



GRIPLINK FÜR UNIVERSAL ROBOTS

Version 2.1.0
August 2024



Inhalt

1	Einführung.....	3
1.1	Notation und Symbole	3
1.2	Bestimmungsgemäße Verwendung.....	3
1.3	Systemvoraussetzungen	4
1.4	Lizenzbestimmungen	4
1.5	Demo-Programme	4
2	Installation	4
2.1	Installation der Software	4
2.2	Deinstallieren der Software	7
3	Hardware-Setup.....	8
4	Funktionsumfang des Plugins	9
4.1	Betrieb mittels Programm-Knoten	9
4.2	Betrieb mittels URScript	9
5	Vorbereitung des Roboters.....	10
5.1	Setup	10
6	Programm-Knoten (Nodes).....	12
6.1	Prinzipieller Programmablauf	12
6.2	Verbindung aufbauen – GRIPLINK Connect	14
6.3	Verbindung schließen – GRIPLINK Bye	15
6.4	Ein angeschlossenes Gerät prüfen – GRIPLINK Check Device.....	16
6.5	Gerät referenzieren – GRIPLINK Home.....	18
6.6	Gerät aktivieren und deaktivieren – GRIPLINK Enable/Disable	20
6.7	Greifen und Freigeben – GRIPLINK Grip/Release	23
6.8	Flexibles Greifen, Freigeben und Vorpositionieren – GRIPLINK Flexgrip/Flexrelease	26
6.9	Auswertung der Fingerposition – GRIPLINK Get Position.....	29
6.10	Zustandsabfrage – GRIPLINK Devstate	31
6.11	Greifen und Freigeben mit mehreren Greifern – GRIPLINK Multi Grip/Release	33
6.12	Gerätewerte auslesen – GRIPLINK Value.....	36
6.13	Gerätewert setzen – GRIPLINK Set Value	38
6.14	Auf Gerätewert warten – GRIPLINK Wait Value	40
6.15	Ansteuerung des LED-Leuchtrings – GRIPLINK LED	42
6.16	Konfigurieren eines Griff-Presets – GRIPLINK Set Grip Config.....	44
6.17	Mechanische Klemmung steuern – GRIPLINK Clamp	46
Anhang A	Gerätezustand.....	48

1 Einführung

Mit der GRIPLINK-Technologie können IO-Link kompatible Automationskomponenten über eine Netzwerkverbindung mit Robotersystemen führender Hersteller verbunden werden. Das GRIPLINK-Plugin für Universal Robots ist das steuerungsseitige Bindeglied und ermöglicht die einfache Einbindung der GRIPLINK-Technologie von WEISS ROBOTICS in Robotersysteme des Herstellers Universal Robots.



Diese Anleitung beschreibt die Funktionen des FLEXGRIP-Plugins für Roboter von Universal Robots. Informationen über Montage, Inbetriebnahme und Betrieb des GRIPKIT-Easy Greifmoduls entnehmen Sie der Betriebsanleitung. Diese finden Sie online unter www.weiss-robotics.com/gripkit-easy/.

1.1 Notation und Symbole

Zur besseren Übersicht werden in dieser Anleitung folgende Symbole verwendet:



Funktions- oder sicherheitsrelevanter Hinweis. Nichtbeachtung kann die Sicherheit von Personal und Anlage gefährden, das Gerät beschädigen oder die Funktion des Gerätes beeinträchtigen.



Zusatzinformation zum besseren Verständnis des beschriebenen Sachverhalts.



Verweis auf weiterführende Informationen.

Das GRIPLINK-Plugin ist kompatibel mit allen Geräten, die das GRIPLINK-Protokoll über TCP/IP unterstützen. Dies umfasst den GRIPLINK-ET4, sowie die Greifmodule der WPG-Serie.

Zur leichteren Lesbarkeit werden diese Geräte unter dem Begriff „GRIPLINK Controller“ zusammengefasst.

1.2 Bestimmungsgemäße Verwendung

Die Software „GRIPLINK-Plugin“ ist zur Kommunikation zwischen einem GRIPLINK Controller von WEISS ROBOTICS und einer Robotersteuerung bestimmt. Die Anforderungen der zutreffenden Richtlinien sowie die Installations- und Betriebshinweise in dieser Anleitung müssen beachtet und eingehalten werden. Eine andere oder darüberhinausgehende Verwendung gilt als nicht bestimmungsgemäß. Für hieraus resultierende Schäden haftet der Hersteller nicht.

1.3 Systemvoraussetzungen

Dieses Plugin ist kompatibel zu GRIPLINK-ET4 ab Firmware-Version 5.0.0 und WPG-Greifmodulen ab Firmware-Version 2.0.1.

Zum Betrieb wird eine der folgenden Universal Robots Robotersteuerungen benötigt:

- UR CB3.1 mit Softwarestand 3.11 oder höher
- UR e-Series mit Softwarestand 5.5 oder höher



Die IP-Adresse des GRIPLINK Controllers muss im selben Subnetz liegen wie die der Robotersteuerung.

1.4 Lizenzbestimmungen

Das GRIPLINK-Plugin ist urheberrechtlich geschützt. Die jeweils gültigen Lizenzbestimmungen liegen dem Softwarepaket bei. Mit der Installation akzeptieren Sie diese Lizenzbestimmungen.

1.5 Demo-Programme

Die im Softwarepaket enthaltenen Demo-Programme zeigen die Anwendung des Plugins. Sie sind ausschließlich für Testzwecke gedacht!

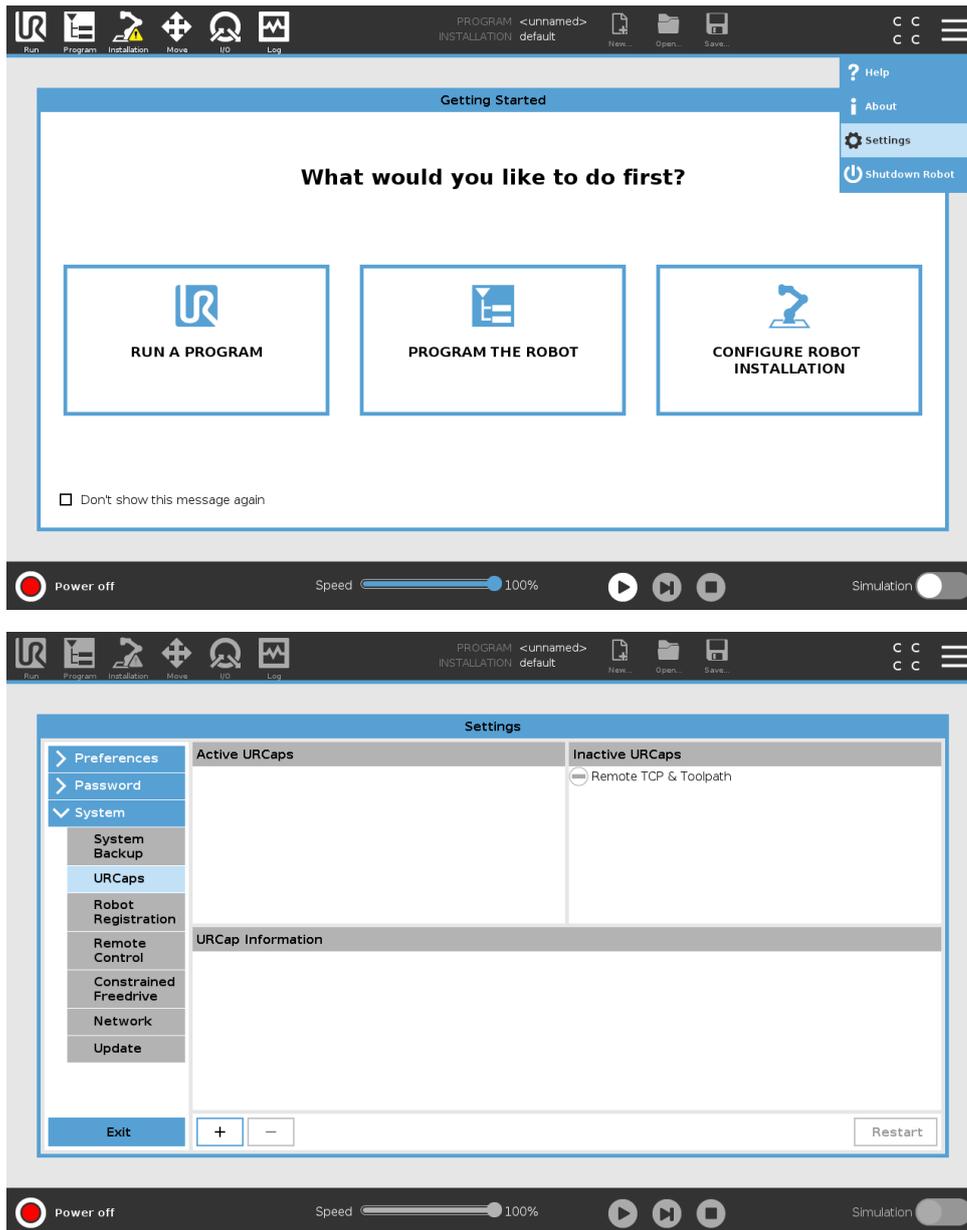
2 Installation

2.1 Installation der Software

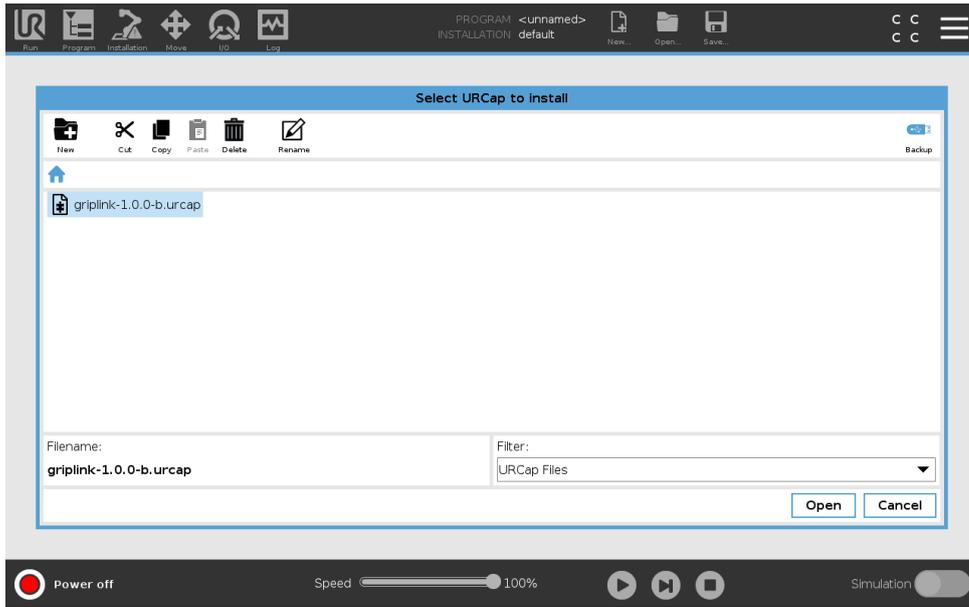


Stellen Sie sicher, dass Sie die aktuelle Version des GRIPLINK-Plugins verwenden. Die aktuelle Version kann unter www.griplink.de heruntergeladen werden.

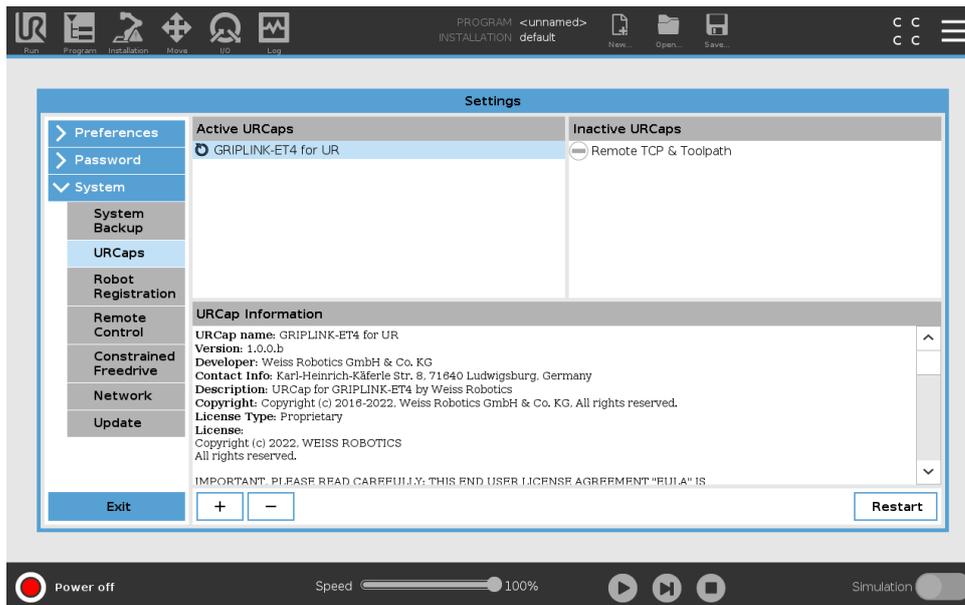
1. Laden Sie die Plugin-Datei „griplink_plugin_universalrobots_<Version>.zip“ herunter.
2. Entpacken Sie das zuvor heruntergeladene ZIP-Archiv mit dem GRIPLINK-Plugin in das Stammverzeichnis eines USB-Sticks und stecken Sie diesen in den USB-Slot des Teach Pendants.
3. Öffnen Sie die Einstellungen und navigieren Sie in das Menü „System/URCaps“



4. Drücken Sie die Taste „+“ und wählen Sie die zuvor entpackte .urcap-Datei aus



5. Starten Sie den Roboter neu, indem Sie die Taste „Restart“ drücken



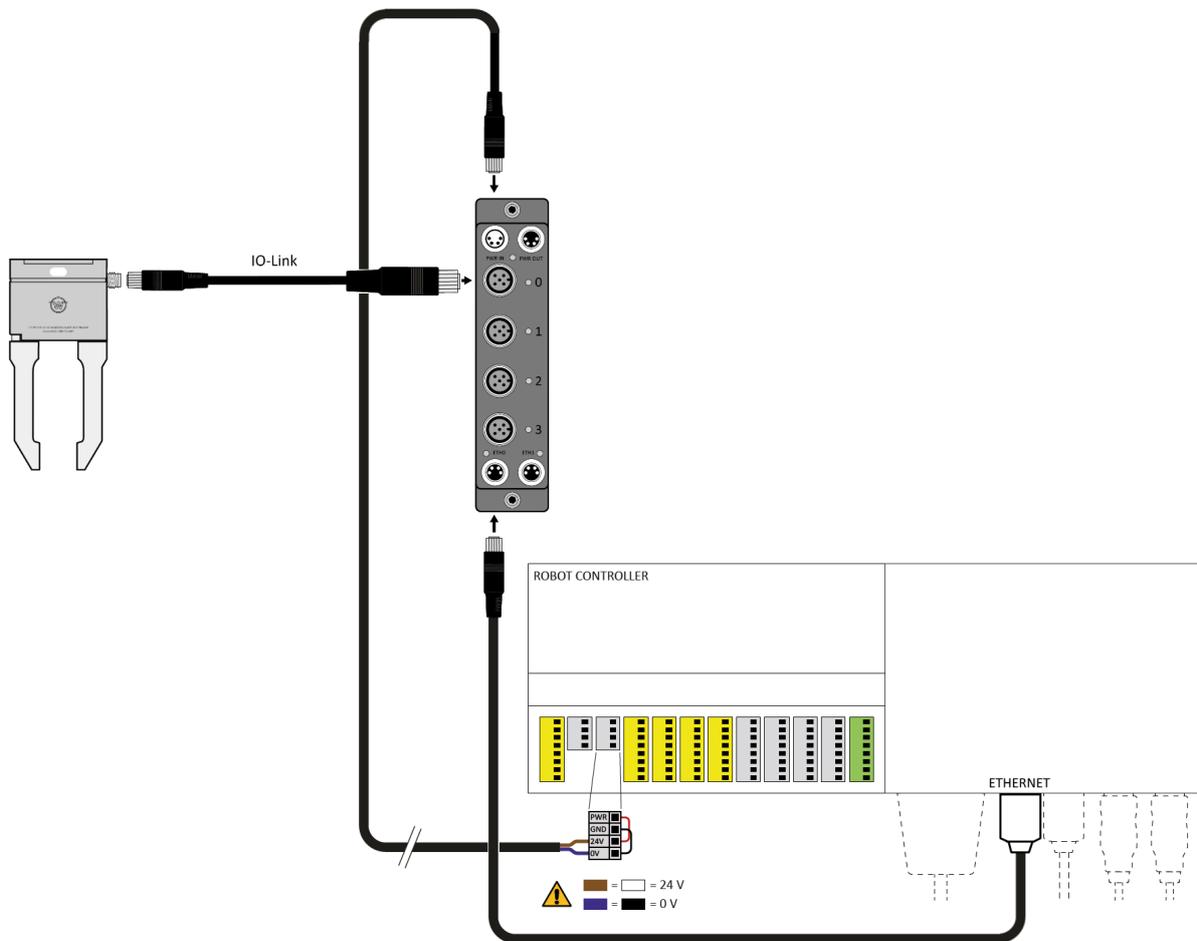
2.1.1 Überprüfen der Installation

Nachdem Sie den Installationsprozess beendet haben, erscheinen im Hauptmenü „Installation“ der Eintrag „GRIPLINK“, sowie im Hauptmenü „Program“ unter dem Menüpunkt „URCaps“ verschiedene GRIPLINK-Nodes.

2.2 Deinstallieren der Software

Um das GRIPLINK-Plugin wieder von Ihrem Roboter zu entfernen, befolgen Sie die Anweisungen in der Anleitung der Robotersteuerung.

3 Hardware-Setup



Falls Sie sich nicht sicher sind, ob die Spannungsversorgung über den Anschluss der Robotersteuerung ausreichend dimensioniert ist, um alle am GRIPLINK angeschlossenen Geräte zu betreiben, nutzen Sie eine externe Spannungsversorgung!



Die Spannungsversorgung muss im Zusammenhang mit den angeschlossenen Geräten ausreichend ausgelegt werden!

4 Funktionsumfang des Plugins

4.1 Betrieb mittels Programm-Knoten

Das Plugin stellt nach der Installation diverse Funktionsknoten zur Verfügung, die wie andere Roboter-Befehle in ein Roboter-Programm eingefügt werden können.

4.2 Betrieb mittels URScript

Das Plugin umfasst einen Satz an Grundfunktionen in URScript, die auch alleinstehend aus dem Roboterprogramm heraus mittels des entsprechenden Skript-Programmknotts heraus ausgerufen werden können. Diese Grundfunktionen werden automatisch in das Roboterprogramm eingebaut, wenn das Plugin installiert ist.

In Abschnitt 6 werden die verfügbaren Funktionen genauer beschrieben.

5 Vorbereitung des Roboters

Die grundlegenden Einstellungen des GRIPLINK-URCap sind in der entsprechenden Installation durchzuführen.



Die Konfiguration des GRIPLINK Controllers erfolgt über die jeweilige Weboberfläche, die unter der eingestellten IP-Adresse (Standard: 192.168.1.40) erreichbar ist.

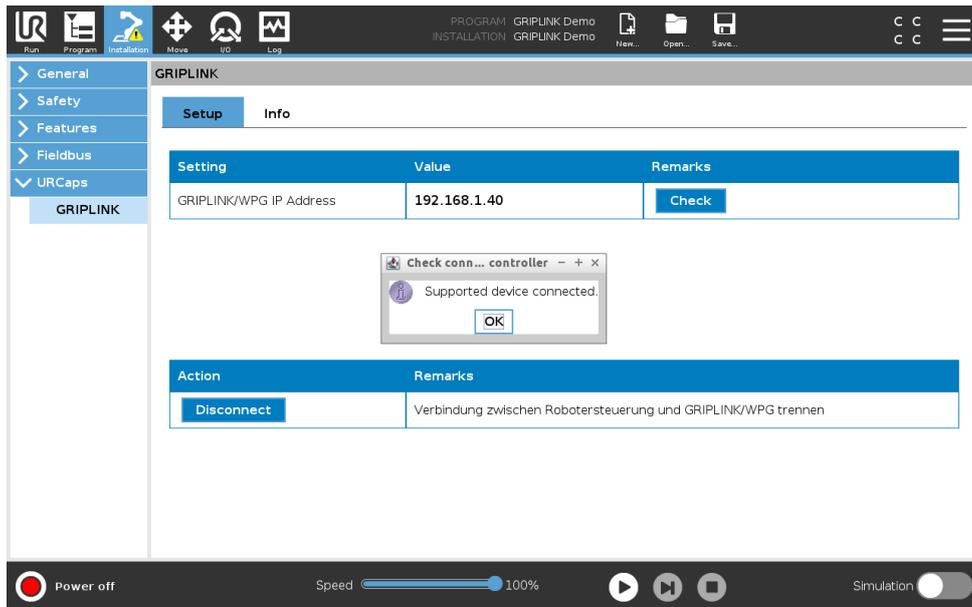
5.1 Setup

Im Setup wird die IP-Adresse des am Roboter angeschlossenen GRIPLINK Controllers eingetragen.

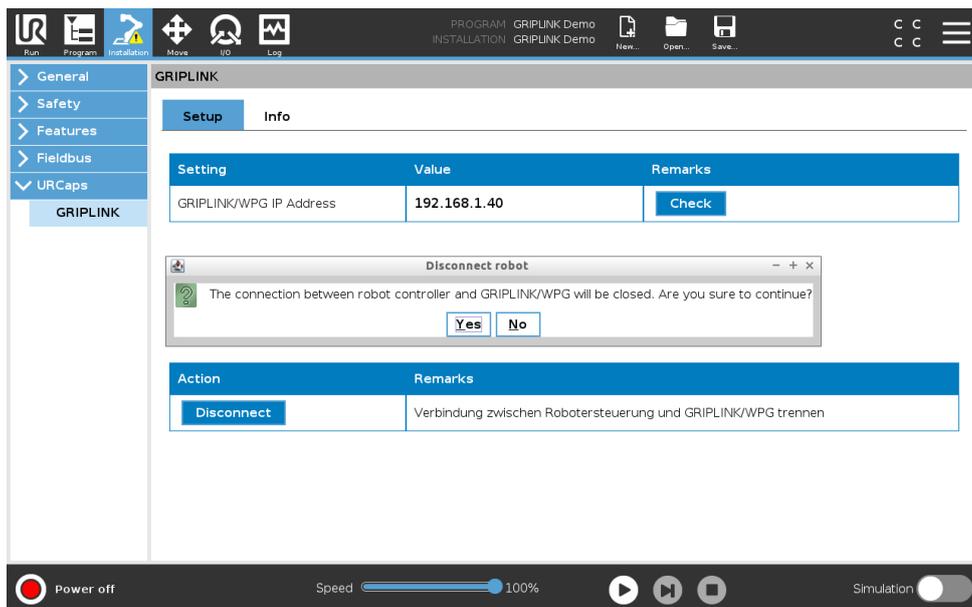
Setting	Value	Remarks
GRIPLINK/WPG IP Address	192.168.1.40	<input type="button" value="Check"/>

Action	Remarks
<input type="button" value="Disconnect"/>	Verbindung zwischen Robotersteuerung und GRIPLINK/WPG trennen

Über die Taste „Check“ kann geprüft werden, ob der konfigurierte GRIPLINK Controller verbunden ist.



Ist eine Verbindung zwischen Robotersteuerung und GRIPLINK Controller aktiv, können keine Greifbefehle über die Schaltflächen der Programmknoten ausgeführt werden. Um die Verbindung dafür zu trennen, kann die Schaltfläche „Trennen“ verwendet werden:



Trennen Sie die Verbindung nicht während eines laufenden Roboterprogramms!
Beschädigung und Verletzungsgefahr durch herabfallende Teile möglich!

6 Programm-Knoten (Nodes)

6.1 Prinzipieller Programmablauf

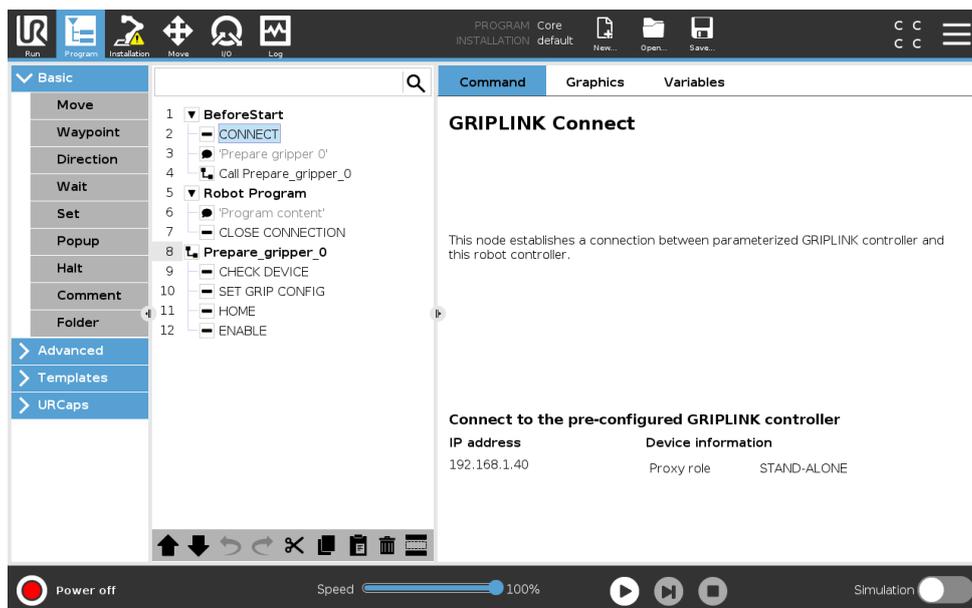
Jedes Roboterprogramm, welches Geräte wie Greifer und Sensoren über den GRIPLINK ansteuert, sollte die folgenden Design-Richtlinien befolgen.

6.1.1 Programm

Der BeforeStart-Abschnitt

1. enthält zu Beginn den Node „GRIPLINK Connect“
2. enthält danach für jedes am GRIPLINK angeschlossene Gerät den Aufruf eines Unterprogramms zur Greifer-Initialisierung.

Hier wird mittels „GRIPLINK Check Device“ das korrekte Gerät gewährleistet und anschließend gegebenenfalls gerätespezifisch konfiguriert, referenziert und aktiviert



Der Robot Program-Abschnitt

1. enthält die Befehle zum Greifen und Freigeben
2. verarbeitet Variablen, die durch Nodes z.B. zur Zustands- und Werteabfrage beschrieben werden
3. Bei Programmen, die nicht dauerhaft durchlaufen, sollte am Ende des Programms der Node „GRIPLINK Bye“ aufgerufen werden, um ein sauberes Beenden der Netzwerkschnittstelle zu gewährleisten

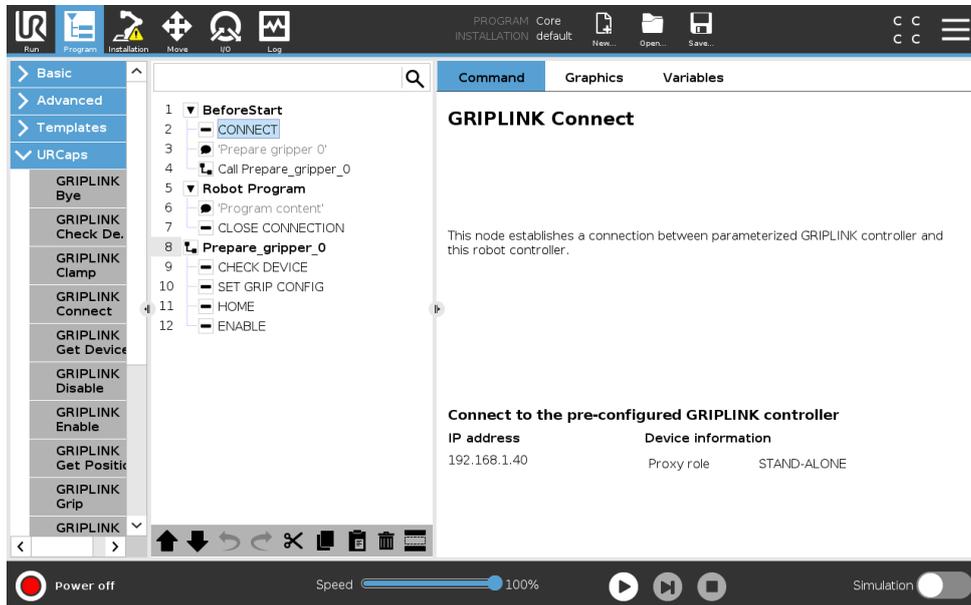


Nur wenn dieser prinzipielle Ablauf eingehalten wird, kann die korrekte Funktion gewährleistet werden.

In den folgenden Abschnitten werden verfügbaren Befehle des Plugins beschrieben. Jeder Befehl ist sowohl als grafischer Knoten als auch als URScript-Implementierung verwendbar.

6.2 Verbindung aufbauen – GRIPLINK Connect

Dieser Befehl stellt die Verbindung zwischen GRIPLINK Controller und der Robotersteuerung her. Es wird die auf der Installationsseite des Plugins eingestellte IP-Adresse angezeigt.



Ist der GRIPLINK Controller unter der in der Installation eingestellten IP-Adresse erreichbar, werden zusätzliche Informationen wie die Proxy-Rolle angezeigt. Falls nicht, ist die Verbindung zu prüfen.

6.2.1 Befehlsaufruf mit Script-Code

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

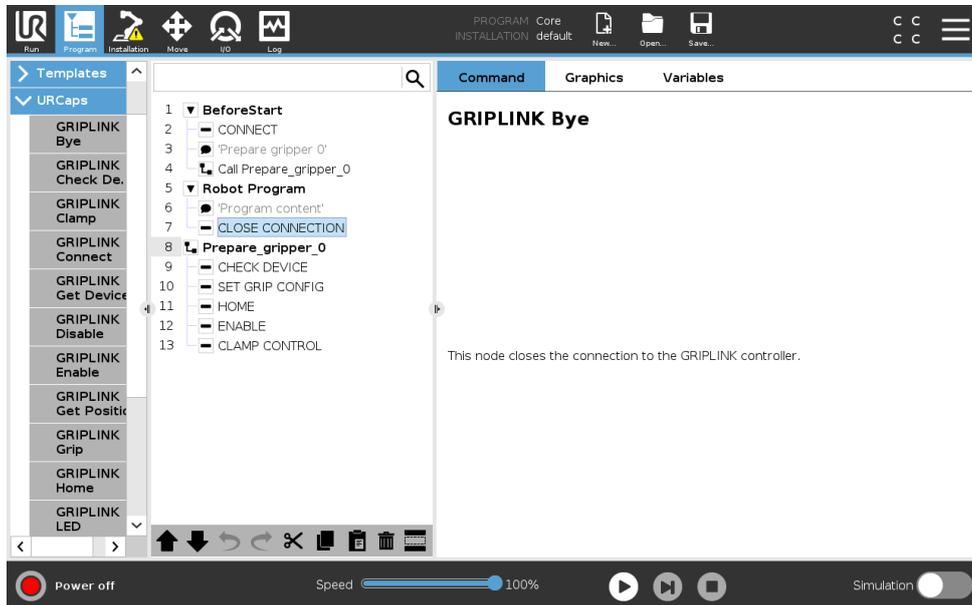
Parameter	Typ	Bedeutung
<GRIPLINK_IP>	String	IP-Adresse der GRIPLINK-Controllers Bspw. "192.168.1.40"
<SOCKET_NAME>	String	Name des zu verwendenden Sockets, bspw. "sock_griplink" Der Socket-Name wird in allen nachfolgenden Knoten im Roboterprogramm verwendet!

Beispiel:

```
GL_CONNECT("192.168.1.40", "sock_griplink")
```

6.3 Verbindung schließen – GRIPLINK Bye

Dieser Befehl trennt die Verbindung zwischen GRIPLINK Controller und der Robotersteuerung. Dies kann dann sinnvoll sein, wenn der GRIPLINK Controller hinter einem Wechsler angebracht ist.



Nach dem Trennen der Verbindung können bis zum nächsten Aufruf von GRIPLINK Connect keine Befehle gesendet werden!

6.3.1 Befehlsaufruf mit Script-Code

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

Parameter	Typ	Bedeutung
<SOCKET_NAME>	String	Name des zu verwendenden Sockets, bspw. "sock_griplink" Der Socket-Name wird in allen nachfolgenden Knoten im Roboterprogramm verwendet!

Beispiel:

```
GL_BYE("sock_griplink")
```

6.4 Ein angeschlossenes Gerät prüfen – GRIPLINK Check Device

Bevor ein Gerät im Programm verwendet wird, empfiehlt es sich zu prüfen, ob das erwartete Gerät angeschlossen auch ist. Der Knoten „GRIPLINK Check Device“ prüft am ausgewählten Port, ob Vendor- und Product-ID (VID/PID) des angeschlossenen Geräts mit den erwarteten Werten übereinstimmen. Das Roboterprogramm wird sofort angehalten, sollten diese nicht übereinstimmen.



Weitere Informationen zu Vendor- und Product-ID finden Sie in der Betriebsanleitung des jeweiligen Geräts.

In den Einstellungen des „GRIPLINK Check Device“-Knotens können Port, Vendor- und Product-ID eingestellt werden. Im Beispiel wird an Port 0 ein IEG 55-020 von Weiss Robotics (VID 815, PID 20) erwartet.

The screenshot displays the GRIPLINK software interface. On the left, a sidebar lists various GRIPLINK nodes, with 'GRIPLINK Check Device' selected. The main workspace shows a program tree with the following steps:

- 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

The 'CHECK DEVICE' node is expanded, showing its configuration:

- Port:** Port 0
- Vendor ID:** 815
- Product ID:** 20

Below the configuration, a description reads: "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)." At the bottom of the interface, there is a 'Power off' button, a 'Speed' slider set to 100%, and a 'Simulation' toggle switch.

6.4.1 Befehlsaufruf mit Script-Code

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

Parameter	Typ	Bedeutung
<PORT>	Integer	Port-Index (0..31)
<VENDOR_ID>	Integer	Vendor-ID des erwarteten Gerätes
<PRODUCT_ID>	Integer	Product-ID des erwarteten Gerätes
<SOCKET_NAME>	String	Name des zu verwendenden Sockets, bspw. "sock_griplink" Der Socket-Name wird in allen nachfolgenden Knoten im Roboterprogramm verwendet!

Beispiel:

```
# 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 Gerät referenzieren – GRIPLINK Home

Gegebenenfalls muss ein Gerät zu Beginn referenziert werden. Hierfür wird der Node „GRIPLINK Home“ genutzt.



Weitere Details finden sich in der Betriebsanleitung des jeweiligen Gerätes.

The screenshot displays the GRIPLINK software interface. On the left, a tree view shows the program structure with the 'HOME' node under 'Prepare_gripper_0' selected. The main window is titled 'GRIPLINK Home' and contains the following information:

- Command:** GRIPLINK Home
- Graphics:** (Empty)
- Variables:** (Empty)
- Description:** This node references the device connected to the selected port. *Note: This node is only supported by compatible devices.*
- Ports:** A grid of checkboxes for ports 0-7, with Port 0 selected.
- Device actions:** Three buttons labeled 'Home', 'Enable', and 'Disable'.

At the bottom of the interface, there is a 'Power off' button, a 'Speed' slider set to 100%, and a 'Simulation' toggle switch.



Es können mehrere Greifmodule gleichzeitig referenziert werden.



Es muss mindestens ein Port ausgewählt sein!

6.5.1 Befehlsaufruf mit Script-Code

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

Parameter	Typ	Bedeutung
<PORT>	Integer	Port-Index (0..31)
<SOCKET_NAME>	String	Name des zu verwendenden Sockets, bspw. "sock_griplink" Der Socket-Name wird in allen nachfolgenden Knoten im Roboterprogramm verwendet!

Beispiel:

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

6.6 Gerät aktivieren und deaktivieren – GRIPLINK Enable/Disable

Geräte können im Betrieb aktiviert und deaktiviert werden.

Mit dem Haken bei „Warten auf Zustandsübergänge“ kann gesteuert werden, ob nach Ausführung des Befehls kontrolliert werden soll, dass sich der Gerätezustand der Geräte an den selektierten Ports auch wirklich ändert.

Befindet sich ein Gerät bereits im Zustand DISABLED wenn der Programm-Knoten „GRIPLINK Disable“ aufgerufen wird, so führt dies zu einem Timeout und das Programm wird abgebrochen.



Einige Geräte müssen vor der Verwendung zunächst durch den Enable-Befehl aktiviert werden!

Weitere Details finden sich in der Betriebsanleitung des jeweiligen Gerätetreibers.



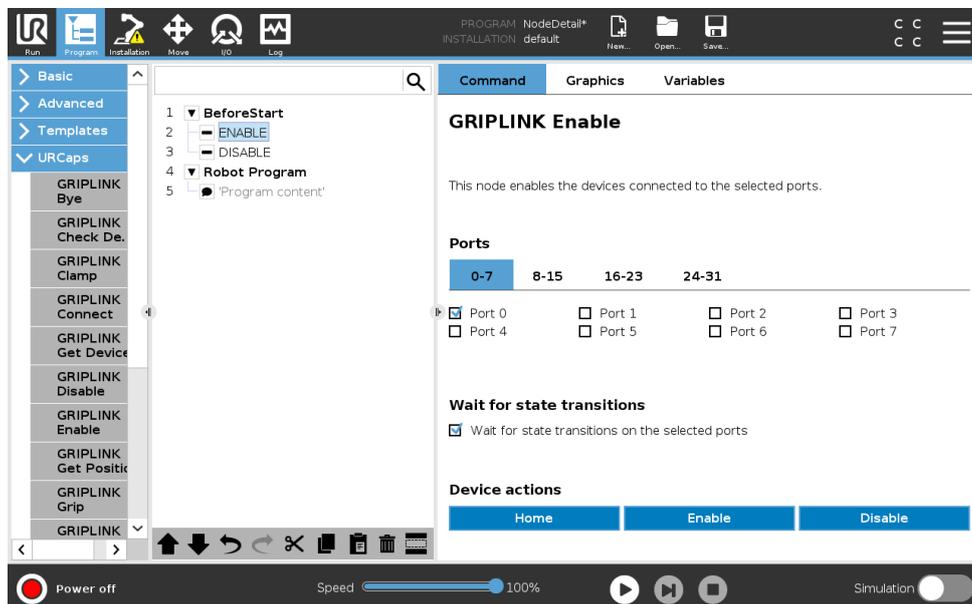
Es können mehrere Geräte gleichzeitig aktiviert/deaktiviert werden.



Es muss mindestens ein Port ausgewählt sein!



Stellen Sie sicher, dass sich die Geräte in einem zulässigen Zustand befinden, wenn der Haken bei „Warten auf Zustandsübergänge“ gesetzt ist!





Es können mehrere Greifmodule gleichzeitig aktiviert/deaktiviert werden.



Es muss mindestens ein Port ausgewählt sein!

6.6.1 Befehlsaufruf mit Script-Code

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

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

Parameter	Typ	Bedeutung
<PORT>	Integer	Port-Index (0..31)
<WSTR_ENABLED>	Boolean	„True“, um auf Zustandsübergang des Gerätes zu warten „False“, um nicht auf Zustandsübergang des Gerätes zu warten
<SOCKET_NAME>	String	Name des zu verwendenden Sockets, bspw. "sock_griplink"

		Der Socket-Name wird in allen nachfolgenden Knoten im Roboterprogramm verwendet!
--	--	--

Beispiel:

```
# 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 Greifen und Freigeben – GRIPLINK Grip/Release

Jedes Greifmodul kennt die Grundbefehle Greifen („Grip“) und Freigeben („Release“). Abhängig vom Greifmodul können bis zu acht frei konfigurierbare Griff-Presets ausgeführt werden.



Weitere Informationen zu den Griffen und deren Parametrierung finden Sie in der Betriebsanleitung des jeweiligen Greifmoduls.

In den Nodes „GRIPLINK Grip“ und „GRIPLINK Release“ können Port des gewünschten Greifers und der Index des Griffs ausgewählt werden

The screenshot shows the 'GRIPLINK Grip' node configuration. The left sidebar lists various GRIPLINK nodes, with 'GRIPLINK Grip' selected. The main workspace displays a tree view of the program structure, including 'BeforeStart', 'Robot Program', 'GRIP', 'HOLDING', 'NO PART', and 'RELEASE'. The 'GRIP' node is expanded, showing its sub-nodes. The right panel shows the configuration for the 'GRIPLINK Grip' node, including a description, a note about the number of available grip indices, and configuration options for 'Port' (set to 'Port 0') and 'Preset index' (set to 'Preset 0'). Below these options are 'Device actions' buttons: 'Home', 'Enable', 'Disable', 'Grip', and 'Release'. The bottom status bar shows 'Power off', 'Speed' at 100%, and a 'Simulation' toggle.

The screenshot shows the 'GRIPLINK Release' node configuration. The left sidebar lists various GRIPLINK nodes, with 'GRIPLINK Release' selected. The main workspace displays a tree view of the program structure, including 'BeforeStart', 'Robot Program', 'GRIP', 'HOLDING', 'NO PART', and 'RELEASE'. The 'RELEASE' node is expanded, showing its sub-nodes. The right panel shows the configuration for the 'GRIPLINK Release' node, including a description, a note about the number of available grip indices, and configuration options for 'Port' (set to 'Port 0') and 'Preset index' (set to 'Preset 0'). Below these options are 'Device actions' buttons: 'Home', 'Enable', 'Disable', 'Grip', and 'Release'. A 'Wait for state transitions' section is also visible, with a checked checkbox for 'Wait for state transition on the selected port'. The bottom status bar shows 'Power off', 'Speed' at 100%, and a 'Simulation' toggle.

Mit dem Haken bei „Warten auf Zustandsübergänge“ kann beim Programm-Knoten „GRIPLINK Release“ gesteuert werden, ob nach Ausführung des Befehls gewartet werden soll, bis sich der Gerätezustand des Geräts am selektierten Port geändert hat.

Befindet sich ein Gerät bereits im Zustand RELEASED wenn der Programm-Knoten „GRIPLINK Release“ aufgerufen wird, so führt dies zu einem Timeout und das Programm wird abgebrochen.



Stellen Sie sicher, dass sich die Geräte in einem zulässigen Zustand befinden, wenn der Haken bei „Warten auf Zustandsübergänge“ gesetzt ist.

6.7.1 Befehlsaufruf mit Script-Code

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

Parameter	Typ	Bedeutung
<PORT>	Integer	Port-Index (0..31)
<PRESET_INDEX>	Integer	Preset-Index (0..7)
<WSTR_ENABLED>	Boolean	„True“, um auf Zustandsübergang des Gerätes zu warten „False“, um nicht auf Zustandsübergang des Gerätes zu warten
<SOCKET_NAME>	String	Name des zu verwendenden Sockets, bspw. "sock_griplink" Der Socket-Name wird in allen nachfolgenden Knoten im Roboterprogramm verwendet!

Beispiel:

```
# 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 Auswertung des Greifzustands

Der Node „GRIPLINK Grip“ erkennt nach dem Greifen, ob der Greifer ein Bauteil gegriffen hat oder nicht. Je nach dem werden dann die Child-Nodes unter „HOLDING“ (Bauteil wurde gegriffen) oder unter „NO PART“ (Bauteil wurde nicht gegriffen) ausgeführt.

6.8 Flexibles Greifen, Freigeben und Vorpositionieren – GRIPLINK Flexgrip/Flexrelease

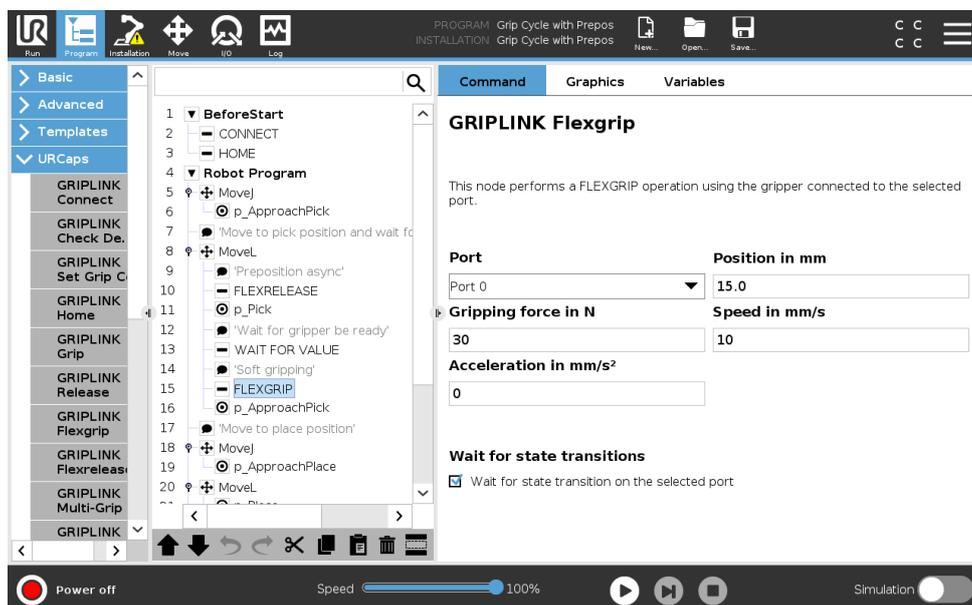
Die Greifmodule der WPG-Serie unterstützen flexibles Greifen und Freigeben. Hier können Position, Greifkraft und Bewegungsparameter unabhängig von Presets vorgegeben werden. Dadurch ist schnelles Vorpositionieren und sanftes Greifen möglich, sodass Taktzeiten gesenkt und Greifteile geschont werden können.



Weitere Informationen zu den Bewegungsparametern finden Sie in der Betriebsanleitung des jeweiligen Greifmoduls.

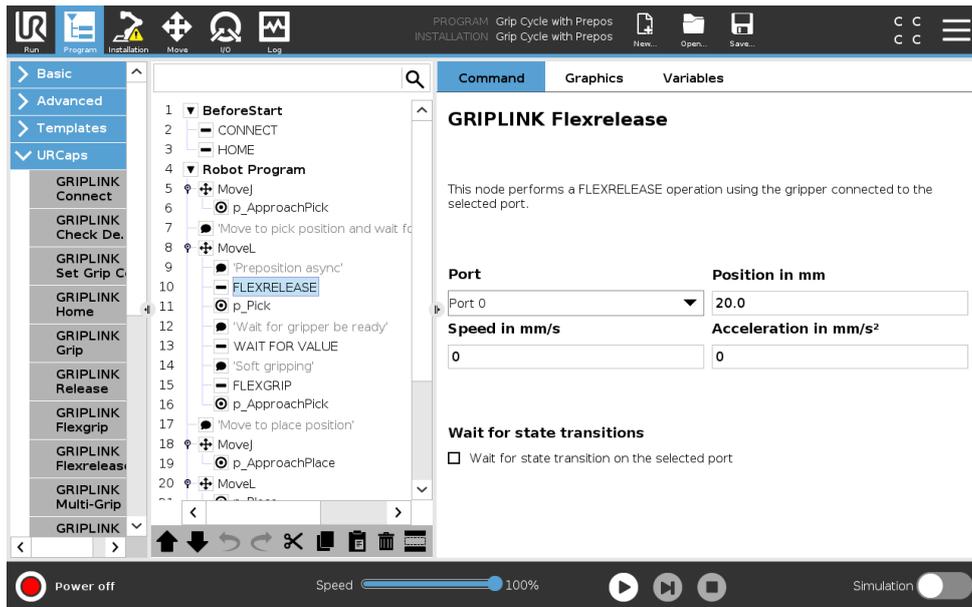
Im Programm-Knoten „GRIPLINK Flexgrip“ kann der Port des gewünschten Greifers ausgewählt werden. Die Zielposition wird in Millimetern angegeben. Die Greifkraft wird in Newton eingestellt. Als Bewegungsparameter können die Geschwindigkeit in mm/s und die Beschleunigung in mm/s² vorgegeben werden.

Mit dem Haken bei „Warten auf Zustandsübergänge“ kann gesteuert werden, ob nach Ausführung des Befehls gewartet werden soll, bis sich der Gerätezustand des Greifers am selektierten Port geändert hat.



Im Programm-Knoten „GRIPLINK Flexrelease“ kann der Port des gewünschten Greifers ausgewählt werden. Die Zielposition wird in Millimetern angegeben. Als Bewegungsparameter können die Geschwindigkeit in mm/s und die Beschleunigung in mm/s² vorgegeben werden.

Mit dem Haken bei „Warten auf Zustandsübergänge“ kann beim Programm-Knoten „GRIPLINK Release“ gesteuert werden, ob nach Ausführung des Befehls gewartet werden soll, bis der Greifer am selektierten Port die Zielposition erreicht hat.



6.8.1 Befehlsaufruf mit Script-Code

```

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

```

Parameter	Typ	Bedeutung
<PORT>	Integer	Port-Index (0..31)
<POSITION>	Double	Zielposition in 1/1000 mm
<GRIPPING_FORCE>	Integer	Greifkraft in 1/1000 N (nur für GL_FLEXGRIP)
<SPEED>	Integer	Geschwindigkeit in 1/1000 mm/s
<ACCELERATION>	Integer	Beschleunigung in 1/1000 mm/s ²

<WSTR_ENABLED>	Boolean	<u>GL_FLEXGRIP</u> „True“, um auf Zustandsübergang des Gerätes zu warten „False“, um direkt mit dem nächsten Knoten fortzufahren <u>GL_FLEXRELEASE</u> „True“, um auf Erreichen der Zielposition „False“, um direkt mit dem nächsten Knoten fortzufahren
<SOCKET_NAME>	String	Name des zu verwendenden Sockets, bspw. "sock_griplink" Der Socket-Name wird in allen nachfolgenden Knoten im Roboterprogramm verwendet!

Beispiel:

```

# 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 Auswertung der Fingerposition – GRIPLINK Get Position

Um die Position der Fingerbacken auszulesen, beispielsweise zur Überprüfung des gegriffenen Bauteils anhand dessen Größe, kann der Node „GRIPLINK Get Position“ verwendet werden. Damit wird die aktuelle Position der Fingerbacken des ausgewählten Greifmoduls in Tausendstel Millimeter in eine vom Benutzer frei definierbare Variable geschrieben.



Achten Sie bei der Definition der Variable darauf, dass diese nur für die Auswertung der Griffweite des Greifmoduls verwendet wird!

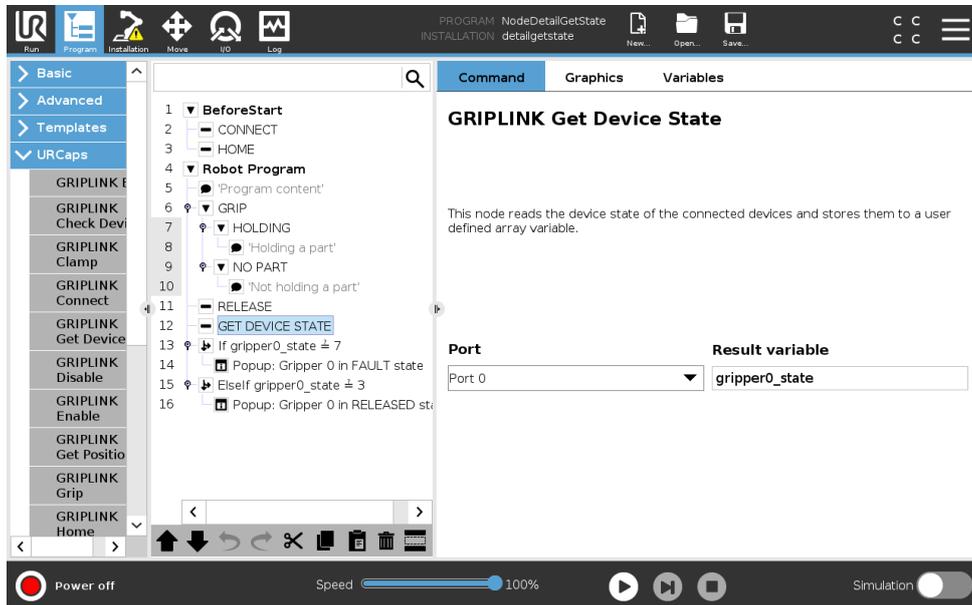
Eine Verarbeitung der Fingerposition kann anhand einfacher If-Else-Abfragen erfolgen:

6.9.1 Befehlsaufruf mit Script-Code

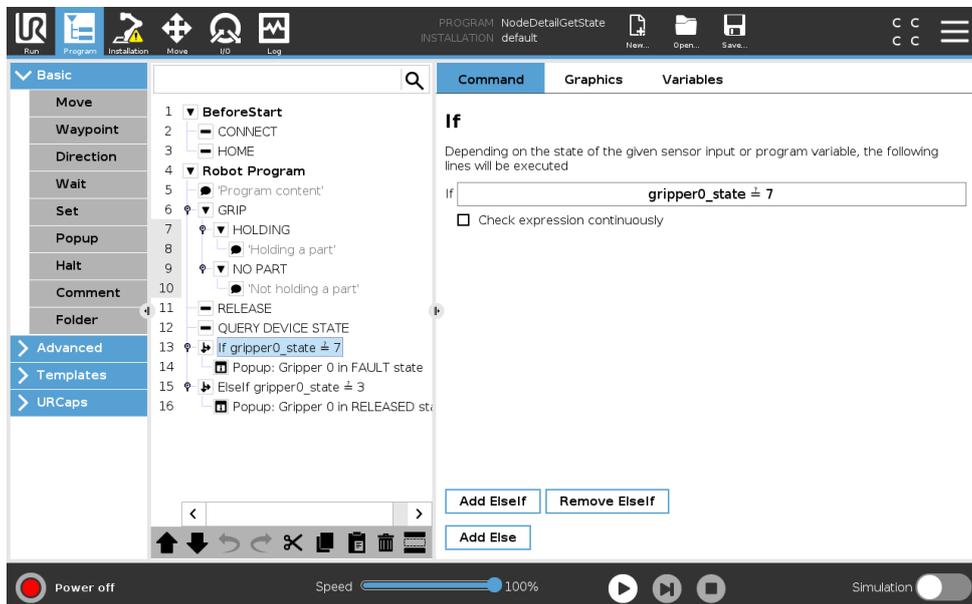
Die Abfrage der Position wird über die Funktion `GL_VALUE` umgesetzt. Als Wert-Index muss hierbei der Wert 0 verwendet werden (siehe Abschnitt 6.12.1).

6.10 Zustandsabfrage – GRIPLINK Devstate

Um den Zustand eines Geräts abzufragen, wird der Node „GRIPLINK Devstate“ genutzt. Dieser schreibt den gelesenen Zustand als Zahlenwert in die vom Benutzer definierte Variable.



Der Zustand kann nun mittels If-Else-Konstrukten verarbeitet werden. Im folgenden Beispiel wird der Zustand eines Greifers an Port 0 verarbeitet:



Die möglichen Gerätezustände sind in Anhang A aufgeführt!

6.10.1 Befehlsaufruf mit Script-Code

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

Parameter	Typ	Bedeutung
<PORT>	Integer	Port-Index (0..31)
<SOCKET_NAME>	String	Name des zu verwendenden Sockets, bspw. "sock_griplink" Der Socket-Name wird in allen nachfolgenden Knoten im Roboterprogramm verwendet!

Typ Rückgabewert	Bedeutung
Integer	Gerätezustand (siehe Anhang A)

Es können vordefinierte Konstanten für die Gerätezustände verwendet werden.

Beispiel:

```
# 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 Greifen und Freigeben mit mehreren Greifern – GRIPLINK Multi Grip/Release

Über die Nodes „GRIPLINK Multi-Grip“ und „GRIPLINK Multi-Release“ können Greif- und Freigabebefehle mit mehreren Greifern zeitsynchron ausgeführt werden. Es können die Ports der gewünschten Greifer und der einheitliche Index des Griff-Presets ausgewählt werden.

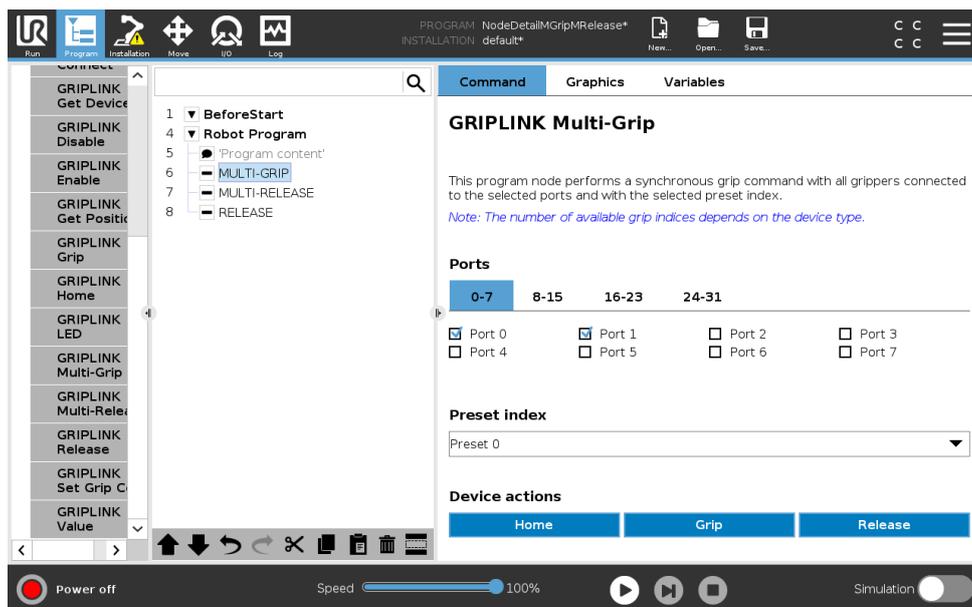
Mit dem Haken bei „Warten auf Zustandsübergänge“ kann beim Programm-Knoten „GRIPLINK Multi-Release“ gesteuert werden, ob nach Ausführung des Befehls gewartet werden soll, bis sich der Gerätezustand aller Geräte an den selektierten Ports geändert hat.

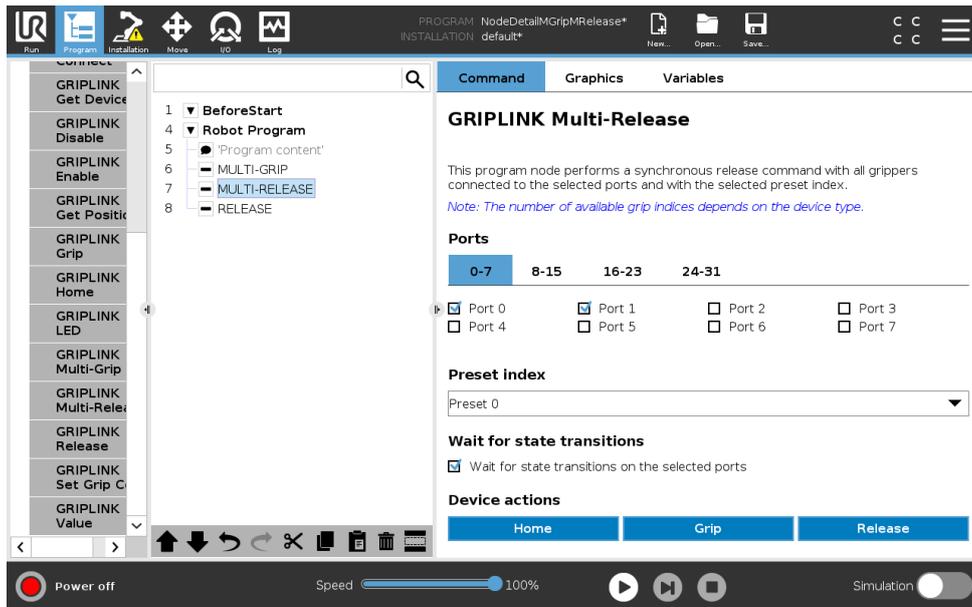
Ändert sich der Zustand eines Gerätes nicht innerhalb einer vorgegebenen Zeit, so führt dies zu einem Timeout und das Programm wird abgebrochen.

Beim Programm-Knoten „GRIPLINK Multi-Grip“ wird immer auf einen Zustandswechsel gewartet.



Stellen Sie sicher, dass sich die Geräte in einem zulässigen Zustand befinden, wenn beim Programm-Knoten „GRIPLINK Multi-Release“ der Haken bei „Warten auf Zustandsübergänge“ gesetzt ist!





Alle selektierten Greifer führen den Greif-/Freigabebefehl mit demselben Preset-Index aus. Stellen Sie vorab sicher, dass die Preset-Konfiguration korrekt eingestellt ist!

6.11.1 Befehlsaufruf mit Script-Code

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

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

Parameter	Typ	Bedeutung
<PORTS>	Integer List	<p>Ausgewählte Ports als Liste mit Werten 0 oder 1</p> <p>Liste umfasst mindestens ein und maximal 31 Elemente Jedes Element der Liste entspricht dem Port des jeweiligen Listen-Indizes Ein Port wird ausgewählt, wenn das entsprechende Element der Liste den Wert 1 enthält.</p> <p>Beispiel: Greifer an Ports 0, 2 und 3 sollen verwendet werden →Liste [1, 0, 1, 1]</p>
<PRESET_INDEX>	Integer	Preset-Index (0..7)
<WSTR_ENABLED>	Boolean	<p>„True“, um auf Zustandsübergang der Geräte zu warten „False“, um nicht auf Zustandsübergang der Geräte zu warten</p>
<SOCKET_NAME>	String	<p>Name des zu verwendenden Sockets, bspw. "sock_griplink"</p> <p>Der Socket-Name wird in allen nachfolgenden Knoten im Roboterprogramm verwendet!</p>

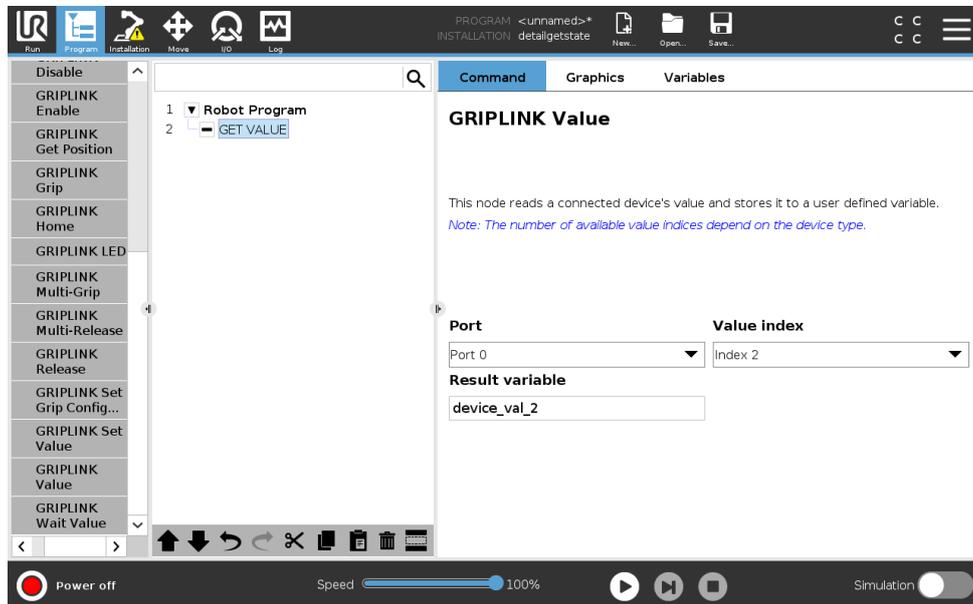
Beispiel:

```
# 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 Gerätewerte auslesen – GRIPLINK Value

Geräte wie Sensoren liefern Daten in Form indizierter Werte. Diese Werte können über den Node „GRIPLINK Value“ ausgelesen werden. Je nach Gerät können unterschiedlich viele Werte ausgelesen werden.



Es sind maximal acht auslesbare Werte möglich, je nach Gerät können es allerdings auch weniger sein.

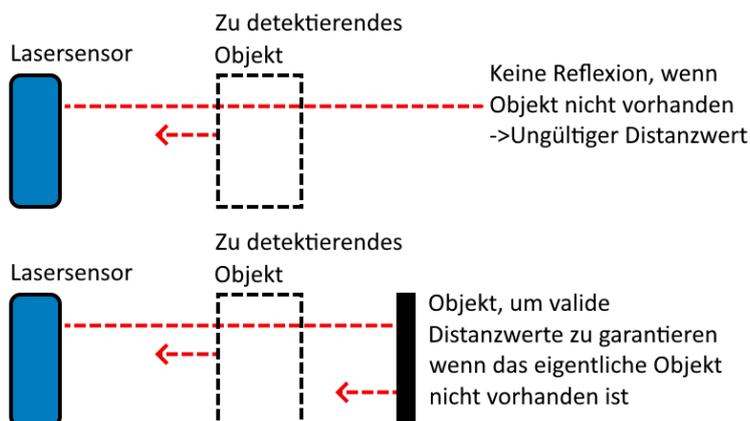


Stellen Sie deshalb sicher, dass stets ein valider Wert-Index eingestellt wird. Weitere Informationen finden sich in der Anleitung des jeweiligen Gerätetreibers.

Der Wert 2147483647 wird im GRIPLINK z.B. für Sensoren verwendet, deren Messgröße außerhalb des Messbereichs liegt.

Um Fehler im Programmablauf zu verhindern, sollte sichergestellt werden, dass stets valide Werte (< 2147483647) vom jeweiligen Gerät ausgelesen werden!

Dies kann z.B. sichergestellt werden, indem ein Distanzsensor stets ein Objekt detektiert und somit statt dem null-Wert immer eine valide Distanz liefert:



6.12.1 Befehlsaufruf mit Script-Code

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

Parameter	Typ	Bedeutung
<PORT>	Integer	Port-Index (0..31)
<VALUE_INDEX>	Integer	Value-Index (0..7)
<SOCKET_NAME>	String	Name des zu verwendenden Sockets, bspw. "sock_griplink" Der Socket-Name wird in allen nachfolgenden Knoten im Roboterprogramm verwendet!

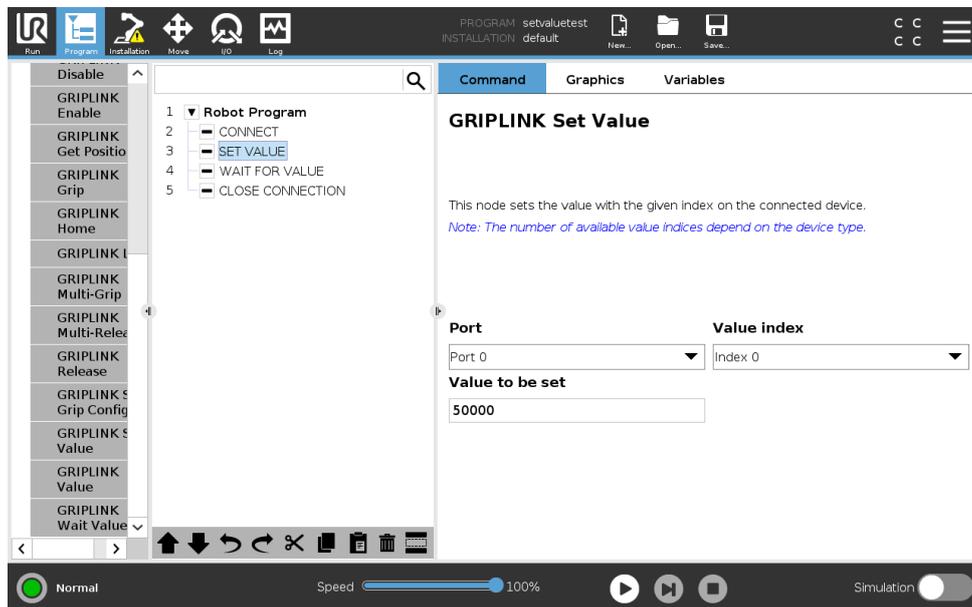
Typ Rückgabewert	Bedeutung
Integer	Wert des ausgelesenen Gerätewertes

Beispiel:

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

6.13 Gerätewert setzen – GRIPLINK Set Value

Einige Geräte unterstützen das Setzen von Werten. Hierfür wird der Node „GRIPLINK Set Value“ verwendet.



Diese Funktion ist nicht auf allen Geräten verfügbar.

6.13.1 Befehlsaufruf mit Script-Code

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

Parameter	Typ	Bedeutung
<PORT>	Integer	Port-Index (0..31)
<VALUE_INDEX>	Integer	Wert-Index (0..7) Je nach Gerät können auch weniger als 8 Werte verfügbar sein!
<VALUE>	Integer	Wert, der gesetzt werden soll Skaliert mit 1/1000
<SOCKET_NAME>	String	Name des zu verwendenden Sockets, bspw. "sock_griplink" Der Socket-Name wird in allen nachfolgenden Knoten im Roboterprogramm verwendet!

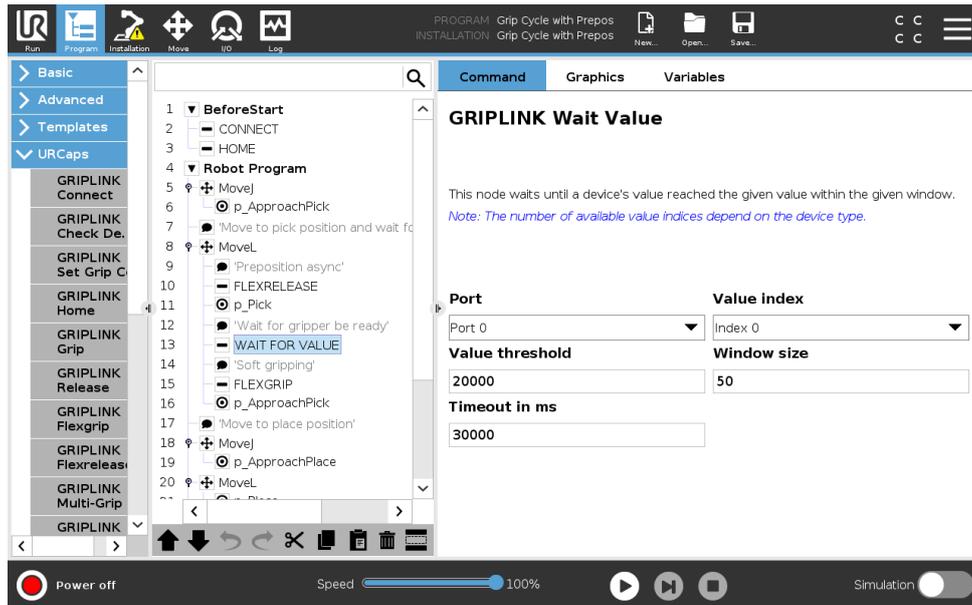
Beispiel:

```
# 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 Auf Gerätewert warten – GRIPLINK Wait Value

Nach dem Setzen eines Gerätewertes kann es erforderlich sein, dass gewartet wird, bis der Wert auch erreicht wurde. Bei Aktoren kann das beispielsweise eine Zielposition sein. Mit dem Node „GRIPLINK Wait Value“ kann dieses Verhalten abgebildet werden.

Es wird gewartet, bis der Gerätewert innerhalb der vorgegebenen Zeit in den Wertebereich (Fenster) um den Zielwert (Schwellwert) eintritt. Wird das Fenster nicht innerhalb einer bestimmten Zeit erreicht, so wird ein Zeitüberschreitungsfehler generiert.



Diese Funktion ist nicht auf allen Geräten verfügbar.



Die Fenstergröße sollte abhängig von der Änderungsrate des Gerätewertes vorab passend gewählt werden, um eine zuverlässige und genaue Erkennung zu ermöglichen!

6.14.1 Befehlsaufruf mit Script-Code

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

Parameter	Typ	Bedeutung
<PORT>	Integer	Port-Index (0..31)
<VALUE_INDEX>	Integer	Wert-Index (0..7) Je nach Gerät können auch weniger als 8 Werte verfügbar sein!
<VALUE_THRESHOLD>	Integer	Schwellwert, der erreicht werden soll Skaliert mit 1/1000
<WINDOW_SIZE>	Integer	Fenstergröße, die um den Schwellwert zentriert den Wertebereich vorgibt, innerhalb dessen der WAITVAL-Befehl ohne Fehler abschließt Skaliert mit 1/1000
<TIMEOUT_MS>	Integer	Maximale Dauer in ms, für die auf das Erreichen des Zielwertes gewartet wird
<SOCKET_NAME>	String	Name des zu verwendenden Sockets, bspw. "sock_griplink" Der Socket-Name wird in allen nachfolgenden Knoten im Roboterprogramm verwendet!

Beispiel:

```
# 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")
```

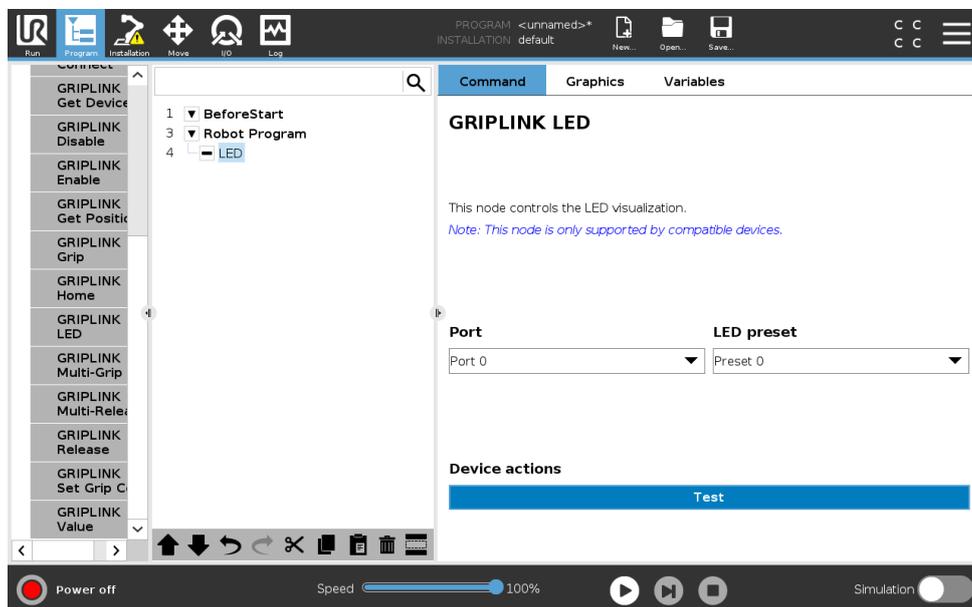
6.15 Ansteuerung des LED-Leuchtrings – GRIPLINK LED

Die Greifmodule der CRG-Serie von Weiss Robotics besitzen einen LED-Leuchtring zur Visualisierung verschiedener Betriebszustände. Die Ansteuerung kann als eigenständige Instruktion erfolgen. Alle CRG-Greifmodule besitzen jeweils acht konfigurierbare Presets, die über die Instruktion gestartet werden können.



Weitere Informationen zur LED-Visualisierung und deren Parametrierung finden Sie in der Betriebsanleitung der CRG-Greifmodule.

Über die Einstellungen des Nodes „GRIPLINK LED“ können Port des gewünschten Greifers und der Index des Presets ausgewählt werden.



Die LED-Funktion ist nur bei kompatiblen Geräten verfügbar!

6.15.1 Befehlsaufruf mit Script-Code

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

Parameter	Typ	Bedeutung
<PORT>	Integer	Port-Index (0..31)
<PRESET_INDEX>	Integer	Preset-Index (0..7)
<SOCKET_NAME>	String	Name des zu verwendenden Sockets, bspw. "sock_griplink" Der Socket-Name wird in allen nachfolgenden Knoten im Roboterprogramm verwendet!

Beispiel:

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

6.16 Konfigurieren eines Griff-Presets – GRIPLINK Set Grip Config

Abhängig vom Greifmodul können bis zu acht frei konfigurierbare Griffe ausgeführt werden. Mit dem Node „GRIPLINK Set Grip Config“ können im laufenden Betrieb die Griffparameter angepasst werden, um beispielsweise zu Beginn eines Programms ein Greifmodul zu initialisieren oder flexibel auf Änderungen im Ablauf zu reagieren.



Weitere Informationen zu den Griffparametern, insbesondere der Limitierungen, finden Sie in den Betriebsanleitungen der jeweiligen Greifmodule.

In der Übersicht des Nodes „GRIPLINK Set Grip Config“ können Port des gewünschten Greifers und der Index des Presets ausgewählt werden.

Über das Dropdown-Feld „Value Interpretation“ kann die Darstellung der Werte angepasst werden, je nach dem welche Bedeutung die einzelnen Parameter haben. Für bestimmte Geräte werden nicht-relevante Parameter ausgeblendet.

Für Greifmodule von Weiss Robotics können No Part-Limit und Release-Limit, sowie die prozentuale Greifkraft (Force Factor) vorgegeben werden



Für Greifmodule von Weiss Robotics ist das Parameter Template „Weiss Robotics“ zu wählen, damit die eingegebenen Werte korrekt geprüft und dargestellt werden.

PROGRAM Core default
INSTALLATION New... Open... Save...

Command Graphics Variables

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

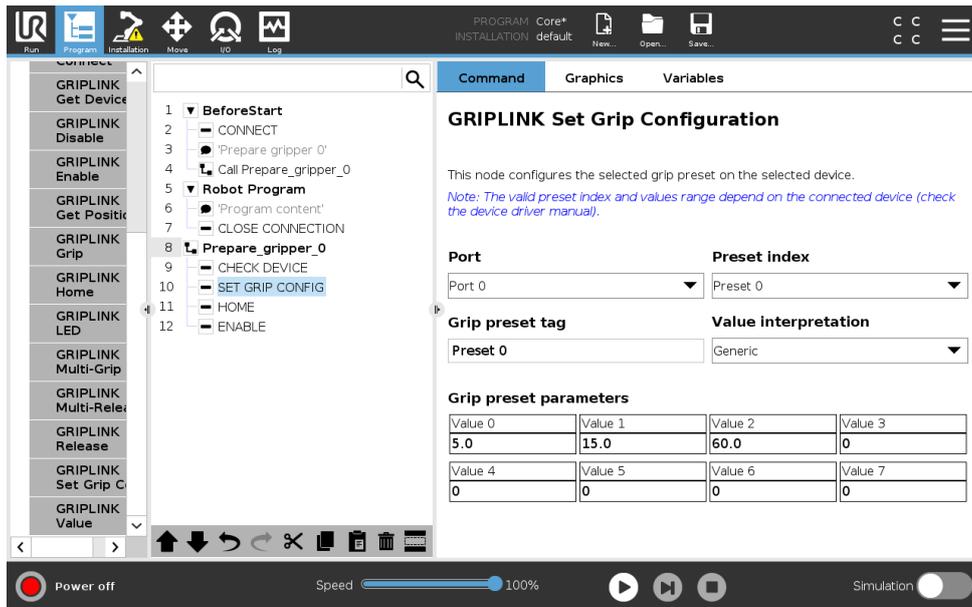
No Part Limit / mm	Release Limit / mm	Force Factor / %
5.0	15.0	60.0

Power off Speed 100% Simulation



Beachten Sie, dass abhängig vom Greifmodul nicht immer alle acht konfigurierbare Griffe zur Verfügung stehen!

Für andere Geräte kann die generische Ansicht verwendet werden. Hier sind alle acht verfügbaren Parameter sichtbar auch wenn gegebenenfalls nicht jeder vom Gerät verwendet wird.



6.16.1 Befehlsaufruf mit Script-Code

```

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

```

Parameter	Typ	Bedeutung
<PORT>	Integer	Port-Index (0..31)
<PRESET_INDEX>	Integer	Preset-Index (0..7)
<PRESET_PARAMS>	Integer List	Ausgewählte Ports als Liste mit skalierten Werten Liste umfasst mindestens ein und maximal 8 Elemente
<SOCKET_NAME>	String	Name des zu verwendenden Sockets, bspw. "sock_griplink" Der Socket-Name wird in allen nachfolgenden Knoten im Roboterprogramm verwendet!

Beispiel:

```

# 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")

```

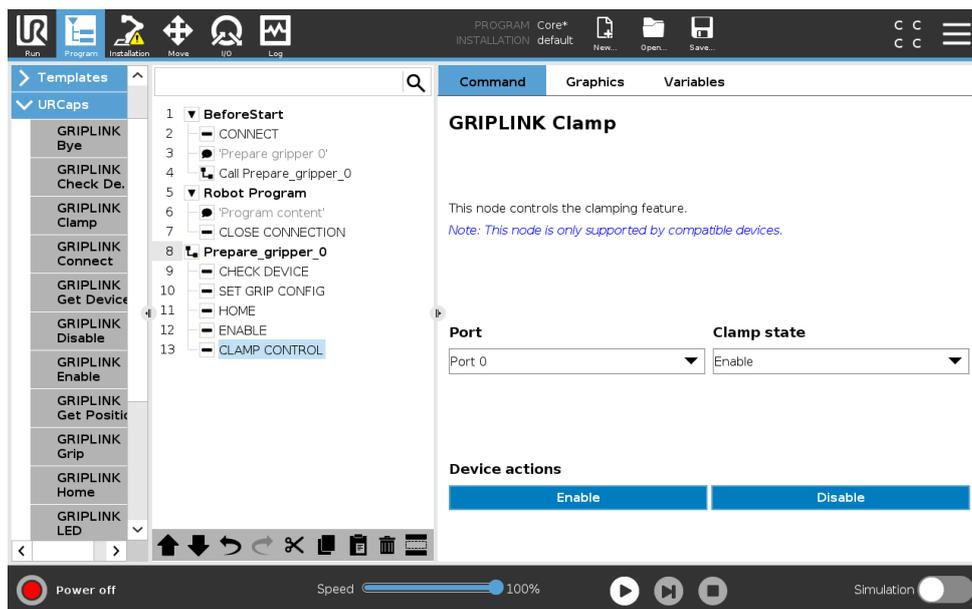
6.17 Mechanische Klemmung steuern – GRIPLINK Clamp

Die Greifer der CRG-Serie von Weiss Robotics sind mit dem PERMAGRIP-Feature ausgestattet. Damit ist es möglich, die Motorregelung während längerem Greifen von Bauteilen abzuschalten und das Bauteil dennoch weiterhin sicher zu halten.

Um diese Klemmung zu aktivieren, wird der Node „GRIPLINK Clamp“ verwendet.



Weitere Informationen zur mechanischen Klemmung, finden Sie in der Betriebsanleitung des jeweiligen Greifmoduls.



6.17.1 Befehlsaufruf mit Script-Code

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

Parameter	Typ	Bedeutung
<PORT>	Integer	Port-Index (0..31)
<CLAMP_STATE>	Boolean	„True“, um Klemmung zu aktivieren „False“, um Klemmung zu deaktivieren
<SOCKET_NAME>	String	Name des zu verwendenden Sockets, bspw. "sock_griplink" Der Socket-Name wird in allen nachfolgenden Knoten im Roboterprogramm verwendet!

Beispiel:

```
# 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 Gerätezustand

Die folgende Tabelle listet die möglichen Zustandswerte angeschlossener Geräte auf. Die in der hinteren Spalte angegebenen Konstanten kann im Roboterprogramm verwendet werden (siehe Abschnitt 6.1).

Gerätezustand	Wert	Bedeutung	Name der URScript Konstante
NOT CONNECTED	0	Greifmodul nicht verbunden	S_NOT_CONNECTED
NOT INITIALIZED	1	Greifmodul nicht initialisiert	S_NOT_INITIALIZED
IDLE	2	Antrieb inaktiv Finger können manuell verschoben werden Gerät deaktiviert	S_DISABLED
RELEASED	3	Werkstück freigegeben	S_RELEASED
NO PART	4	Kein Werkstück gefunden	S_NO_PART
HOLDING	5	Werkstück wird gehalten	S_HOLDING
ENABLED	6	Antrieb aktiv Fingerposition wird gehalten Gerät aktiviert	S_OPERATING
FAULT	7	Fehlerzustand	S_FAULT



Weisen Sie den UR-Script-Konstanten keine anderen als die in der Tabelle gelisteten Werte zu, da ansonsten die Funktion des URCaps beeinträchtigt wird und es zu Fehlverhalten kommen kann!

© 2024 WEISS ROBOTICS GmbH & Co. KG. Alle Rechte vorbehalten.

GRIPLINK und PERMAGRIP sind eingetragene Marken der WEISS ROBOTICS GmbH & Co. KG. Alle weiteren Marken sind Eigentum ihrer jeweiligen Inhaber.

Die in diesem Dokument angegebenen technischen Daten können zum Zwecke der Produktverbesserung ohne Vorankündigung geändert werden. Warenzeichen sind Eigentum des jeweiligen Eigentümers. Unsere Produkte sind nicht für den Einsatz in lebenserhaltenden Systemen oder für Systeme, bei denen ein Fehlverhalten zu Personenschäden führen könnte, vorgesehen.