String	Name of the socket to be used, e.g. „sock_griplink“ The socket name is used in all subsequent nodes in the robot program!

Type Return value	Meaning
Integer	Device status (see Appendix A)

Predefined constants can be used for the device statuses.

Example:

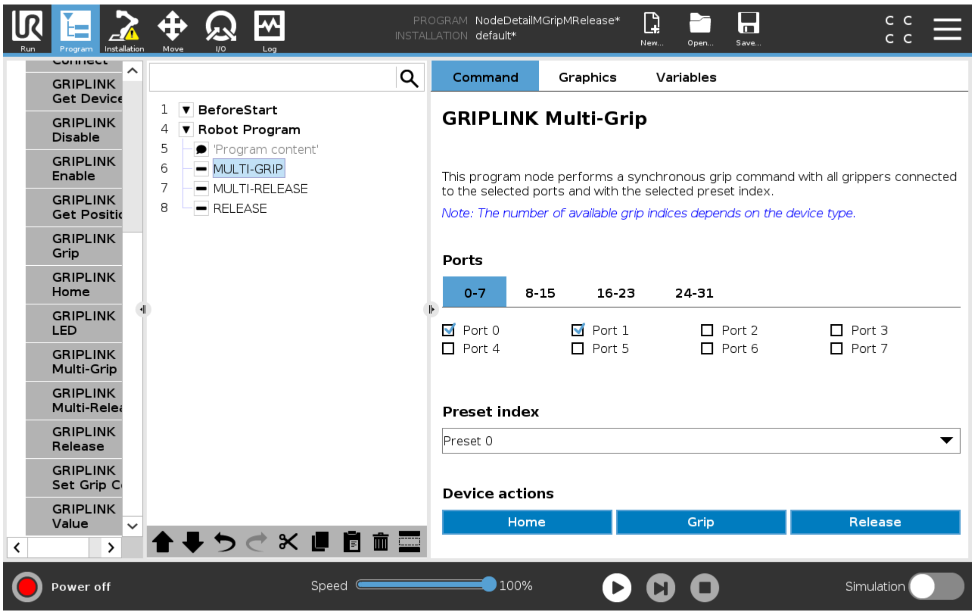
```
     # Grip with gripper connected to port 0 and preset index 3
     # Wait until gripper state changed
     GL_GRIP(0,3,True, „sock_griplink‟)

     # When state is HOLDING, move to place position
     gripper0_state = GL_DEVSTATE(0, „sock_griplink‟)
     if (gripper0_state == S_HOLDING):
          # Move to place position
          # ...
     else:
          # Process other states here
     end
```

6.11 Gripping and releasing with multiple grippers – GRIPLINK Multi Grip/Release

The „GRIPLINK Multi-Grip“ and „GRIPLINK Multi-Release“ nodes can be used to execute grip and release commands synchronously with several grippers. The ports of the desired grippers and the uniform index of the grip preset can be selected.

The „Wait for state transitions“ checkbox in the „GRIPLINK Multi-Release“ program node can be used to control whether the system should wait until the device state of all devices on the selected ports has changed after the command has been executed.
If the status of a device does not change within a specified time, this leads to a timeout and the program is aborted.
The „GRIPLINK Multi-Grip“ program node always waits for a state change.

Make sure that the devices are in a permissible state if the „Wait for state transitions“ checkbox is set in the „GRIPLINK Multi-Release“ program node!



1 ▼ BeforeStart
4 ▼ Robot Program
5 'Program content'
6 MULTI-GRIP
7 MULTI-RELEASE
8 RELEASE

GRIPLINK Multi-Grip

This program node performs a synchronous grip command with all grippers connected to the selected ports and with the selected preset index.

Note: The number of available grip indices depends on the device type.

Ports

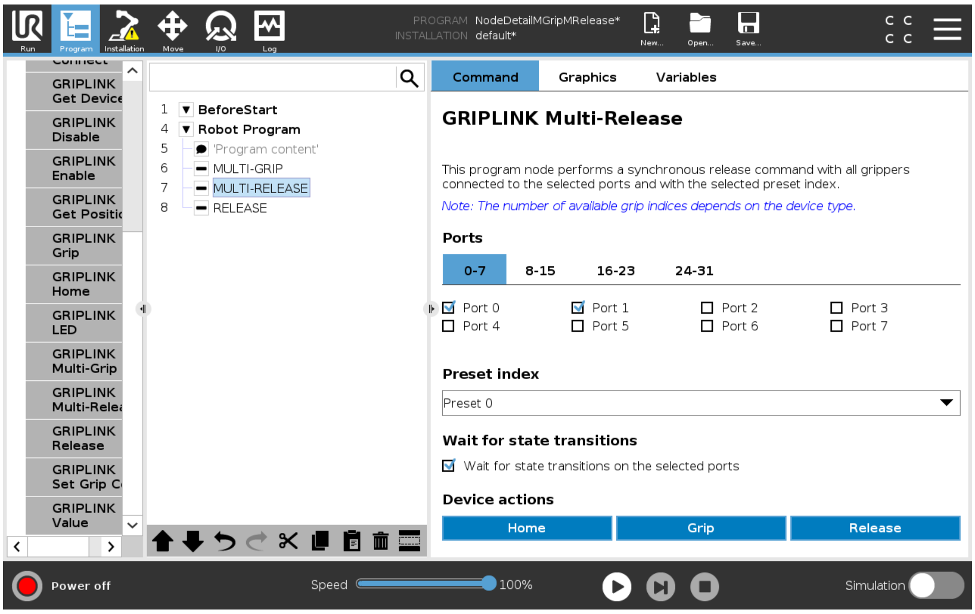
0-7 | 8-15 | 16-23 | 24-31

☑ Port 0 ☑ Port 1 ☐ Port 2 ☐ Port 3
☐ Port 4 ☐ Port 5 ☐ Port 6 ☐ Port 7

Preset index

Preset 0

Device actions



PROGRAM NodeDetailMGripMRelease*
INSTALLATION default*

```
1  ▼ BeforeStart
4  ▼ Robot Program
5       'Program content'
6       MULTI-GRIP
7       MULTI-RELEASE
8       RELEASE
```

GRIPLINK Multi-Release

This program node performs a synchronous release command with all grippers connected to the selected ports and with the selected preset index.

Note: The number of available grip indices depends on the device type.

Ports

0-7 8-15 16-23 24-31

☑ Port 0 ☑ Port 1 □ Port 2 □ Port 3
□ Port 4 □ Port 5 □ Port 6 □ Port 7

Preset index

Preset 0

Wait for state transitions

☑ Wait for state transitions on the selected ports

Device actions

Home Grip Release

All selected grippers execute the grip/release command with the same preset index. Make sure in advance that the preset configuration is set correctly!

6.11.1 Command call with script code

```
GL_MGRIP(
	<PORTS>,
	<PRESET_INDEX>,
	<SOCKET_NAME>
)

GL_MRELEASE(
	<PORTS>,
	<PRESET_INDEX>,
	<SOCKET_NAME>
)
```

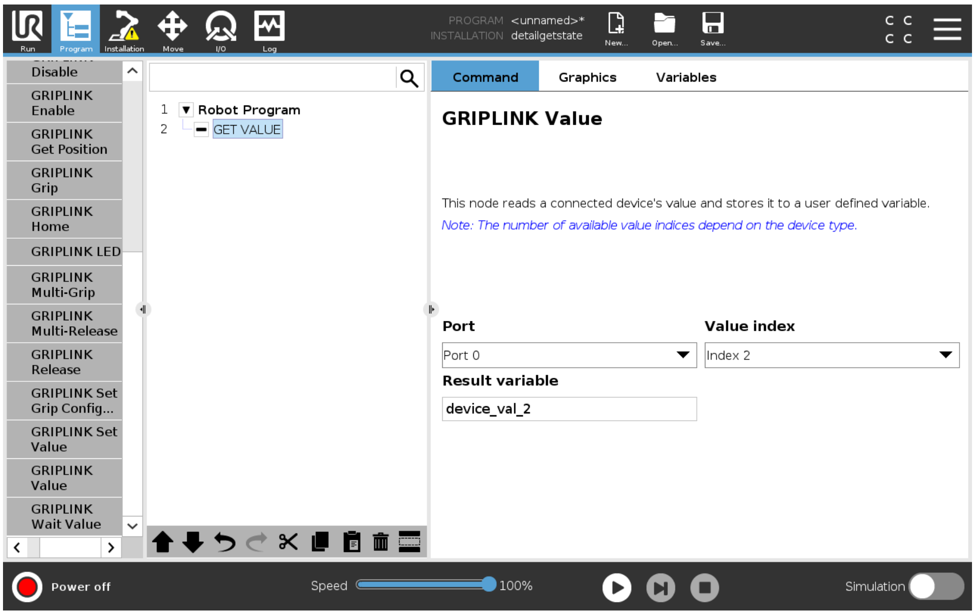
Parameters	Type	Meaning
<PORTS>	Integer List	Selected ports as a list with values 0 or 1 List contains at least one and a maximum of 31 elements Each element of the list corresponds to the port of the respective list index A port is selected if the corresponding element in the list contains the value 1. Example: Grippers on ports 0, 2 and 3 should be used →List [1, 0, 1, 1]
<PRESET_INDEX>	Integer	Preset index (0..7)
<WSTR_ENABLED>	Boolean	„True“ to wait for status transition of the devices „False“ so as not to wait for the status transition of the devices
<SOCKET_NAME>	String	Name of the socket to be used, e.g. „sock_griplink“ The socket name is used in all subsequent nodes in the robot program!

Example:

```
	# Grip with the grippers connected to ports 0 and 2. Use preset 5.
	GL_MGRIP(5,[1,0,1], „sock_griplink“)

	# Release with the grippers connected to ports 0 and 2. Use preset 5.
	GL_MRELEASE(5,[1,0,1], „sock_griplink“)
```

6.12 Reading device values – GRIPLINK Value

Devices such as sensors provide data in the form of indexed values. These values can be read out via the „GRIPLINK Value“ node. Depending on the device, different numbers of values can be read out.



GRIPLINK Value

This node reads a connected device's value and stores it to a user defined variable.

Note: The number of available value indices depend on the device type.

Port: Port 0

Value index: Index 2

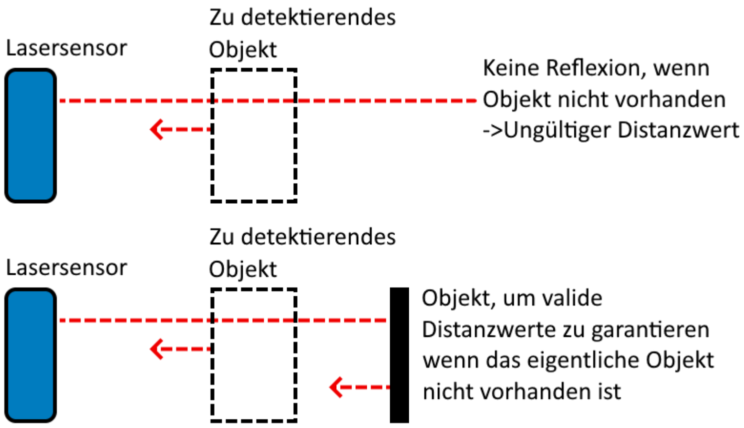
Result variable: device_val_2

A maximum of eight readable values are possible, although there may be fewer depending on the device.

Therefore, make sure that a valid value index is always set. Further information can be found in the instructions for the respective device driver.

The value 2147483647 is used in GRIPLINK, for example, for sensors whose measured variable is outside the measuring range.

To prevent errors in the program sequence, it should be ensured that valid values (< 2147483647) are always read from the respective device!
This can be ensured, for example, by a distance sensor always detecting an object and thus always providing a valid distance instead of the zero value:


Lasersensor
Zu detektierendes Objekt
Keine Reflexion, wenn Objekt nicht vorhanden ->Ungültiger Distanzwert
Lasersensor
Zu detektierendes Objekt
Objekt, um valide Distanzwerte zu garantieren wenn das eigentliche Objekt nicht vorhanden ist

6.12.1 Command call with script code

```
<RETURN_VARIABLE> = GL_VALUE(
        <PORT>,
        <VALUE_INDEX>,
        <SOCKET_NAME>
)
```

Parameters	Type	Meaning
<PORT>	Integer	Port index (0..31)
<VALUE_INDEX>	Integer	Value index (0..7)
<SOCKET_NAME>	String	Name of the socket to be used, e.g. „sock_griplink“ The socket name is used in all subsequent nodes in the robot program!

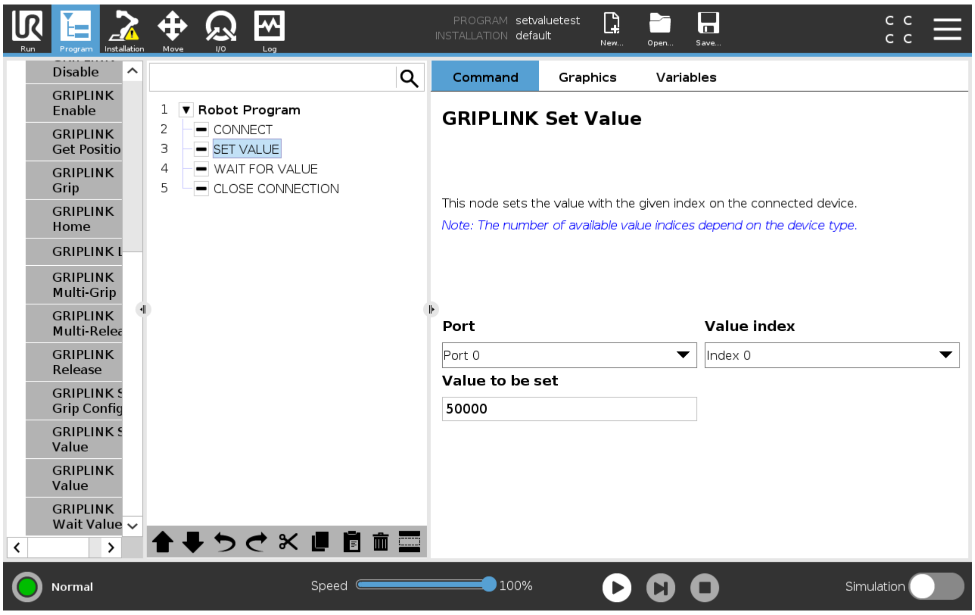
Type Return value	Meaning
Integer	Value of the device value read out

Example:

```
        # Get value 0 (position) of gripper IEG 55-020 connected to port 0
        gripper0_position = GL_VALUE(0,0, „sock_griplink“)
```

6.13 Set device value – GRIPLINK Set Value

Some devices support the setting of values. The „GRIPLINK Set Value“ node is used for this.



Robot Program
CONNECT
SET VALUE
WAIT FOR VALUE
CLOSE CONNECTION

GRIPLINK Set Value

This node sets the value with the given index on the connected device.

Note: The number of available value indices depend on the device type.

Port: Port 0

Value index: Index 0

Value to be set: 50000

This function is not available on all devices.

6.13.1 Command call with script code

```
GL_SETVAL(
    <PORT>,
    <VALUE_INDEX>,
    <VALUE>,
    <SOCKET_NAME>
)
```

Parameters	Type	Meaning
<PORT>	Integer	Port index (0..31)
<VALUE_INDEX>	Integer	Value index (0..7) Depending on the device, fewer than 8 values may be available!
<VALUE>	Integer	Value to be set Scaled with 1/1000
<SOCKET_NAME>	String	Name of the socket to be used, e.g. „sock_griplink“ The socket name is used in all subsequent nodes in the robot program!

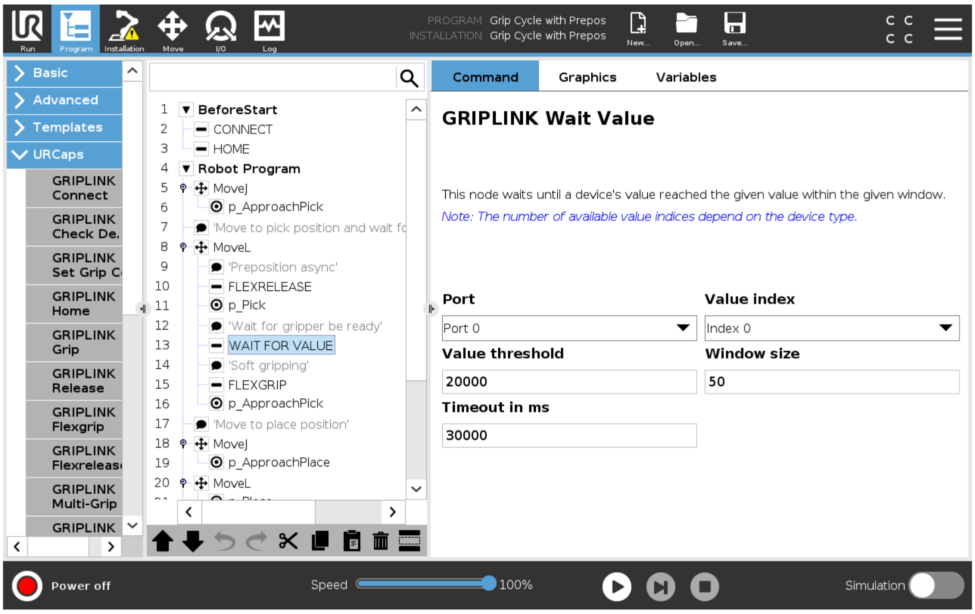
Example:

```
# Set device value 2 of device connected to port 3 to value 12
# Value 12 scaled with 1/1000 => 12,000
GL_SETVAL(3,2,12000,“sock_griplink“)
```

6.14 Wait for device value – GRIPLINK Wait Value

After setting a device value, it may be necessary to wait until the value has been reached. For actuators, for example, this can be a target position. This behavior can be mapped with the „GRIPLINK Wait Value“ node.

The system waits until the device value enters the value range (window) around the target value (threshold value) within the specified time. If the window is not reached within a certain time, a timeout error is generated.



```
▼ BeforeStart
    CONNECT
    HOME
▼ Robot Program
    MoveJ
      p_ApproachPick
    'Move to pick position and wait fo
    MoveL
      'Preposition async'
      FLEXRELEASE
      p_Pick
      'Wait for gripper be ready'
      WAIT FOR VALUE
      'Soft gripping'
      FLEXGRIP
      p_ApproachPick
    'Move to place position'
    MoveJ
      p_ApproachPlace
    MoveL
```

GRIPLINK Wait Value

This node waits until a device's value reached the given value within the given window.

Note: The number of available value indices depend on the device type.

Port: Port 0

Value index: Index 0

Value threshold: 20000

Window size: 50

Timeout in ms: 30000

This function is not available on all devices.

The window size should be selected in advance depending on the rate of change of the device value in order to enable reliable and accurate detection!

6.14.1 Command call with script code

```
GL_WAITVAL(
     <PORT>,
     <VALUE_INDEX>,
     <VALUE_THRESHOLD>,
     <WINDOW_SIZE>,
     <TIMEOUT_MS>,
     <SOCKET_NAME>
)
```

Parameters	Type	Meaning
<PORT>	Integer	Port index (0..31)
<VALUE_INDEX>	Integer	Value index (0..7) Depending on the device, fewer than 8 values may be available!
<VALUE_THRESHOLD>	Integer	Threshold value to be reached Scaled with 1/1000
<WINDOW_SIZE>	Integer	Window size that specifies the value range centered around the threshold value within which the WAITVAL command terminates without error Scaled with 1/1000
<TIMEOUT_MS>	Integer	Maximum duration in ms for which the system waits for the target value to be reached
<SOCKET_NAME>	String	Name of the socket to be used, e.g. „sock_griplink“ The socket name is used in all subsequent nodes in the robot program!

<u>Example:</u>

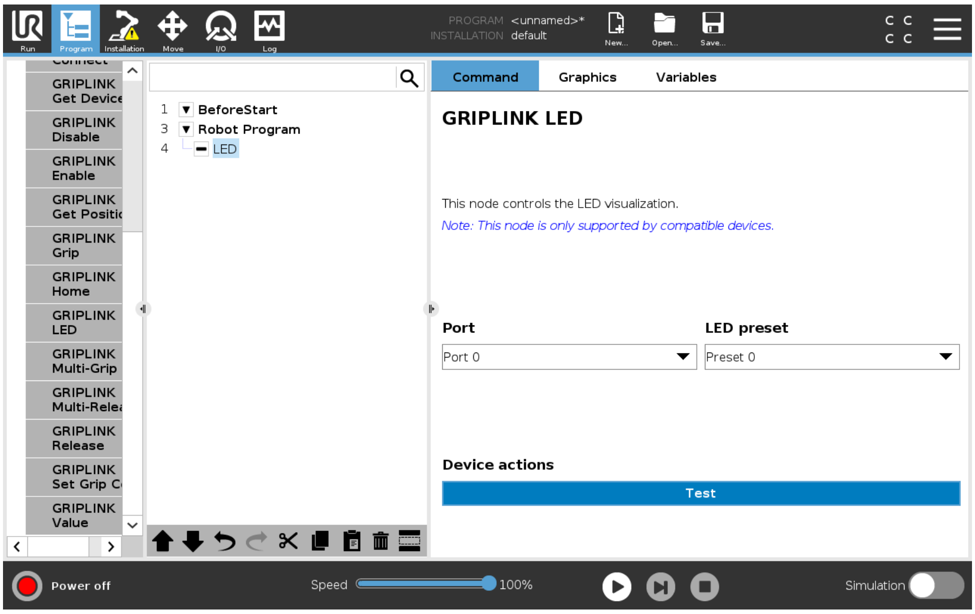
```
     # Wait for sensor connected to port 0 to reach a value of 4000 +/- 500
     # within at most 15 seconds
     # Value threshold 4000 scaled with 1/1000 => 4,000,000
     # Window size 500 scaled with 1/1000 => 500,000
     # 15 s => 15,000 ms
     GL_WAITVAL(0,4000000,500000,15000,“sock_griplink“)
```

6.15 Control of the LED illuminated ring – GRIPLINK LED

The gripper modules in the CRG series from WEISS ROBOTICS have an LED light ring for visualizing various operating states. They can be controlled as independent instructions. All CRG gripper modules each have eight configurable presets that can be started via the instruction.

Further information on LED visualization and its parameterization can be found in the operating instructions for the CRG gripping modules.

The port of the desired gripper and the index of the preset can be selected via the settings of the „GRIPLINK LED“ node.



1 BeforeStart
3 Robot Program
4 LED

GRIPLINK LED

This node controls the LED visualization.

Note: This node is only supported by compatible devices.

Port: Port 0

LED preset: Preset 0

Device actions

The LED function is only available on compatible devices!

6.15.1 Command call with script code

```
GL_LED(
      <PORT>,
      <PRESET_INDEX>,
      <SOCKET_NAME>
)
```

Parameters	Type	Meaning
<PORT>	Integer	Port index (0..31)
<PRESET_INDEX>	Integer	Preset index (0..7)
<SOCKET_NAME>	String	Name of the socket to be used, e.g. „sock_griplink“ The socket name is used in all subsequent nodes in the robot program!

Example:

```
      # Set LED pattern preset 4 of CRG 200-085 connected to port 0
      GL_LED(0,4, „sock_griplink“)
```

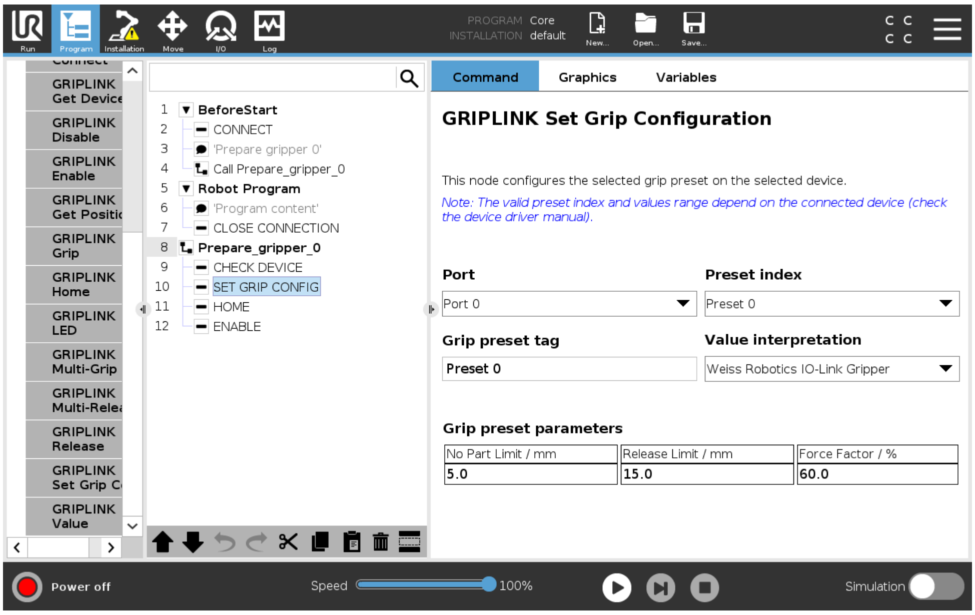
6.16 Configuring a grip preset – GRIPLINK Set Grip Config

Depending on the gripper module, up to eight freely configurable grips can be executed. The „GRIPLINK Set Grip Config“ node can be used to adjust the grip parameters during operation, for example to initialize a gripping module at the start of a program or to react flexibly to changes in the sequence.

Further information on the gripping parameters, in particular the limitations, can be found in the operating instructions for the respective gripping modules.

The port of the desired gripper and the index of the preset can be selected in the „GRIPLINK Set Grip Config“ node overview.
The „Value Interpretation“ drop-down field can be used to adjust the display of the values, depending on the meaning of the individual parameters. Non-relevant parameters are hidden for certain devices. No part limit and release limit as well as the percentage gripping force (force factor) can be specified for gripping modules from WEISS ROBOTICS

For gripping modules from WEISS ROBOTICS, the „Weiss Robotics“ parameter template must be selected so that the values entered are checked and displayed correctly.



```
▼ BeforeStart
  CONNECT
  'Prepare gripper 0'
  Call Prepare_gripper_0
▼ Robot Program
  'Program content'
  CLOSE CONNECTION
Prepare_gripper_0
  CHECK DEVICE
  SET GRIP CONFIG
  HOME
  ENABLE
```

GRIPLINK Set Grip Configuration

This node configures the selected grip preset on the selected device.

Note: The valid preset index and values range depend on the connected device (check the device driver manual).

Port: Port 0

Preset index: Preset 0

Grip preset tag: Preset 0

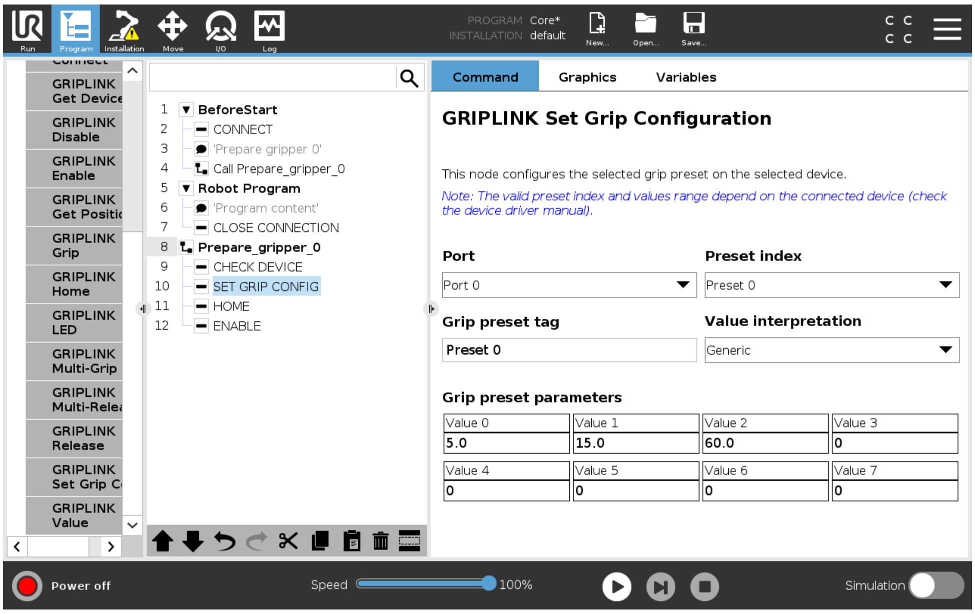
Value interpretation: Weiss Robotics IO-Link Gripper

Grip preset parameters

No Part Limit / mm	Release Limit / mm	Force Factor / %
5.0	15.0	60.0

Please note that, depending on the gripper module, not all eight configurable handles are always available!

The generic view can be used for other devices. All eight available parameters are visible here, even if not all of them are used by the device.



```
▼ BeforeStart
    CONNECT
    'Prepare gripper 0'
    Call Prepare_gripper_0
▼ Robot Program
    'Program content'
    CLOSE CONNECTION
Prepare_gripper_0
    CHECK DEVICE
    SET GRIP CONFIG
    HOME
    ENABLE
```

GRIPLINK Set Grip Configuration

This node configures the selected grip preset on the selected device.

Note: The valid preset index and values range depend on the connected device (check the device driver manual).

Port: Port 0

Preset index: Preset 0

Grip preset tag: Preset 0

Value interpretation: Generic

Grip preset parameters

Value 0	Value 1	Value 2	Value 3
5.0	15.0	60.0	0
Value 4	Value 5	Value 6	Value 7
0	0	0	0

6.16.1 Command call with script code

```
GL_SETGRIPCFG(
      <PORT>,
      <PRESET_INDEX>,
      <PRESET_PARAMS>,
      <SOCKET_NAME>
)
```

Parameters	Type	Meaning
<PORT>	Integer	Port index (0..31)
<PRESET_INDEX>	Integer	Preset index (0..7)
<PRESET_PARAMS>	Integer List	Selected ports as a list with scaled values List contains at least one and a maximum of 8 elements
<SOCKET_NAME>	String	Name of the socket to be used, e.g. „sock_griplink“ The socket name is used in all subsequent nodes in the robot program!

Example:

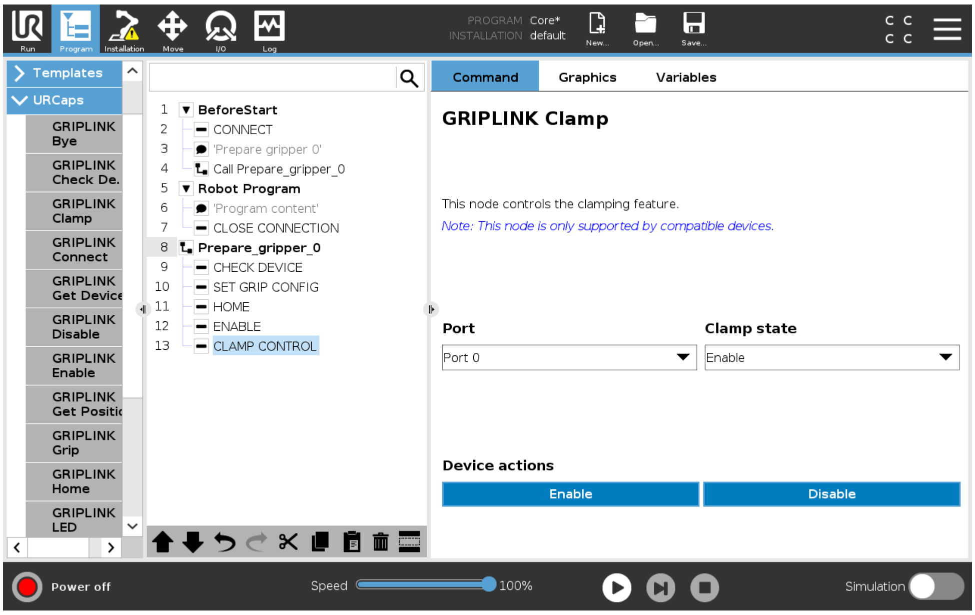
```
    # Prepare preset parameters for params 0, 1, and 2
    preset_params = [10000,20000,30000]

    # Set grip preset 3 of device connected to port 0
    GL_SETGRIPCFG(0,3,preset_params, „sock_griplink“)
```

6.17 Control mechanical clamping – GRIPLINK Clamp

The CRG series grippers from WEISS ROBOTICS are equipped with the PERMAGRIP feature. This makes it possible to switch off the motor control during prolonged gripping of components and still hold the component securely.

The “GRIPLINK Clamp“ node is used to activate this clamp.

Further information on mechanical clamping can be found in the operating instructions for the respective gripping module.



```
▼ BeforeStart
    CONNECT
    'Prepare gripper 0'
    Call Prepare_gripper_0
▼ Robot Program
    'Program content'
    CLOSE CONNECTION
Prepare_gripper_0
    CHECK DEVICE
    SET GRIP CONFIG
    HOME
    ENABLE
    CLAMP CONTROL
```

GRIPLINK Clamp

This node controls the clamping feature.

Note: This node is only supported by compatible devices.

Port: Port 0

Clamp state: Enable

Device actions

6.17.1 Command call with script code

```
GL_CLAMP(
      <PORT>,
      <CLAMP_STATE>,
      <SOCKET_NAME>
)
```

Parameters	Type	Meaning
<PORT>	Integer	Port index (0..31)
<CLAMP_STATE>	Boolean	„True“ to activate clamping „False“ to deactivate clamping
<SOCKET_NAME>	String	Name of the socket to be used, e.g. „sock_griplink“ The socket name is used in all subsequent nodes in the robot program!

Example:

```
# Enable clamping of CRG 200-085 at port 0
GL_CLAMP(0,True, „sock_griplink“)

# Grip with CRG 200-085 at port 0
GL_GRIP(0,0, „sock_griplink“)

# Hold part for longer time
#...
```

Anhang A Device status

The following table lists the possible status values of connected devices. The constants specified in the rear column can be used in the robot program (see section 6.1).

Device status	Value	Meaning	Name of the URScript constant
NOT CONNECTED	0	Gripper module not connected	S_NOT_CONNECTED
NOT INITIALIZED	1	Gripper module not initialized	S_NOT_INITIALIZED
IDLE	2	Drive inactive Fingers can be moved manually Device deactivated	S_DISABLED
RELEASED	3	Workpiece released	S_RELEASED
NO PART	4	No workpiece found	S_NO_PART
HOLDING	5	Workpiece is held	S_HOLDING
ENABLED	6	Drive active Finger position is held Device activated	S_OPERATING
FAULT	7	Error state	S_FAULT

⚠ Do not assign any values to the UR script constants other than those listed in the table, as otherwise the function of the URCap will be impaired and malfunctions may occur!

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 where misuse could result in personal injury.