



GRIPLINK-PLUGIN FÜR UNIVERSAL ROBOTS

Version 3.0.0

April 2026



Inhaltsverzeichnis

1	Einführung.....	3
1.1	Notation und Symbole	3
1.2	Bestimmungsgemäße Verwendung.....	3
1.3	Systemvoraussetzungen	4
1.4	Lizenzbestimmungen	5
1.5	Demo-Programme	5
2	Installation	7
2.1	Installation der Software	7
2.1.1	Überprüfen der Installation	9
2.2	Deinstallieren der Software	9
2.3	Verhalten im Fehlerfall	10
3	Hardware-Setup.....	10
3.1	Montage am Roboterflansch	11
3.2	Stromversorgung	11
3.3	Tool-I/O-Topologie.....	11
4	Funktionsumfang des Plugins	13
4.1	Aufbau des Plugins.....	13
4.2	Betrieb mittels Programmknoten	13
4.3	Betrieb mittels URScript	13
5	Vorbereitung des Roboters.....	14
5.1	Setup	14
5.1.1	Ethernet-Schnittstelle	14
5.1.2	Tool-I/O-Schnittstelle.....	16
5.1.2.1	Inbetriebnahme der Tool-I/O-Schnittstelle	18
5.1.2.2	Werkzeugwechsel an der Tool-I/O-Schnittstelle	19
5.1.3	Port-Konfiguration	19
5.1.4	Logging-Konfiguration.....	20
6	GRIPLINK Toolbar	22
6.1	Aufbau der Toolbar	23
6.2	IGRIP-Tab - Arbeiten mit Grip-Presets	24
6.3	PGRIP-Tab - Flexgrip/Flexrelease	26
6.4	VALUE-Tab - Gerätedaten beobachten.....	27
6.5	Hinweise zur Verwendung der Toolbar	28

7	Programmknotten (Nodes)	29
7.1	Prinzipieller Programmablauf	29
7.1.1	Initialisierungsabschnitt (z. B. BeforeStart)	29
7.1.2	Typischer Ablauf pro Gerät	29
7.1.3	Hauptprogrammabschnitt	30
7.2	Verbindung aufbauen - GRIPLINK Connect.....	31
7.3	Ein angeschlossenes Gerät prüfen - GRIPLINK Check Device	33
7.4	Gerät referenzieren - GRIPLINK Home.....	35
7.5	Gerät aktivieren und deaktivieren - GRIPLINK Enable/Disable.....	37
7.6	Greifen und Freigeben - GRIPLINK Grip/Release	40
7.6.1	Auswertung des Greifzustands	44
7.7	Flexibles Greifen, Freigeben und Vorpositionieren - GRIPLINK Flexgrip/Flexrelease.....	46
7.8	Zustandsabfrage - GRIPLINK Get Dev State	50
7.9	Gerätewerte auslesen und warten - GRIPLINK Value/Wait.....	56
7.9.1	Gerätewert lesen	56
7.9.2	Gerätewert warten und lesen.....	58
7.10	Konfigurieren eines Grip-Presets - GRIPLINK Set Grip Config.....	62
7.11	Gerätewert setzen - GRIPLINK Set Value	66
7.12	Ansteuerung des LED-Leuchtrings - GRIPLINK LED	67
7.13	Mechanische Klemmung steuern - GRIPLINK Clamp	68
8	Fehlersuche.....	69
8.1	Kein Zugriff auf die Tool-I/O-Schnittstelle	69
8.2	Gerät wird im Tool-I/O-Betrieb nicht erkannt	69
8.3	Timeout bei aktivierter Option „Wait for state transitions“	69
8.4	Nicht blockierende Ausführung bei GRIPKIT EASY Firmware-Versionen unter 3.0.0.....	70
Anhang A	Gerätezustand.....	71
Anhang B	Statuscodes.....	72
Anhang C	Tool-I/O-spezifische Statuscodes.....	74
Anhang D	Versionsverlauf	75

1 Einführung

Mit der GRIPLINK-Technologie können Automationskomponenten von WEISS ROBOTICS über eine standardisierte Kommunikationsschnittstelle in Robotersysteme integriert werden. Das GRIPLINK-Plugin für Universal Robots ist das steuerungsseitige Bindeglied und ermöglicht die Anbindung sowie die Bedienung kompatibler Geräte von WEISS ROBOTICS an Robotersystemen des Herstellers Universal Robots.

Das GRIPLINK-Plugin unterstützt sowohl Geräte, die das GRIPLINK-Protokoll nativ über TCP/IP verwenden, als auch kompatible Modbus-RTU-Geräte, die über die Tool-I/O-Schnittstelle der UR-Steuerung angebunden werden.

Der Begriff „GRIPLINK-Controller“ wird in dieser Anleitung stellvertretend für alle Produkte verwendet, die die GRIPLINK-Technologie integriert haben. Dazu zählen sowohl eigenständige GRIPLINK-Controller (z. B. GRIPLINK-ET4) als auch Geräte mit eingebetteter GRIPLINK-Funktionalität, wie etwa WPG- oder INTRAPAL-Servogreifer von WEISS ROBOTICS. Alle diese Produkte implementieren dasselbe GRIPLINK-Protokoll und können daher über das Plugin auf identische Weise angesteuert werden.



Diese Anleitung beschreibt die Funktionen des GRIPLINK-Plugins. Informationen über Montage, Inbetriebnahme und Betrieb des jeweiligen Geräts entnehmen Sie bitte der zugehörigen Betriebsanleitung. Weiterführende Informationen und Dokumente finden Sie unter <https://weiss-robotics.com/>

1.1 Notation und Symbole

Zur besseren Übersicht werden in dieser Anleitung folgende Symbole verwendet:



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



Zusatzinformation zum besseren Verständnis des beschriebenen Sachverhalts.



Verweis auf weiterführende Informationen.

1.2 Bestimmungsgemäße Verwendung

Die Software „GRIPLINK-Plugin“ ist zur Kommunikation zwischen kompatiblen Geräten von WEISS ROBOTICS und einer Robotersteuerung von Universal Robots bestimmt. Sie ermöglicht die Integration und Steuerung unterstützter Geräte über die vorgesehenen Kommunikationsschnittstellen Ethernet und Tool-I/O. Die Anforderungen der zutreffenden Richtlinien sowie die Installations- und

Betriebshinweise in dieser Anleitung müssen beachtet und eingehalten werden. Eine andere oder darüber hinausgehende Verwendung gilt als nicht bestimmungsgemäß. Für hieraus resultierende Schäden haftet der Hersteller nicht.

1.3 Systemvoraussetzungen

Für den Ethernet-Betrieb ist dieses Plugin mit GRIPLINK-Protokollversion 3 und höher kompatibel. Für den Tool-I/O-Betrieb gelten die nachfolgend aufgeführten geräte- und firmwareabhängigen Voraussetzungen.

Für den Betrieb werden folgende Universal Robots Robotersteuerungen unterstützt:

- **Ethernet:** UR e-Series mit Softwarestand 5.5 (oder höher) sowie UR CB3.1 mit Softwarestand 3.11 (oder höher).
- **Tool-I/O (Modbus RTU):** UR e-Series mit Softwarestand 5.4 (oder höher).

Schnittstellen	Unterstützte Geräte / Firmware	Lizenz
Ethernet	<ul style="list-style-type: none"> • GRIPLINK-ET4 ab FW-Version 5.2.0 • WPG-Serie ab FW-Version 2.4.0 • INTRAPAL-Serie ab FW-Version 5.2.0 	-
Tool-I/O (Modbus RTU)	<ul style="list-style-type: none"> • GRIPKIT EASY ab FW-Version 2.1.1 • GRIPKIT LS ab FW-Version 1.1.0 • INTRAPAL ab FW-Version 5.3.0 	<ul style="list-style-type: none"> • OPT-GKEASY-MB für GRIPKIT EASY • OPT-IPL-MB für INTRAPAL



Die IP-Adresse des GRIPLINK-Controllers muss im selben Subnetz liegen wie die der Robotersteuerung. In der Anleitung des GRIPLINK-Controllers ist der genaue Vorgang beschrieben, wie Sie die IP-Adresse ändern.



Die Tool-I/O-Schnittstelle wird von Universal Robots nur ab der e-Series unterstützt. Robotersteuerungen der CB-Series, einschließlich CB3.1, unterstützen keine Tool-I/O-Schnittstelle. Auf diesen Systemen kann das GRIPLINK-Plugin daher ausschließlich im Ethernet-Betrieb verwendet werden.



Wichtiger Hinweis: Eine Aktualisierung des URCaps kann Auswirkungen auf bestehende Systeme, Konfigurationen und Roboterprogramme haben. Dadurch können Anpassungen an der Anwendung erforderlich werden. Prüfen Sie daher vor jeder Aktualisierung sorgfältig die Systemvoraussetzungen sowie die Änderungen gegenüber vorigen Versionen.



Es wird empfohlen, nach der Migration eine vollständige Funktionsprüfung der Anwendung durchzuführen.

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!

ExampleBasicGripCycle

Einfacher Greifzyklus mit GRIPLINK GRIP und GRIPLINK RELEASE. Nach dem Greifen wird über die Verzweigung HOLDING/NO PART eine Meldung angezeigt. Anschließend wird je nach Ergebnis entweder GRIPLINK RELEASE und GRIPLINK DISABLE oder nur GRIPLINK DISABLE ausgeführt.

ExampleFlexGripCycle

Demonstriert flexibles Greifen mit GRIPLINK FLEXGRIP und GRIPLINK FLEXRELEASE. Nach dem Befehl FLEXGRIP wird der Gerätestatus über GRIPLINK GET DEV STATE gelesen. Abhängig vom Gerätestatus wird entweder FLEXRELEASE und DISABLE ausgeführt oder das Programm mit einem Halt abgebrochen (z. B. bei Fehler bzw. unerwartetem Zustand).

ExampleAsyncGrip

Zeigt, wie mit `GL_GRIP(..., ..., do_block = 0, ...)` ein Greifvorgang asynchron gestartet wird. Der Greifstatus wird anschließend über den Programmknoten GRIPLINK GET DEV STATE mit aktivierter Option „Warten auf Zustandsübergang (WSTR)“ geprüft. Je nach Ergebnis verzweigt das Programm zu GRIPLINK RELEASE und DISABLE, nur DISABLE oder einem Halt, wenn ein Fehlerzustand vorliegt.

ExampleMultiGrip

Beispiel für paralleles Greifen mit drei Greifern an Port 0, 1 und 2. Zunächst werden die Greifer mit GRIPLINK ENABLE und GRIPLINK HOME initialisiert. Anschließend wird ein quasi-paralleler GRIPLINK GRIP mit `GL_GRIP(..., ..., do_block = 0, ...)` gestartet. Danach werden die Zustände über GRIPLINK GET DEV STATE mit aktivierter Option „Warten auf Zustandsübergang (WSTR)“ geprüft. Je nach Ergebnis wird unterschieden in:

- Mindestens ein Greifer befindet sich in S_NO_PART
- Alle Greifer befinden sich in S_NO_PART.
- Alle Greifer befinden sich in S_HOLDING.

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_3.0.0.zip“ herunter.
2. Entpacken Sie das zuvor heruntergeladene ZIP-Archiv des GRIPLINK-Plugins in das Stammverzeichnis eines USB-Sticks und stecken Sie diesen in den USB-Anschluss des Teach Pendants.
3. Öffnen Sie die Einstellungen und navigieren Sie in das Menü „System/URCaps“.

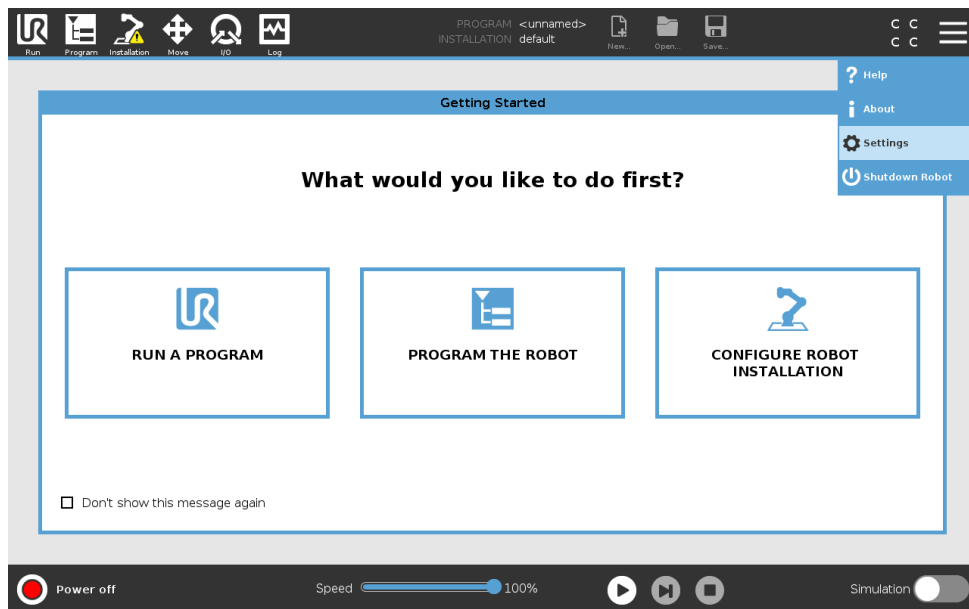


Abbildung 1: PolyScope 5 - Getting Started.

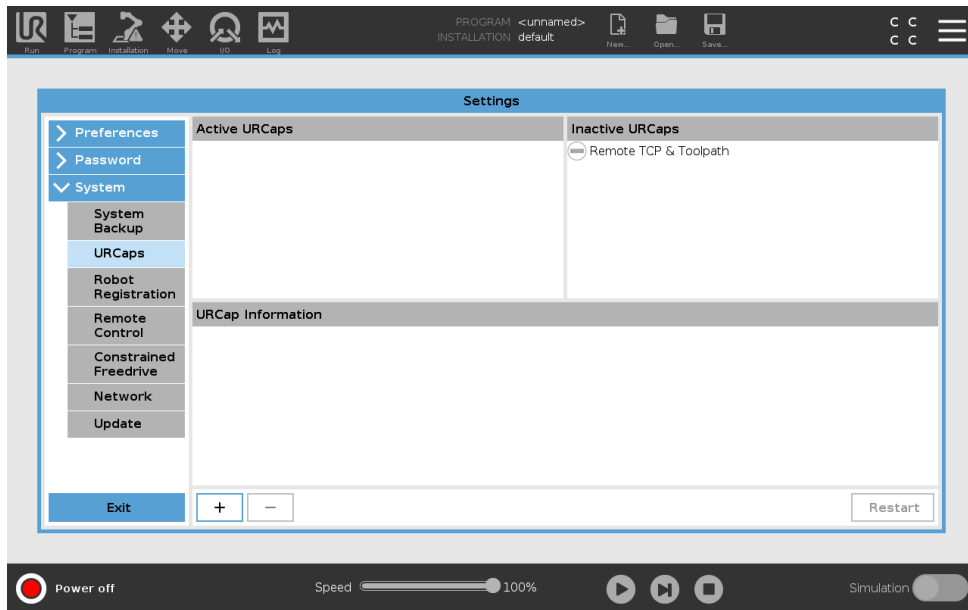


Abbildung 2: PolyScope 5 - Settings/System/URCaps.

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

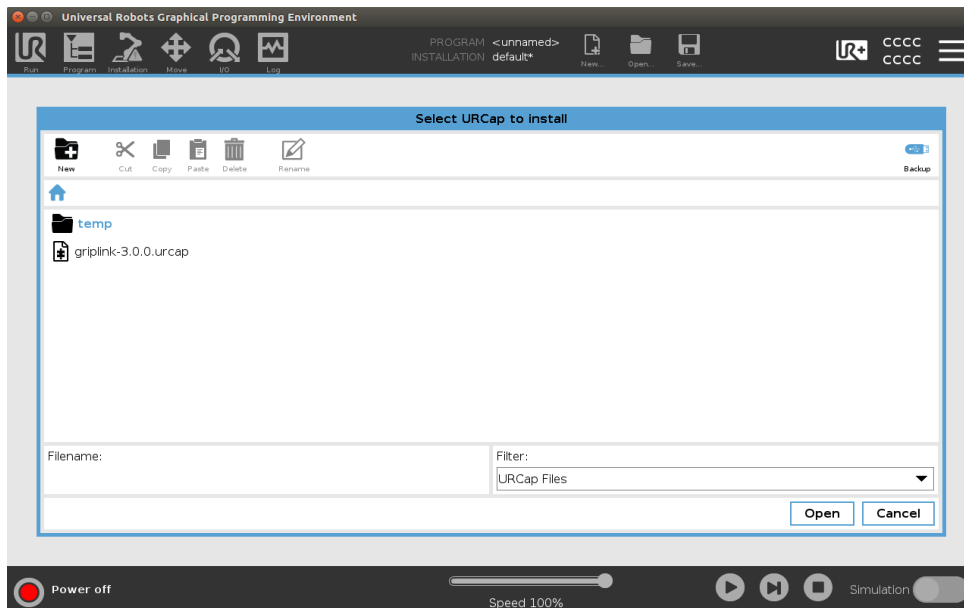


Abbildung 3: PolyScope 5 - Select URCap to install.

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

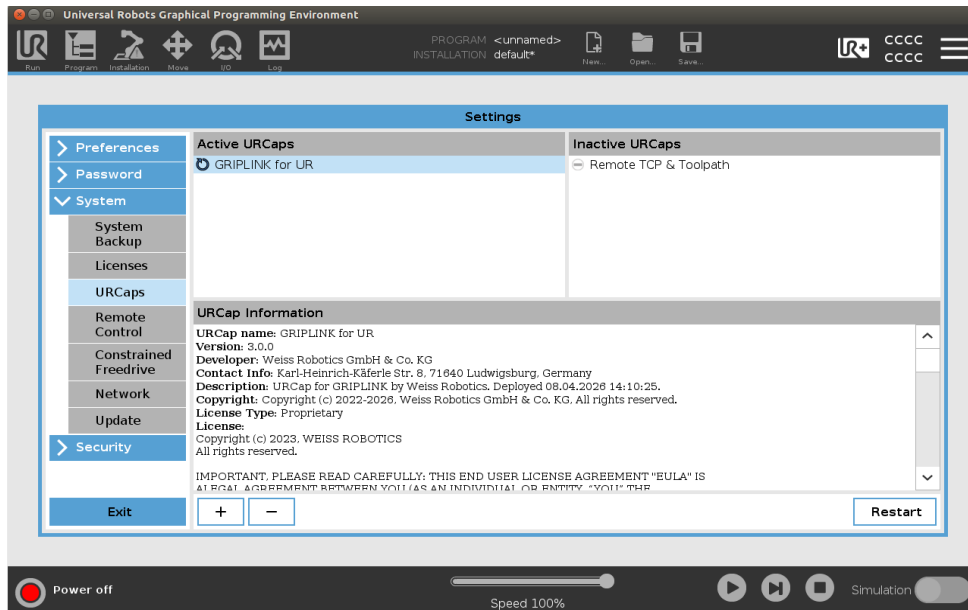


Abbildung 4: PolyScope 5 - Active URCaps: GRIPLINK for UR.

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-Programmknotten.

2.2 Deinstallieren der Software

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

2.3 Verhalten im Fehlerfall

Tritt innerhalb des GRIPLINK-Plugins oder während der Kommunikation mit einem angeschlossenen Gerät ein Fehler auf, wird die Ausführung des laufenden Roboterprogramms mit einer Fehlermeldung abgebrochen. In der Folge können auch laufende Roboterbewegungen gestoppt werden. Dadurch wird verhindert, dass der Roboter seine Bewegung fortsetzt, obwohl ein Kommunikations- oder Gerätefehler vorliegt.

Ein Fehler kann beispielsweise auftreten, wenn die Verbindung zum GRIPLINK-Controller unterbrochen wird, ungültige Parameter übergeben werden oder ein angeschlossenes Gerät einen Fehlerzustand meldet. Dasselbe gilt, wenn sich das angesprochene Gerät bereits im Zustand FAULT befindet oder während der Ausführung eines Befehls in diesen Zustand wechselt.

Zusätzliche Diagnosehinweise können der Zustandsanzeige im Installationsknoten sowie – bei aktiviertem Logging – den Logdateien im Verzeichnis `~/griplink` auf der Robotersteuerung entnommen werden. Eine Übersicht typischer Fehlerursachen und empfohlener Maßnahmen befindet sich in **Kapitel 8 „Fehlersuche“**.

3 Hardware-Setup

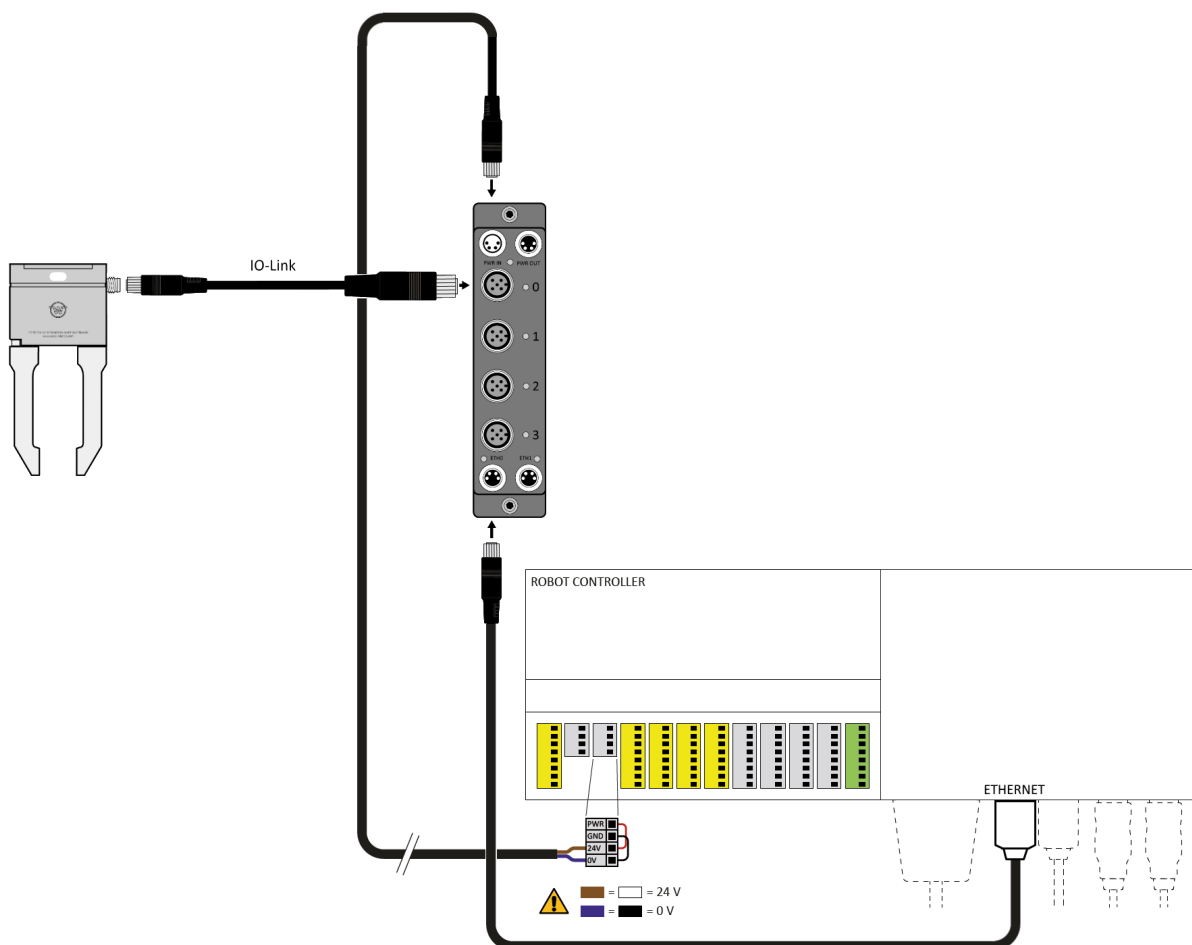


Abbildung 5: IO-Link-Greifer über GRIPLINK-Controller an die Robotersteuerung (Ethernet).



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!

3.1 Montage am Roboterflansch

Die mechanische Montage am Roboterflansch ist in der Betriebsanleitung des jeweiligen Geräts beschrieben.



Montage-, Demontage- und Verdrahtungsarbeiten dürfen nur im spannungsfreien Zustand durchgeführt werden. Vor dem Anschließen, Trennen oder Wechseln eines Tool-I/O-Geräts ist sicherzustellen, dass an der Tool-I/O-Schnittstelle **0 V** anliegen.

3.2 Stromversorgung

Bei der Verwendung mehrerer Geräte ist sicherzustellen, dass die Spannungsversorgung ausreichend dimensioniert ist. Nur so kann ein ordnungsgemäßer und störungsfreier Betrieb aller angeschlossenen Geräte gewährleistet werden.



Die zulässige Stromaufnahme der Tool-I/O-Schnittstelle des Roboters darf durch angeschlossene Geräte nicht überschritten werden. Eine Überschreitung kann zu Störungen oder Schäden am Robotersystem sowie an den angeschlossenen Geräten führen.



Beachten Sie hierzu die Angaben in der Bedienungsanleitung des Roboters zu den zulässigen Grenzwerten der Stromaufnahme sowie die Angaben zur Stromaufnahme in der Betriebsanleitung des jeweiligen Geräts.

3.3 Tool-I/O-Topologie

Im Tool-I/O-Betrieb erfolgt die Kommunikation über die Tool-I/O-Schnittstelle des Roboters. Die in diesem Abschnitt beschriebene Portabbildung gilt für die Tool-I/O-Integration des URcaps und ist von der generischen Portdefinition des GRIPLINK Unified Command Set zu unterscheiden. Die logische Portabbildung basiert auf der jeweiligen Modbus-Station-Adresse. Jede Station-Adresse bildet genau einen Master mit einem Hauptport sowie bis zu drei zugehörigen Unterports ab.

Bei Greifmodulen mit Fingerschnittstelle werden die Finger den zugehörigen Unterports des jeweiligen Hauptports zugeordnet. Befindet sich ein Gerät beispielsweise am Hauptport 0, werden Finger 0 und

Finger 1 als Port 1 und Port 2 abgebildet. Bei einem Gerät am Hauptport 4 erfolgt die Zuordnung entsprechend zu Port 5 und Port 6.

Hauptport	Stationsadresse (Modbus RTU)	Zugeordnete Unterports
0	1	1, 2, 3
4	2	5, 6, 7
8	3	9, 10, 11
12	4	13, 14, 15



Bei Verwendung mehrerer Geräte ist auf eine korrekte Verdrahtung und eine zur Installation passende Konfiguration der Kommunikationsschnittstelle zu achten.



Die Tool-I/O-Schnittstelle darf nur mit dem Original-Tool-I/O-Kabel des jeweiligen Geräts betrieben werden. Andere Kabel, Applikationen oder Verdrahtungen können die Datenverbindung beeinträchtigen und zu Schäden führen.

4 Funktionsumfang des Plugins

4.1 Aufbau des Plugins

Das GRIPLINK-Plugin besteht aus dem Installationsknoten zur Grundkonfiguration, der GRIPLINK-Toolbar für Inbetriebnahme und Diagnose sowie den Programmknoten und URScript-Funktionen für die Verwendung im Roboterprogramm.

Die grafischen Programmknoten und die URScript-Funktionen greifen auf dieselbe Kommunikationsschnittstelle zu. Dadurch können unterstützte Geräte sowohl über die Benutzeroberfläche von PolyScope als auch direkt aus Skriptcode angesprochen werden. Die Toolbar dient ausschließlich der manuellen Bedienung, Diagnose und Inbetriebnahme; Änderungen am eigentlichen Prozessablauf müssen im Roboterprogramm vorgenommen werden.

4.2 Betrieb mittels Programmknoten

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

4.3 Betrieb mittels URScript

Das Plugin umfasst einen Satz an Grundfunktionen in URScript, die auch eigenständig aus dem Roboterprogramm mittels des entsprechenden Skript-Programmknotts aufgerufen werden können. Diese Grundfunktionen werden automatisch in das Roboterprogramm eingebaut, wenn das Plugin installiert ist. In Abschnitt 7 werden die verfügbaren Funktionen genauer beschrieben.

5 Vorbereitung des Roboters

Die grundlegenden Einstellungen des GRIPLINK-Plugins werden im Installationsknoten vorgenommen.



Im Ethernet-Betrieb erfolgt die Konfiguration des GRIPLINK-Controllers über dessen Weboberfläche, die unter der eingestellten IP-Adresse (Standard: 192.168.1.40) erreichbar ist.



Im Tool-I/O-Betrieb kann die gerätespezifische Parametrierung von GRIPKIT EASY bei Bedarf über die Applikation „GRIPKIT Configurator“ erfolgen.

5.1 Setup

5.1.1 Ethernet-Schnittstelle

Im Feld „GRIPLINK-Controller IP Address“ wird die IP-Adresse des am Roboter angeschlossenen GRIPLINK-Controllers eingetragen. Die Standard-IP-Adresse ist 192.168.1.40 und befindet sich im Subnetz 192.168.1.0/24. Diese IP-Adresse kann an das jeweilige Anlagennetzwerk angepasst werden. Die Robotersteuerung und der GRIPLINK-Controller müssen sich im selben Subnetz befinden.

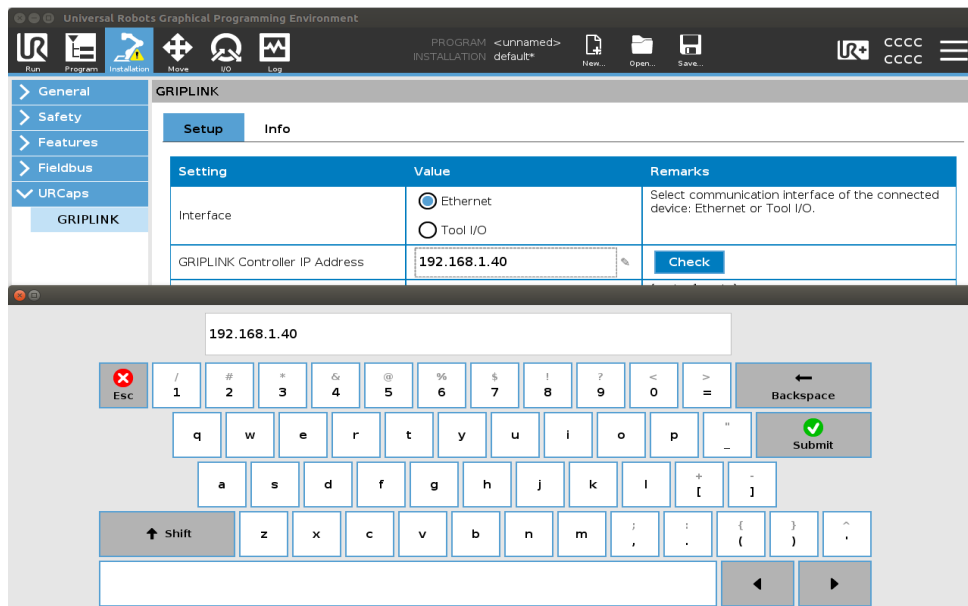


Abbildung 6: PolyScope 5 - Installation/URCaps/GRIPLINK/Setup (IP-Adresse ändern).

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

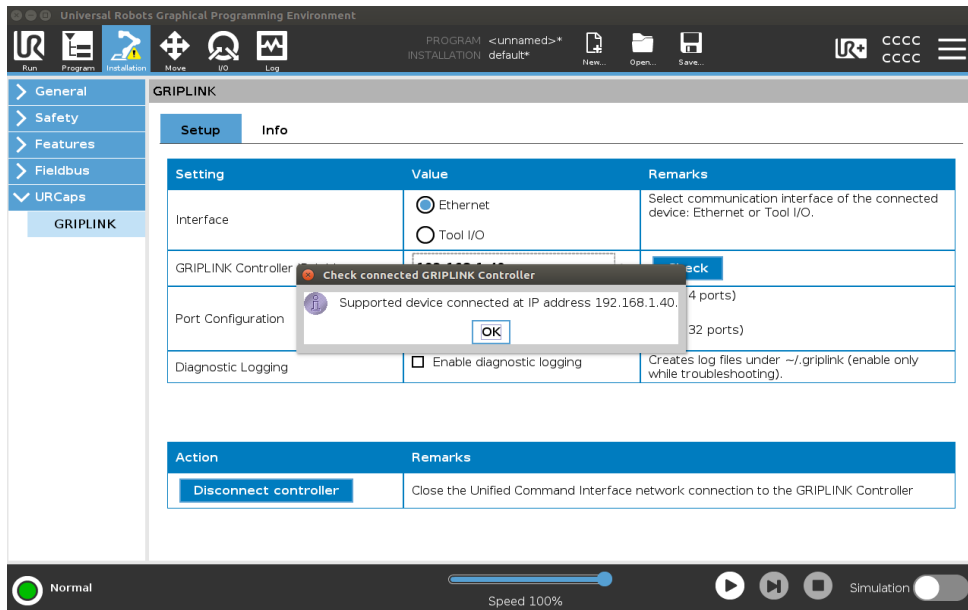


Abbildung 7: Installation/URCaps/GRIPLINK/Setup: Check connected GRIPLINK-Controller.

Ist eine Verbindung zwischen Robotersteuerung und GRIPLINK-Controller aktiv, können keine Greifbefehle über die Action-Buttons der Programmknoten ausgeführt werden. Um die Verbindung zwischen Robotersteuerung und GRIPLINK-Controller zu trennen, kann der Button „Disconnect“ verwendet werden.

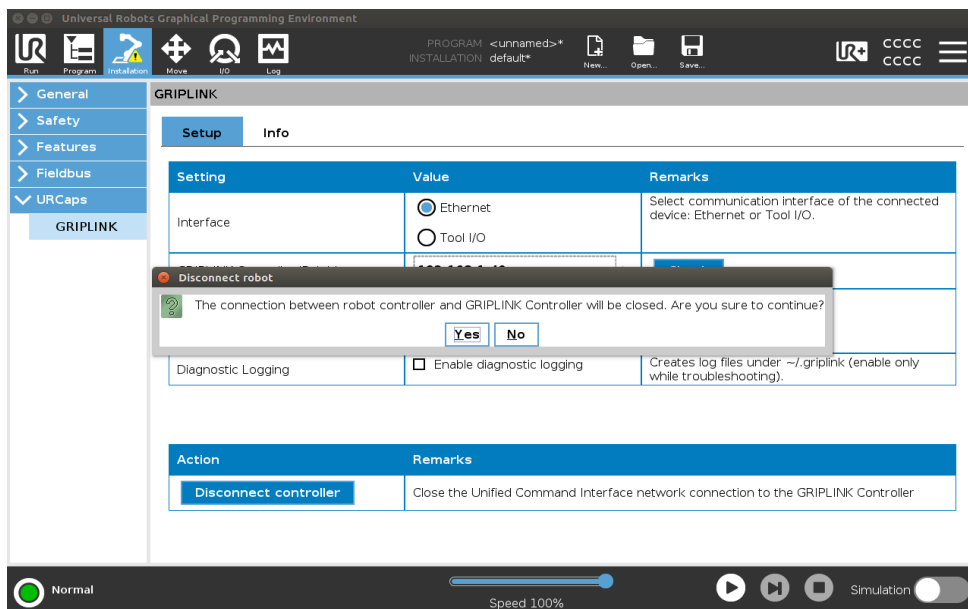


Abbildung 8: PolyScope 5 - Installation/URCaps/GRIPLINK/Setup: Verbindung trennen (Bestätigungsdialog).



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

5.1.2 Tool-I/O-Schnittstelle

Um die Tool-I/O-Schnittstelle nutzen zu können, muss im Installation-Tab von PolyScope die Tool-I/O-Schnittstelle korrekt konfiguriert werden. Dazu wählen Sie im Dropdown „Ansteuerung durch“ die Option „GRIPLINK for UR“ aus.

Für den Tool-I/O-Betrieb werden folgende Kommunikationsparameter automatisch durch das GRIPLINK-Plugin eingestellt: Baud Rate 115200, Parity None, Stop Bits One, RX Idle Chars 1.5 und TX Idle Chars 3.5.

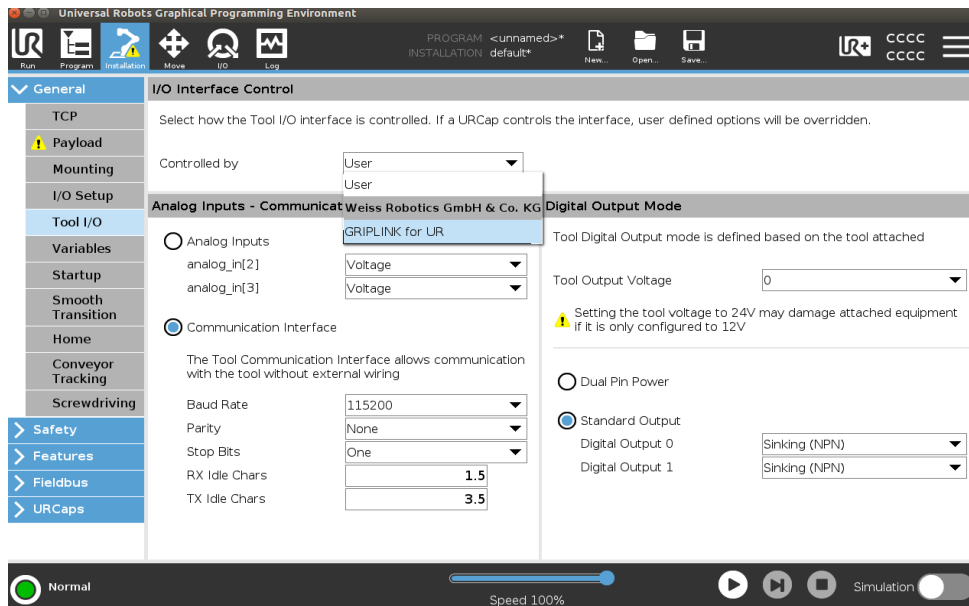


Abbildung 9: PolyScope 5 – Konfiguration der Tool-I/O-Schnittstelle mit „GRIPLINK for UR“ als ausgewählter Ansteuerung.



Wird die Schnittstelle nicht korrekt konfiguriert, können die angeschlossenen Geräte nicht über das Plugin angesteuert werden.



Während des Betriebs darf ausschließlich das GRIPLINK-Plugin auf die Tool-I/O-Schnittstelle zugreifen. Die gleichzeitige Verwendung weiterer URCaps oder Funktionen, die ebenfalls die Tool-I/O-Schnittstelle ansteuern, kann zu Konflikten, Kommunikationsfehlern oder einem unzuverlässigen Betrieb führen.

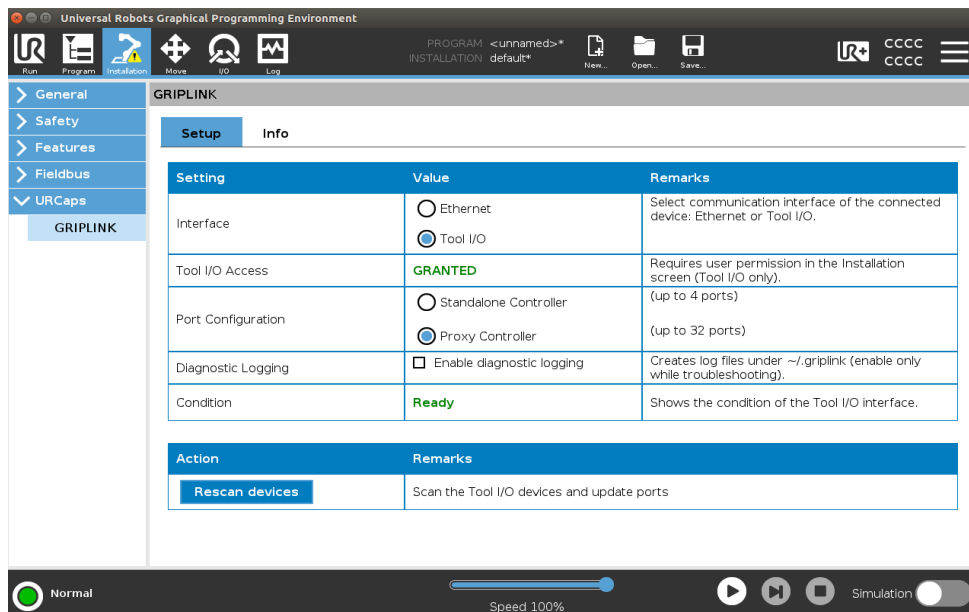


Abbildung 10: Tool-I/O-Schnittstelle mit erteiltem Zugriff und Zustandsanzeige.

Im Installationsknoten werden zwei interaktive Rückmeldungen angezeigt: der Tool-I/O-Zugriff und die Zustandsanzeige. Diese Zustandsanzeige informiert über den aktuellen Betriebszustand der Tool-I/O-Schnittstelle. Die nachfolgenden Statuswerte entsprechen den im Plugin angezeigten Originalbezeichnungen.

Tool-I/O-Zugriff	Bedeutung
NOT_GRANTED	Der Benutzer hat in den Tool-I/O-Einstellungen des Roboters nicht „GRIPLINK for UR“ ausgewählt. Das Plugin besitzt damit keinen Zugriff auf die Tool-I/O-Schnittstelle.
GRANTED	„GRIPLINK for UR“ ist ausgewählt. Das Plugin darf auf die Tool-I/O-Schnittstelle zugreifen.

Zustandsanzeige	Bedeutung
Tool-I/O Interface Error	Fehler in der Tool-I/O-Schnittstelle. Prüfen Sie Zugriff, Verdrahtung und angeschlossenes Gerät.
Waiting for Tool-I/O Access	Der Zugriff auf Tool-I/O wurde noch nicht erteilt. Wählen Sie in den Tool-I/O-Einstellungen „GRIPLINK for UR“ aus.
Scanning devices	Die Gerätesuche wird aktuell ausgeführt. Nach Abschluss des Suchvorgangs werden die erkannten Geräte einschließlich Portzuordnung aktualisiert angezeigt.
Waiting for device	Es wurde noch kein Gerät gefunden. Schließen Sie ein kompatibles Gerät an und führen Sie anschließend einen Rescan aus.
Ready	Mindestens ein kompatibles Gerät wurde gefunden; die Tool-I/O-Schnittstelle ist betriebsbereit.

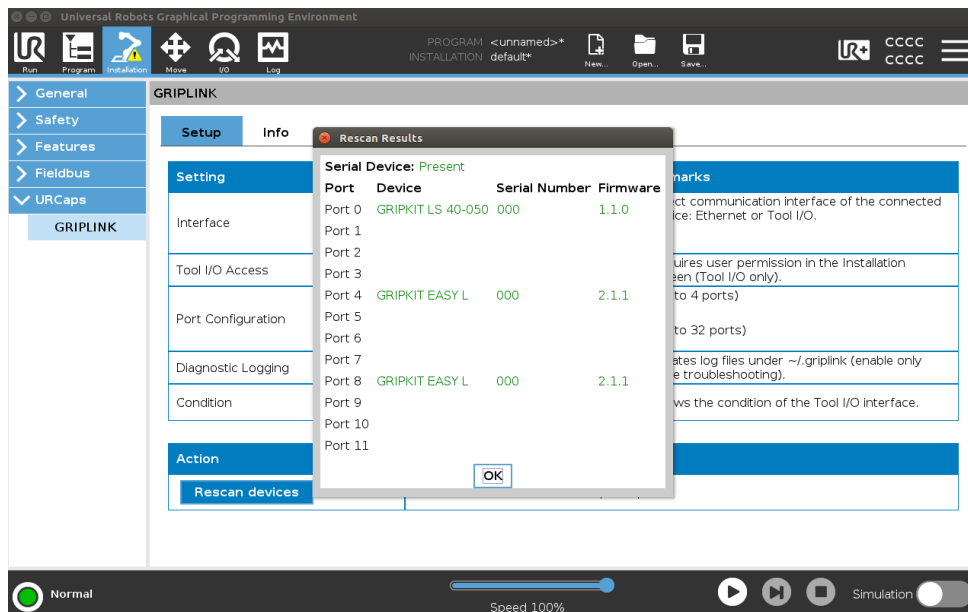


Abbildung 11: Rescan-Ergebnis in der Tool-I/O-Schnittstelle mit erkannten Geräten, Portzuordnung, Seriennummern und Firmware-Versionen.

Mit der Aktion „Rescan devices“ wird eine erneute Gerätesuche an der Tool-I/O-Schnittstelle durchgeführt. Dabei werden die aktuell erkannten Geräte einschließlich Portzuordnung, Seriennummer und Firmware-Version im Dialogfenster dargestellt. Ein Rescan ist insbesondere nach der ersten Inbetriebnahme, nach einem Werkzeugwechsel oder dann sinnvoll, wenn ein angeschlossenes Gerät noch nicht erkannt wurde.

5.1.2.1 Inbetriebnahme der Tool-I/O-Schnittstelle

Für die Inbetriebnahme der Tool-I/O-Schnittstelle wird folgende Reihenfolge empfohlen:

1. Sicherstellen, dass an der Tool-I/O-Schnittstelle 0 V anliegen, bevor das Gerät angeschlossen wird.
2. Ein kompatibles Tool-I/O-Gerät mechanisch montieren und das Original-Tool-I/O-Kabel anschließen.
3. In den Tool-I/O-Einstellungen des Roboters unter „Ansteuerung durch“ die Option „**GRIPLINK for UR**“ auswählen.
4. Im Installationsknoten des GRIPLINK-Plugins die Kommunikationsschnittstelle „**Tool-I/O**“ auswählen.
5. Die Installation speichern, nachdem im GRIPLINK-Plugin die Kommunikationsschnittstelle „**Tool-I/O**“ ausgewählt wurde.
6. Prüfen, ob im Installationsknoten des GRIPLINK-Plugins der Zugriff den Status „**GRANTED**“ anzeigt.
7. Falls noch kein Gerät erkannt wird, „**Rescan devices**“ ausführen.
8. Vor dem ersten Produktionslauf prüfen, ob das Gerät korrekt erkannt wurde. Bei Bedarf zusätzlich den Programmknoten „**GRIPLINK Check Device**“ verwenden.

5.1.2.2 Werkzeugwechsel an der Tool-I/O-Schnittstelle

Für einen sicheren Werkzeugwechsel an der Tool-I/O-Schnittstelle wird folgende Reihenfolge empfohlen:

1. Das laufende Roboterprogramm beenden und den Roboter in einen sicheren Stillstand bringen.
2. In den Tool-I/O-Einstellungen des Roboters zunächst **„Ansteuerung durch: Benutzer“** auswählen.
3. Sicherstellen, dass an der Tool-I/O-Schnittstelle **0 V** anliegen, bevor Kabel oder Werkzeug gelöst werden.
4. Das Tool-I/O-Kabel trennen und das bisherige Werkzeug im spannungsfreien Zustand demontieren.
5. Das neue Werkzeug mechanisch montieren und das Tool-I/O-Kabel wieder anschließen.
6. Anschließend in den Tool-I/O-Einstellungen erneut **„Ansteuerung durch: GRIPLINK for UR“** auswählen.
7. Im GRIPLINK-Plugin die Schnittstelle **„Tool-I/O“** wählen und anschließend **„Rescan devices“** ausführen.
8. Vor dem Produktionsstart prüfen, ob das neu montierte Werkzeug korrekt erkannt wurde und die Tool-I/O-Schnittstelle betriebsbereit ist.

5.1.3 Port-Konfiguration

Unterhalb der IP-Adresse kann der Betriebsmodus des GRIPLINK-Controllers über zwei Radio-Buttons ausgewählt werden.

Single Controller

Die Robotersteuerung kommuniziert mit einem einzelnen GRIPLINK-Controller. In den Programmknoten können die Ports 0 bis 3 ausgewählt werden; sie entsprechen den vier IO-Link-Ports dieses GRIPLINK-Controllers.

Proxy Controller

Die Robotersteuerung kommuniziert mit einem GRIPLINK-Controller, der als Proxy-Master für ein Netzwerk aus mehreren GRIPLINK-Controllern fungiert.

Aus Sicht des URcaps steht damit ein GRIPLINK-Controller mit bis zu 32 Ports zur Verfügung, die in den Programmknoten über die Portnummern 0 bis 31 angesprochen werden können.

Die Proxy-Konfiguration (Rollen, Lizenzen usw.) wird über die Weboberfläche des GRIPLINK-Controllers vorgenommen.

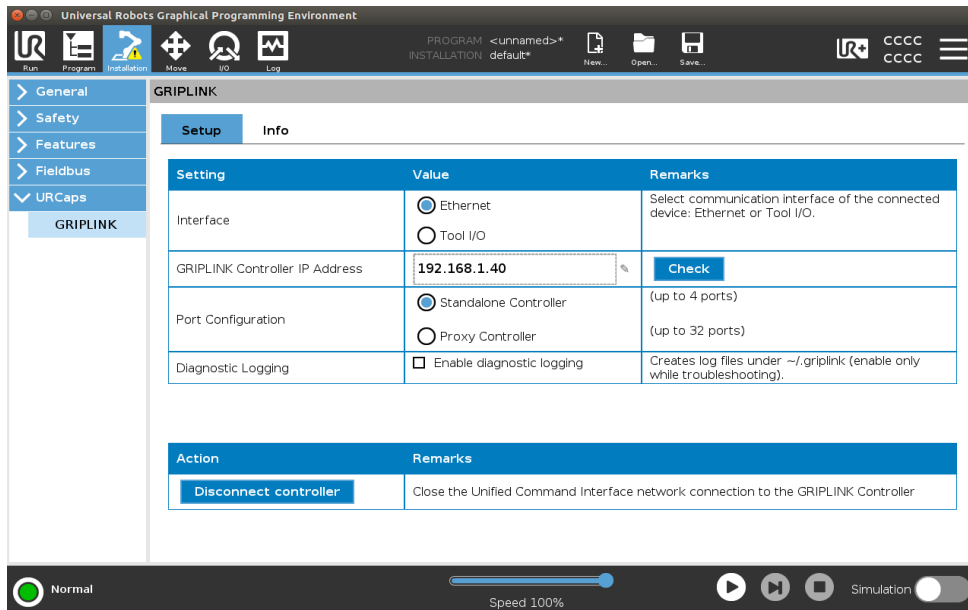


Abbildung 12: Installation/ URCaps/ GRIPLINK/Setup: Controller-Modus wählen (Single vs. Proxy).

5.1.4 Logging-Konfiguration

Mit der Checkbox „Enable Logging“ wird das Logging des GRIPLINK-Plugins ein- bzw. ausgeschaltet. Standardmäßig ist das Logging deaktiviert.

Logging aktiviert

Relevante Informationen, Warnungen und Fehlermeldungen des Plugins werden in Logdateien im Verzeichnis `~/griplink` auf der Robotersteuerung protokolliert.

Logging deaktiviert

Es werden keine zusätzlichen GRIPLINK-spezifischen Logeinträge erzeugt.

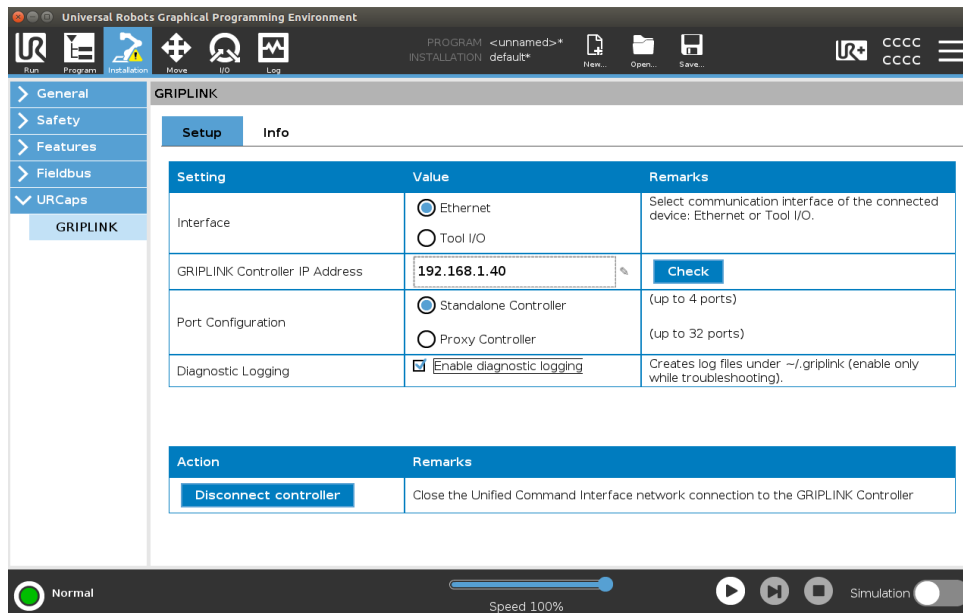


Abbildung 13: PolyScope 5 - Installation/ URCaps/ GRIPLINK/Setup: Proxy Controller gewählt & Logging aktiviert.



- Nach dem Aktivieren der Checkbox beginnt das Logging, sobald die GRIPLINK-Installationsseite einmal geöffnet wurde.
- Die Logdateien dienen der Diagnose und Fehlersuche im Zusammenhang mit dem GRIPLINK-Plugin.
- Für den regulären Betrieb wird empfohlen, das Logging zu deaktivieren, um das Logaufkommen und den Speicherbedarf zu begrenzen.

6 GRIPLINK Toolbar

Die GRIPLINK-Toolbar bietet eine komfortable Bedienoberfläche für die im aktuellen System unterstützten und erkannten angeschlossenen Geräte. Sie kann in der Ansicht von PolyScope 5 eingeblendet werden und unterstützt insbesondere folgende Aufgaben:

- Inbetriebnahme
- Diagnose
- Greifen und Freigeben beim Einrichten und Testen
- Einsehen von Presets
- Beobachten wichtiger Gerätedaten während der Programmierung

Die Toolbar zeigt die Daten an und stellt die unterstützten Funktionen für den aktuell ausgewählten Port bereit.

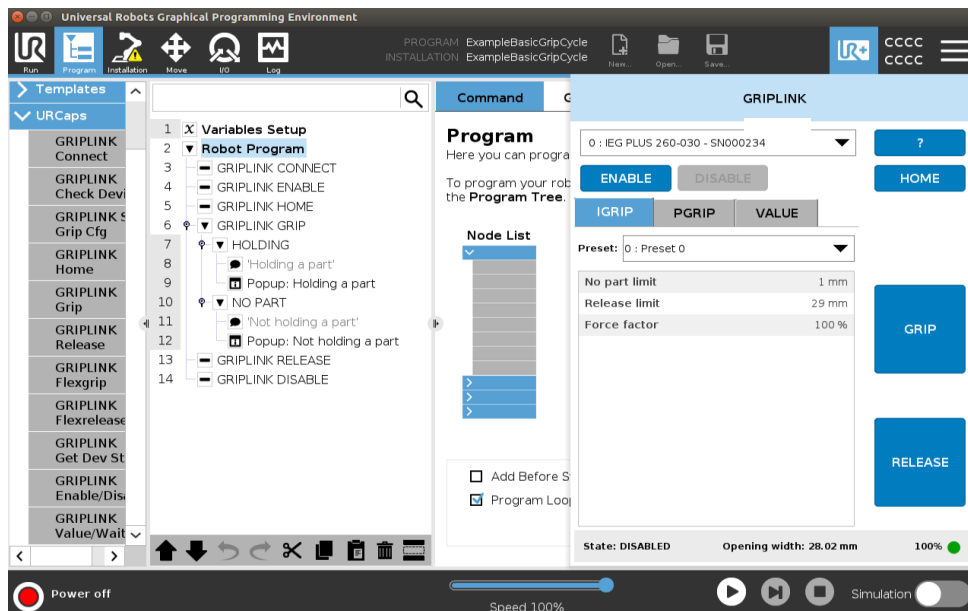


Abbildung 14: PolyScope 5 - GRIPLINK Toolbar.

6.1 Aufbau der Toolbar

Im oberen Bereich der Toolbar befinden sich die Combobox zur Geräteauswahl sowie die Buttons zur Steuerung des ausgewählten Geräts.

Geräteauswahl (Combobox)

Dropdown zur Auswahl des gewünschten Geräts z. B.

- 0 : IEG PLUS 260-030 – SN000000

Angezeigt werden Portnummer, Gerätetyp, Seriennummer oder ein optionaler Tag, der auf dem GRIPLINK-Controller konfiguriert wurde.

ENABLE / DISABLE (Button)

Aktiviert bzw. deaktiviert das ausgewählte Gerät. Der aktuelle Zustand wird in der Statuszeile angezeigt z. B.

- State: DISABLED
- State: RELEASED
- State: HOLDING
- State: INVALID

HOME (Button)

Der Button HOME führt für das ausgewählte Gerät eine Referenzfahrt aus.

Im unteren Bereich der Toolbar werden zyklisch der aktuelle Gerätezustand, die Telemetriedaten und die Statusanzeige des Geräts für das Monitoring angezeigt.

State

State zeigt den aktuellen Gerätezustand an (z. B. RELEASED, HOLDING, FAULT).

Telemetriedaten

Die Telemetriedaten enthalten gerätespezifische Messwerte, z. B. die Öffnungsweite (in mm) bei Greifmodulen.

Statusanzeige

Die Statusanzeige visualisiert den aktuellen Gerätezustand. Grün kennzeichnet einen betriebsbereiten Zustand, Rot einen Fehlerzustand.

6.2 IGRIP-Tab - Arbeiten mit Grip-Presets

Der IGRIP-Tab zeigt die konfigurierten Grip-Presets des ausgewählten Geräts.

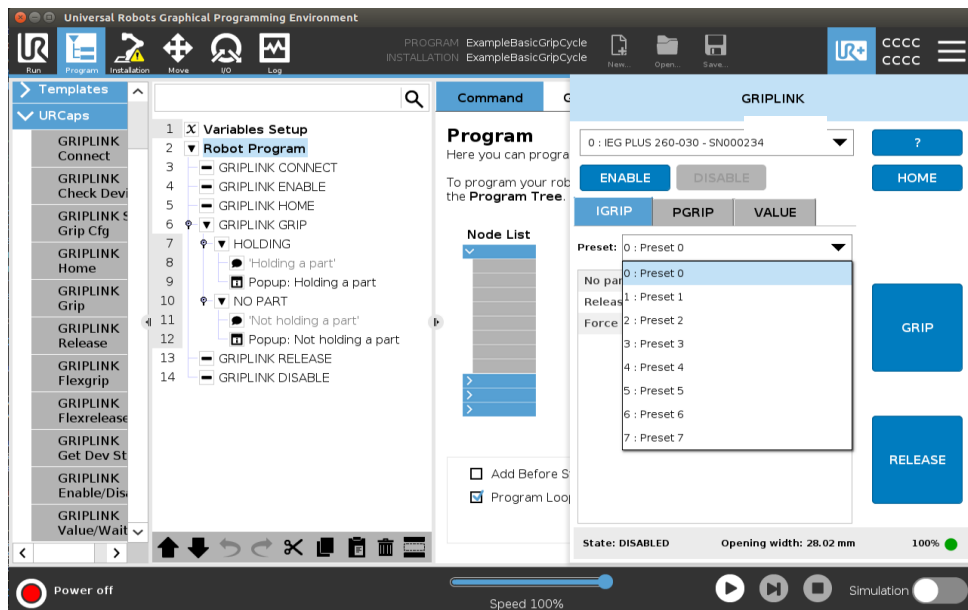


Abbildung 15: GRIPLINK Toolbar (IGRIP-Tab): Preset-Auswahl geöffnet.

Preset-Auswahl (Combobox)

Dropdown mit den verfügbaren Grip-Presets 0...7. Jede Zeile zeigt den Index und den Tag, z. B. 0 : Workpiece_1.

Preset-Parameter-Anzeige (Tabelle)

Je nach Gerät werden u. a. angezeigt:

- No part limit (mm)
- Release limit (mm)
- Force factor (%)

GRIP (Button)

Der Button GRIP führt für das ausgewählte Preset einen Greifvorgang aus.

RELEASE (Button)

Der Button RELEASE führt für das ausgewählte Preset einen Freigabevorgang aus.

Typische Verwendung

- Testen eines neu konfigurierten Presets.

- Einsehen von Presets und deren eingestellten Parametern in der Toolbar.
- Schnelles Einlernen: Preset auf dem GRIPLINK-Controller einstellen, dann über Toolbar mehrmals greifen und das Verhalten beobachten.
- Kontrolle, ob ein Bauteil mit den eingestellten Limits zuverlässig erkannt wird.

6.3 PGRIP-Tab - Flexgrip/Flexrelease

Der **PGRIP-Tab** ermöglicht ein parametrisiertes, flexibles Greifen und Freigeben (analog zu den Programmknoten „**GRIPLINK Flexgrip**“ und „**GRIPLINK Flexrelease**“).

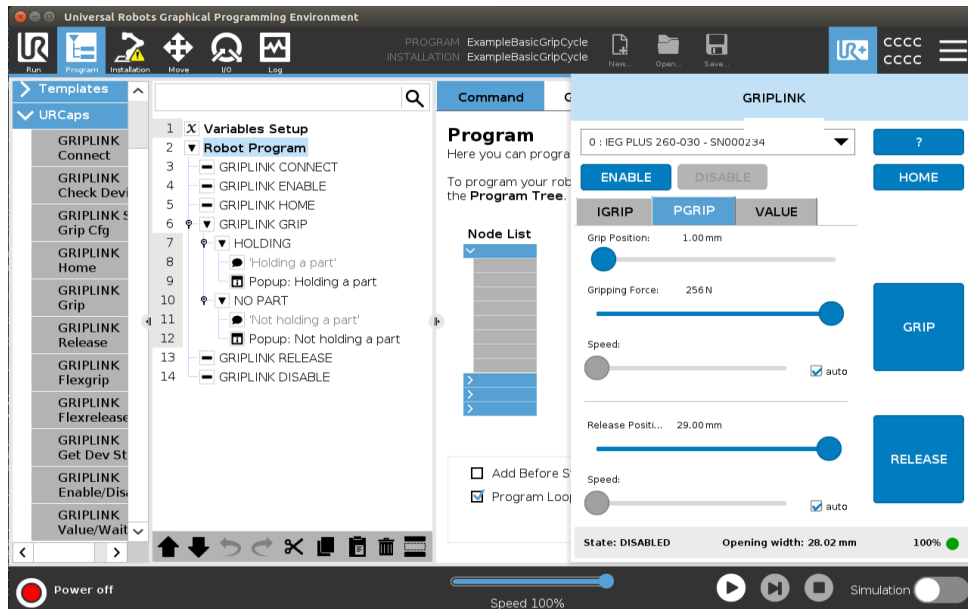


Abbildung 16: GRIPLINK Toolbar (PGRIP-Tab): Position/Force/Speed.

Verfügbare Bedienelemente

- Grip Position: Zielposition zum Greifen in mm (Schieberegler).
- Gripping Force: Greifkraft in N (Schieberegler).
- Speed: Greifgeschwindigkeit; im Ethernet-Betrieb in mm/s, im Tool-I/O-Betrieb als prozentualer Faktor (%) der maximalen Gerätegeschwindigkeit (Optional über Schieberegler).
- Release Position: Zielposition zum Freigeben in mm (Schieberegler).
- GRIP/RELEASE: starten die entsprechende Bewegung mit den eingestellten Parametern.

Einsatzfälle

- Ermitteln einer passenden Greifposition und Greifkraft für ein neues Bauteil.
- Testen, wie sanft oder schnell ein Teil gegriffen bzw. freigegeben werden kann.
- Überprüfen von Toleranzen (z. B. unterschiedliche Werkstückbreiten) vor dem Anlegen eines festen Presets.

6.4 VALUE-Tab - Gerätedaten beobachten

Der **VALUE-Tab** zeigt die aktuellen Werte des ausgewählten Geräts. Welche Indizes verfügbar sind, hängt vom Gerätetyp ab. Die Werte werden zyklisch aktualisiert.

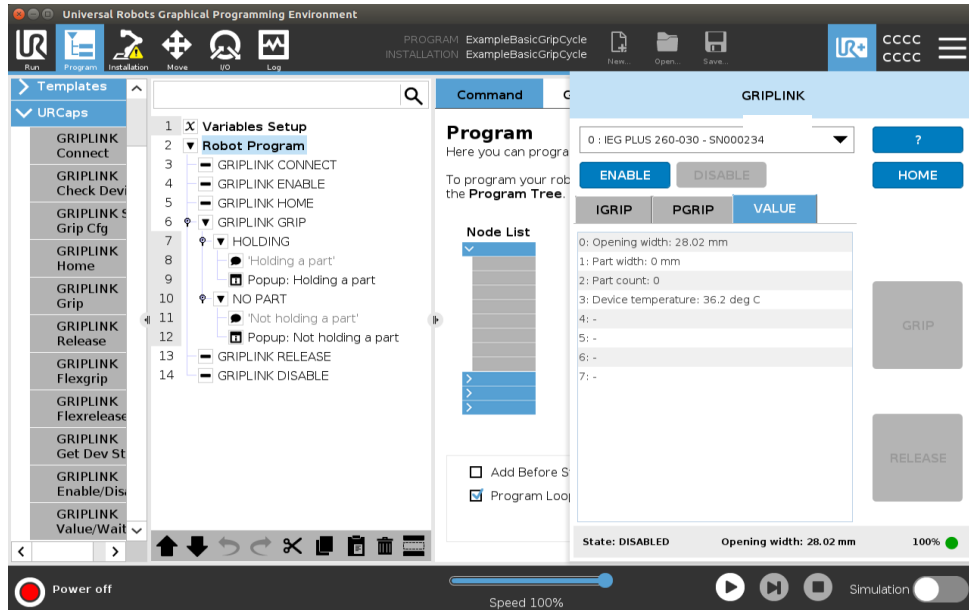


Abbildung 17: GRIPLINK Toolbar (VALUE-Tab) - Telemetriedaten/Statuswerte anzeigen.

Beispiel für die IEG PLUS-Serie

- Index 0: Öffnungsweite (mm)
- Index 1: Werkstückbreite (mm)
- Index 2: Werkstückanzahl
- Index 3: Gerätetemperatur (°C)
- Index 4–7: nicht verwendet bei IEG PLUS-Serie

Typische Verwendung

- Kontrolle, ob ein Bauteil korrekt erkannt wurde (Werkstückbreite / Werkstückanzahl).
- Überwachen der Öffnungsweite während eines Programmlaufs.
- Überprüfen der Gerätetemperatur bei langen Zyklen.
- Unterstützung bei der Wahl sinnvoller Grenzwerte (No Part-Limit, Release-Limit usw.).

6.5 Hinweise zur Verwendung der Toolbar



Während ein Roboterprogramm läuft, dürfen die Aktionen der Toolbar nicht verwendet werden. Geräte können bei aktivem Programm nicht über die Toolbar gesteuert werden.



Werden Buttons in kurzer Folge mehrfach betätigt, werden sie temporär deaktiviert und anschließend automatisch wieder freigegeben. Dies verhindert eine Überlastung von Kommunikation und Robotersteuerung.



Die Toolbar-Oberfläche ist vom aktuellen Gerätezustand und den Geräteeigenschaften abhängig. Bedienelemente werden je nach Status (z. B. DISABLED, FAULT, HOLDING) automatisch aktiviert oder deaktiviert.

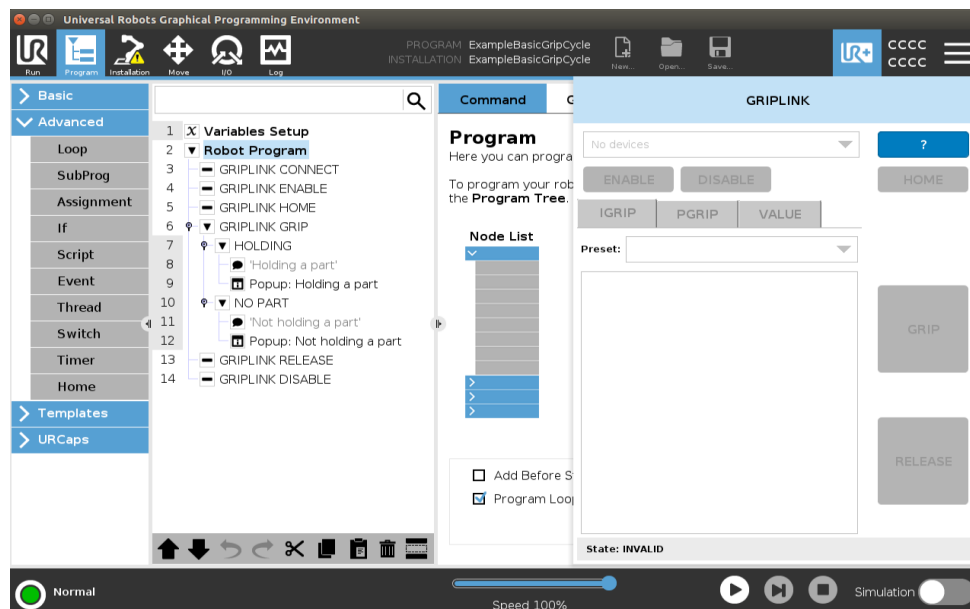


Abbildung 18: GRIPLINK-Toolbar – Kein Gerät verbunden, Bedienelemente deaktiviert.



Die Toolbar ändert bestehende Programmknoten nicht. Sie dient ausschließlich zur manuellen Bedienung, Diagnose und Inbetriebnahme. Änderungen, die dauerhaft im Ablauf wirken sollen, müssen im Roboterprogramm vorgenommen werden.

7 Programmknoten (Nodes)

7.1 Prinzipieller Programmablauf

Jedes Roboterprogramm, das Geräte wie Greifer und Sensoren über den GRIPLINK ansteuert, sollte die folgenden Design-Richtlinien befolgen. Dadurch wird eine robuste Initialisierung der Geräte und ein reproduzierbarer Ablauf im Betrieb sichergestellt.

7.1.1 Initialisierungsabschnitt (z. B. BeforeStart)

Im Initialisierungsabschnitt wird die Verbindung zum GRIPLINK-Controller aufgebaut und alle angeschlossenen Geräte vorbereitet.

Verbindung zum GRIPLINK-Controller aufbauen

Am Anfang des Roboterprogramms wird der Programmknoten GRIPLINK CONNECT eingefügt. Er stellt die Netzwerkverbindung zum in der Installation konfigurierten GRIPLINK-Controller her.

Geräte initialisieren

Für jedes am GRIPLINK-Controller angeschlossene Gerät wird einmalig ein Initialisierungsteil ausgeführt, zum Beispiel in Form eines Unterprogramms (Sub-Program).

7.1.2 Typischer Ablauf pro Gerät

GRIPLINK Check Device

Prüft am entsprechenden Port, ob das erwartete Gerät angeschlossen ist (Abgleich von Vendor-ID und Product-ID).

GRIPLINK Set Grip Config, GRIPLINK Home und GRIPLINK Enable:

- Vorkonfiguration der Grip-Presets (No-Part-Limit, Release-Limit, Kraftfaktor usw.).
- Referenzfahrt des Geräts (Homing).
- Aktivierung des Geräts für den späteren Betrieb.

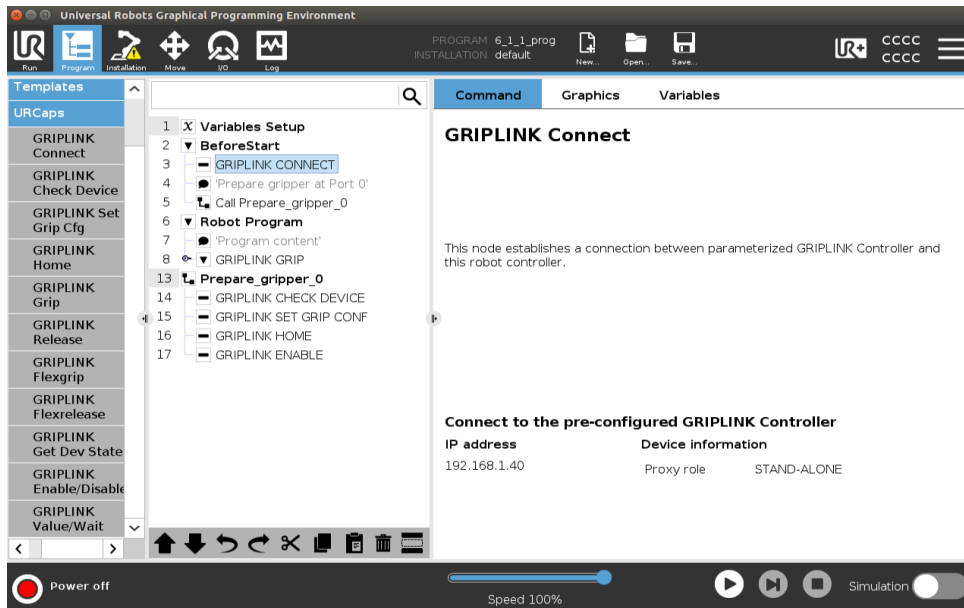


Abbildung 19: Programmknoten im Initialisierungsabschnitt (BeforeStart).

7.1.3 Hauptprogrammabschnitt

Im Hauptteil des Roboterprogramms werden die Geräte im Zyklus verwendet:

Bauteile greifen und freigegeben

- mit vordefinierten Grip-Presets über GRIPLINK Grip und GRIPLINK Release.
- oder mit parametrisierten Greifvorgängen über GRIPLINK Flexgrip/Flexrelease.

Gerätezustand und Werte auslesen, prüfen und überwachen

- Gerätezustände mit **GRIPLINK Get Dev State** prüfen (siehe Anhang A für alle Gerätezustände).
- Gerätewerte mit **GRIPLINK VALUE/WAIT** lesen und überwachen (z. B. Öffnungsweite, Werkstückbreite, Werkstückanzahl).



Die Einhaltung dieses empfohlenen Ablaufes unterstützt eine robuste Initialisierung und einen reproduzierbaren Betrieb.

In den folgenden Abschnitten werden die verfügbaren Befehle des Plugins beschrieben. Jeder Befehl ist sowohl als URScript-Code als auch, sofern vorhanden, als grafischer Programmknoten verwendbar.

7.2 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.

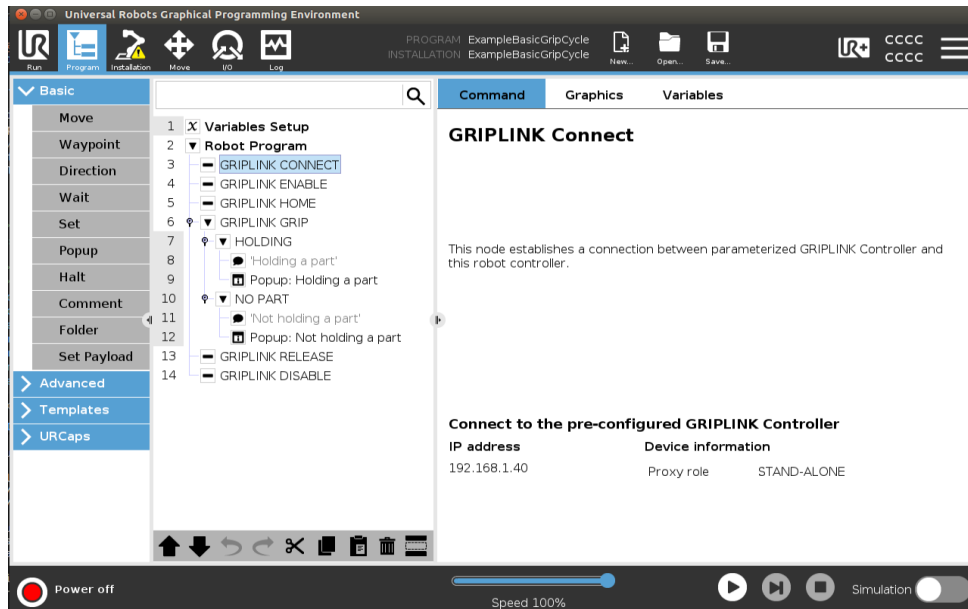


Abbildung 20: GRIPLINK CONNECT - Verbindung zum GRIPLINK-Controller.



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.



Die Tool-I/O-Schnittstelle wird über die IP-Adresse 127.0.0.1 angesprochen. Beim Aufruf der URScript-Funktion `GL_CONNECT()` ist daher die IP-Adresse 127.0.0.1 zu verwenden. Der GRIPLINK-Socket ist für beide Schnittstellen identisch und lautet „sock_griplink“.

Befehlsaufruf mit URScript-Code

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

Parameter	Typ	Bedeutung
<GRIPLINK_IP>	String	IP-Adresse des GRIPLINK-Controllers Z. B. "192.168.1.40"

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

Beispiel für die Ethernet-Schnittstelle:

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

Beispiel für die Tool-I/O-Schnittstelle:

```
GL_CONNECT("127.0.0.1", "sock_griplink")
```

7.3 Ein angeschlossenes Gerät prüfen - GRIPLINK Check Device

Bevor ein Gerät im Programm verwendet wird, empfiehlt es sich zu prüfen, ob das erwartete Gerät auch angeschlossen ist. Der Programmknoten „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 können Port, Vendor- und Product-ID eingestellt werden. Im Beispiel wird an Port 0 ein CRG 30-050 von WEISS ROBOTICS (VID 815, PID 40) erwartet.

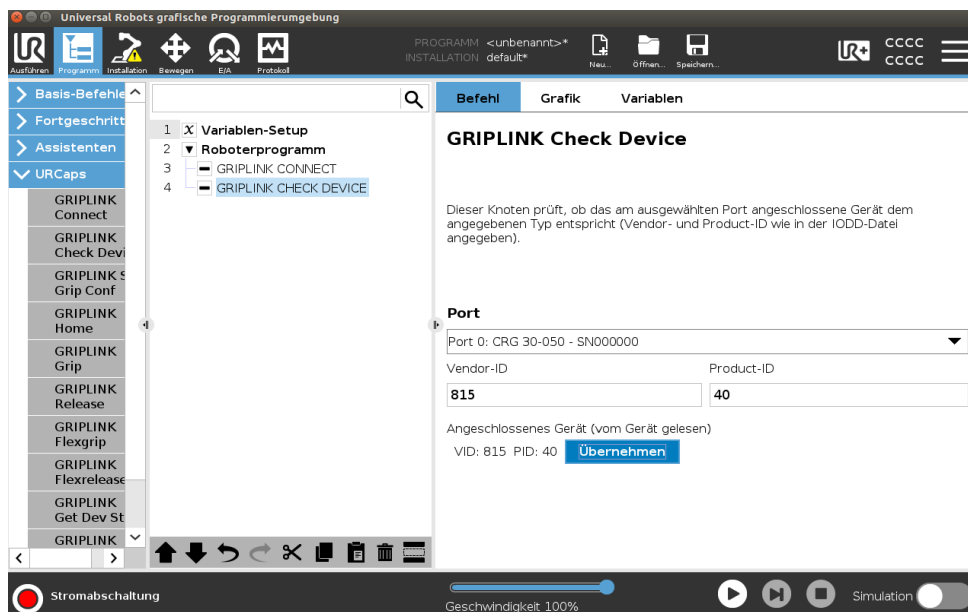


Abbildung 21: GRIPLINK CHECK DEVICE - Geräteprüfung am Port anhand Vendor ID und Product ID.

Befehlsaufruf mit URScript-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, z. B. "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(0,815,20,"sock_griplink")
```

7.4 Gerät referenzieren - GRIPLINK Home

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

Port (Combobox)

Auswahl des Ports, an dem das Gerät angeschlossen ist. Im Online-Betrieb zeigt die Auswahlliste die angeschlossenen Geräte mit

- Gerätetyp,
- Seriennummer und
- ggf. einem frei wählbaren Tag, der auf dem GRIPLINK-Controller konfiguriert werden kann.

So kann der gewünschte Greifer oder Sensor eindeutig identifiziert und direkt aus der Dropdown-Liste ausgewählt werden.

Actions Home / Enable / Disable (Button)

Über die Buttons im unteren Bereich können für den ausgewählten Port direkt folgende Aktionen ausgeführt werden:

- Home: Referenzfahrt des Geräts auslösen
- Enable: Gerät aktivieren
- Disable: Gerät deaktivieren

Diese Funktionen sind insbesondere während Inbetriebnahme und Test der Applikation hilfreich, da sich Geräte gezielt ansteuern lassen, ohne das komplette Roboterprogramm ausführen zu müssen.



Weitere Details finden sich in der Betriebsanleitung des jeweiligen Gerätes.

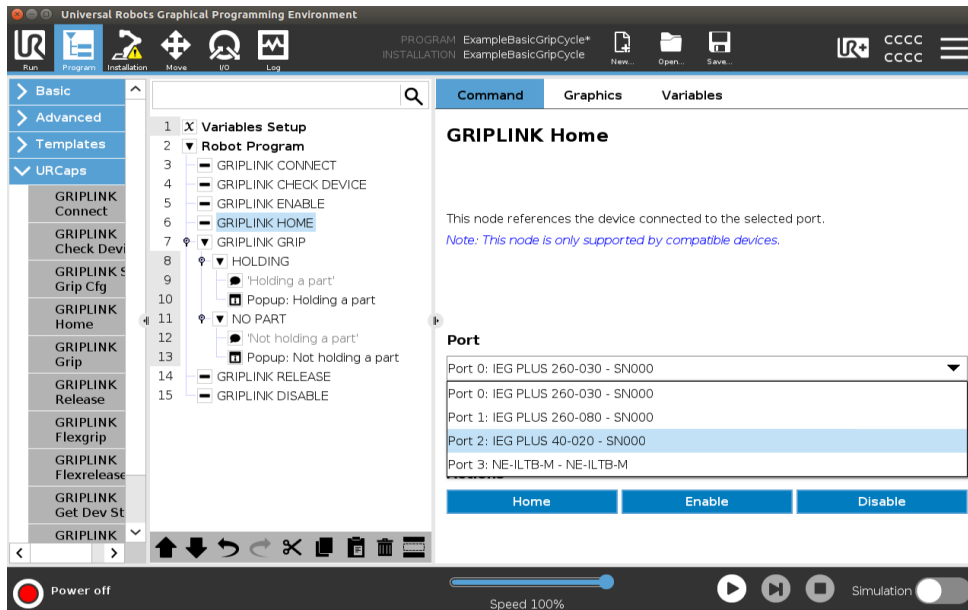


Abbildung 22: GRIPLINK Home - Gerät referenzieren (Port-Auswahl).



Ein Port muss ausgewählt sein!

Befehlsaufruf mit URScript-Code

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

Parameter	Typ	Bedeutung
<PORT>	Integer	Port-Index (0...31)
<SOCKET_NAME>	String	Name des zu verwendenden Sockets, z. B. "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")
```

7.5 Gerät aktivieren und deaktivieren - GRIPLINK Enable/Disable

Geräte können im Betrieb aktiviert und deaktiviert werden. Der Programmknoten „GRIPLINK ENABLE/DISABLE“ fasst beide Funktionen in einem Programmknoten zusammen. Es stehen folgende Einstellungen zur Verfügung.

Port (Combobox)

Auswahl genau eines Ports, dessen Gerät aktiviert bzw. deaktiviert werden soll. Es muss ein Port ausgewählt sein, damit der Befehl ausgeführt werden kann.

Aktionsauswahl (Combobox)

Auswahl der auszuführenden Aktion:

- **Enable:** aktiviert das Gerät am ausgewählten Port
- **Disable:** deaktiviert das Gerät am ausgewählten Port

Die getroffene Auswahl wird im Programm gespeichert. Der Programmknoten passt seinen Titel entsprechend der Benutzerauswahl an, ob der Knoten als **Enable** oder als **Disable** ausgeführt wird.

Wait for state transitions (Checkbox)

Mit dieser Option kann gesteuert werden, ob nach Ausführung des Befehls geprüft werden soll, ob sich der Gerätezustand am ausgewählten Port tatsächlich ändert.

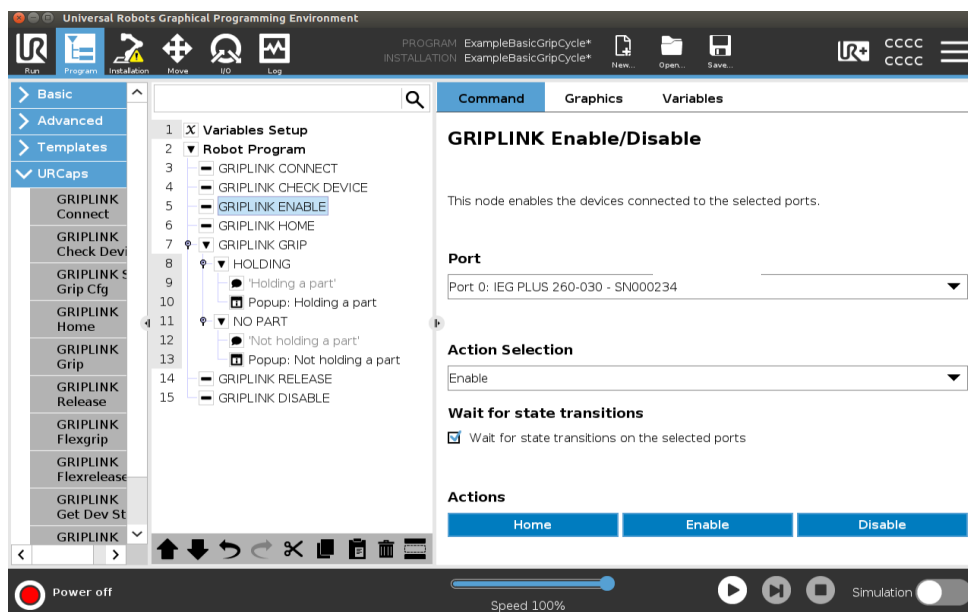


Abbildung 23: GRIPLINK Enable/Disable - Aktion Enable (mit Wait for state transitions).

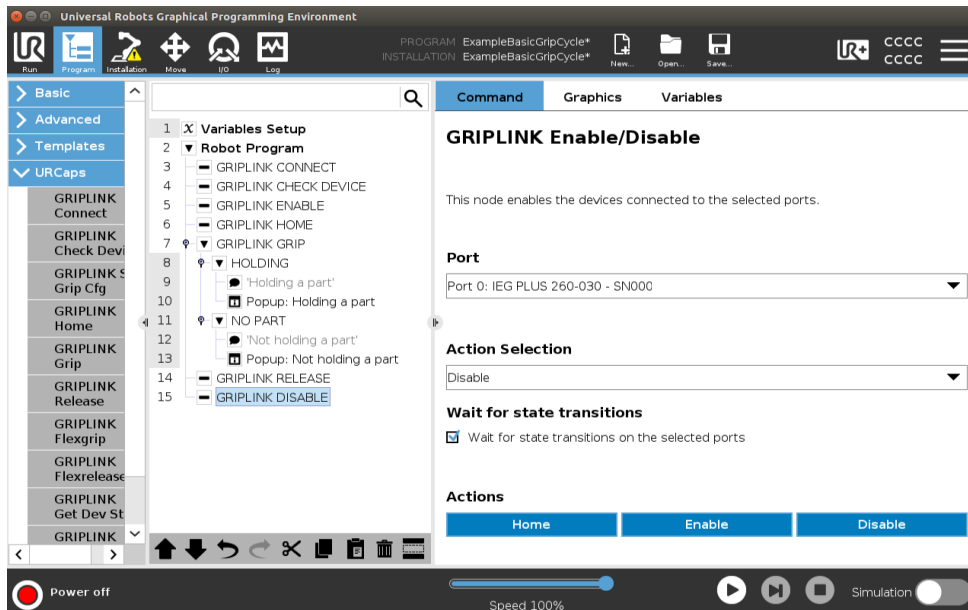


Abbildung 24: GRIPLINK Enable/Disable - Aktion Disable (mit Wait for state transitions).



Einige Geräte müssen vor der Verwendung zunächst durch den Enable-Befehl aktiviert werden! Weitere Details finden sich in der Betriebsanleitung des jeweiligen Gerätetreibers.



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



Befindet sich ein Gerät bereits im Zustand DISABLED, wenn der Programmknoten „GRIPLINK Disable“ mit der Checkbox „Wait for state transitions“ aufgerufen wird, kann dies zu einem Timeout-Fehler führen.

Befehlsaufruf mit URScript-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 Geräts zu warten „False“, um nicht auf Zustandsübergang des Geräts zu warten
<SOCKET_NAME>	String	Name des zu verwendenden Sockets, z. B. "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")
```

7.6 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 Grip-Presets ausgeführt werden.



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

In den Programmknoten „GRIPLINK Grip“ und „GRIPLINK Release“ können

- der Port des gewünschten Greifers und
- der Index des Grip-Presets

ausgewählt werden. Beim Release-Knoten kann zusätzlich das Blockier-Verhalten (blockierend / nicht blockierend) eingestellt werden.

Blockier-Verhalten

- **GRIPLINK Grip:** Der grafische Programmknoten „GRIPLINK Grip“ ist standardmäßig blockierend ausgelegt. Das Roboterprogramm bleibt an diesem Programmknoten, bis der Greifvorgang abgeschlossen ist.
- **GRIPLINK Release:** Beim Knoten „GRIPLINK Release“ kann in der Checkbox gewählt werden, ob der Befehl blockierend oder nicht blockierend ausgeführt wird.

Bei **blockierender Ausführung** wird der Befehl an den GRIPLINK-Controller gesendet und erst dann bestätigt, wenn der Greif- oder Freigabevorgang abgeschlossen ist.

Bei **nicht blockierender Ausführung** wird der Greif- oder Freigabevorgang gestartet und der Knoten sofort abgeschlossen; das Roboterprogramm läuft weiter, während der Greifer noch greift bzw. freigibt.

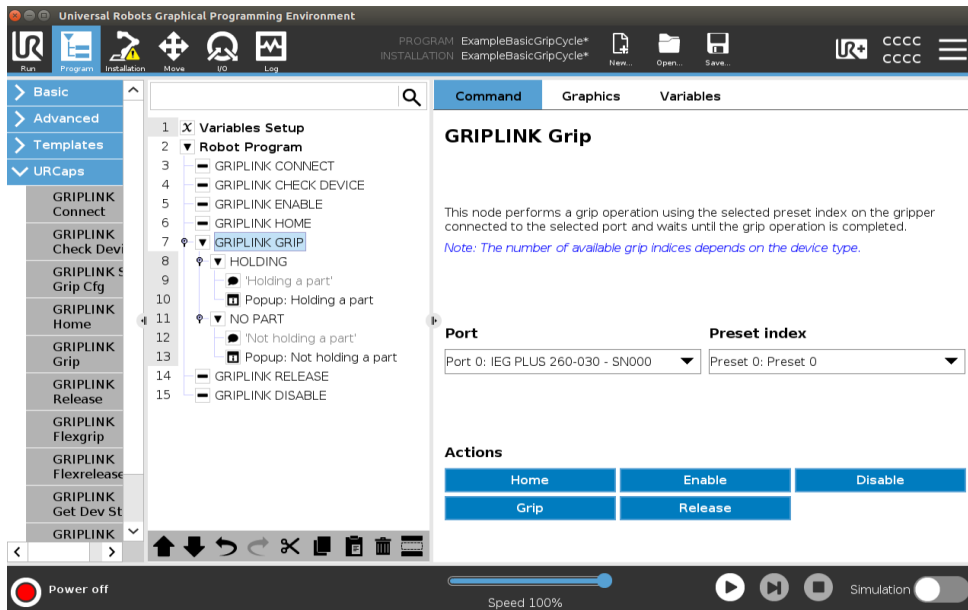


Abbildung 25: GRIPLINK GRIP - Greifvorgang am ausgewählten Port mit Preset-Index ausführen (blockierend).

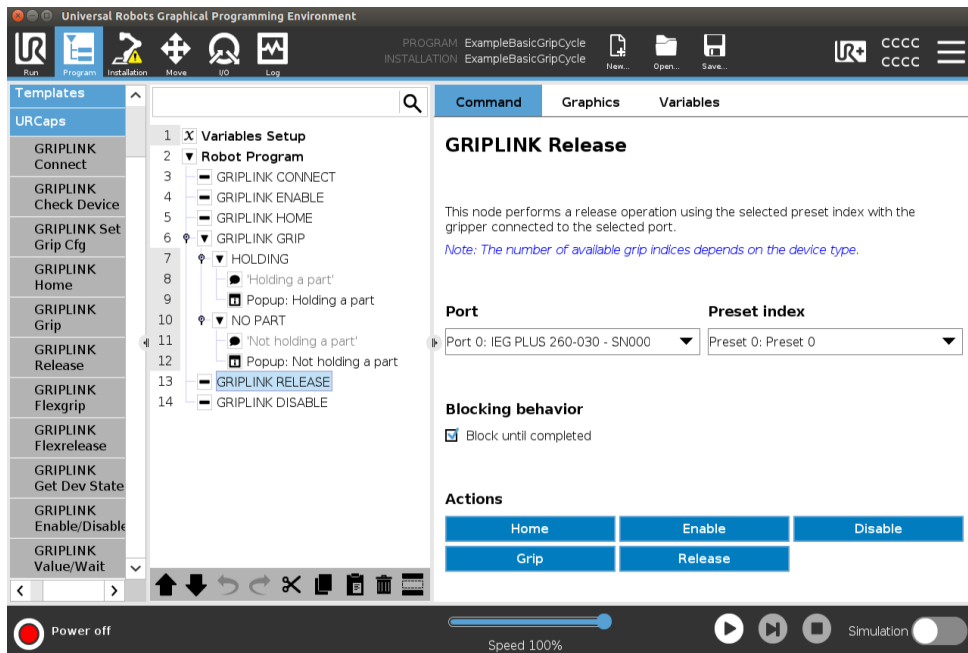


Abbildung 26: GRIPLINK RELEASE - Freigabevorgang am ausgewählten Port mit Preset-Index (optional blockierend).



Der grafische Programmknoten „GRIPLINK Grip“ ist immer blockierend ausgeführt. Eine nicht blockierende Ausführung des Greifbefehls ist nur über die URScript-Funktion `GL_GRIP(..., <DO_BLOCK> = 0)` möglich.



Achten Sie darauf, dass das Bewegungs- und Sicherheitskonzept der Anlage zum gewählten Blockier-Verhalten passt. In vielen Anwendungen ist eine blockierende Ausführung sinnvoll, damit der Roboter erst dann weiterfährt, wenn der Greifvorgang oder Freigabevorgang zuverlässig abgeschlossen ist.



Befindet sich ein Gerät bereits im Zustand „**HOLDING**“ oder „**NO_PART**“, ist es im GRIPLINK-Protokoll nicht zulässig, den Befehl erneut mit dem Programmknoten „GRIPLINK Grip“ bzw. im URScript-Code mit `GL_GRIP()` auszuführen.



Bei GRIPKIT EASY-Greifmodulen mit einer Firmware-Version unter 3.0.0 wird nur die blockierende Ausführung unterstützt. Nicht blockierende Aufrufe sind für diese Geräte nicht zulässig.



Wird der Programmknoten „GRIPLINK Grip“ in einem dieser Zustände (S_HOLDING oder S_NO_PART) dennoch aufgerufen, führt dies zu einer Meldung: „Service unavailable within current state“.

Befehlsaufruf mit URScript-Code

```
GL_GRIP (
    <PORT>,
    <PRESET_INDEX>,
    <DO_BLOCK>,
    <SOCKET_NAME>
)
GL_RELEASE (
    <PORT>,
    <PRESET_INDEX>,
    <DO_BLOCK>,
    <SOCKET_NAME>
)
```

Parameter	Typ	Bedeutung
<PORT>	Integer	Port-Index (0...31)
<PRESET_INDEX>	Integer	Preset-Index (0...7)
<DO_BLOCK>	Integer	0: nicht blockierende Ausführung 1: blockierende Ausführung
<SOCKET_NAME>	String	Name des zu verwendenden Sockets, z. B. "sock_griplink" Der Socket-Name wird in allen nachfolgenden Knoten im Roboterprogramm verwendet!

Beispiel 1 - Blockierende Ausführung mit GRIPLINK Grip/Release:

```
# Grip with gripper connected to port 0 and preset index 3
# Execute the command with blocking
GL_GRIP(0,3,1,"sock_griplink")

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

# Release part again
GL_RELEASE(0,3,1,"sock_griplink")
```

Beispiel 2 - Nicht blockierendes Greifen, WSTR und Get Dev State:

```
# Grip with gripper connected to port 0 and preset index 3
# Execute the command without blocking
GL_GRIP(0,3,0,"sock_griplink")

# ...
# Wait for state transition on port 0
GL_WSTR(0,"sock_griplink")

# Get Device State to evaluate
# See Attachment A for the device States
gripper0_state = GL_DEVSTATE(0,"sock_griplink")

if (gripper0_state == S_HOLDING):
    # Release part again (blocking)
    GL_RELEASE(0,3,1,"sock_griplink")
    GL_DISABLE(0,True,"sock_griplink")
else:
    # Process other states here (e.g. S_NO_PART, S_FAULT,...)
    GL_DISABLE(0,True,"sock_griplink")
end
```

7.6.1 Auswertung des Greifzustands

Der Programmknoten „GRIPLINK Grip“ erkennt nach dem Greifen, ob der Greifer ein Bauteil gegriffen hat oder nicht.

Damit lassen sich im Programm Abläufe abhängig vom Greifzustand einfach abbilden, z. B. alternative Abläufe für den Fall, dass kein Werkstück vorhanden ist (NO_PART), sowie reguläre Bearbeitungsschritte bei erfolgreichem Griff (HOLDING).

Je nach Ergebnis werden die untergeordneten Knoten (Child-Nodes) wie folgt ausgeführt:

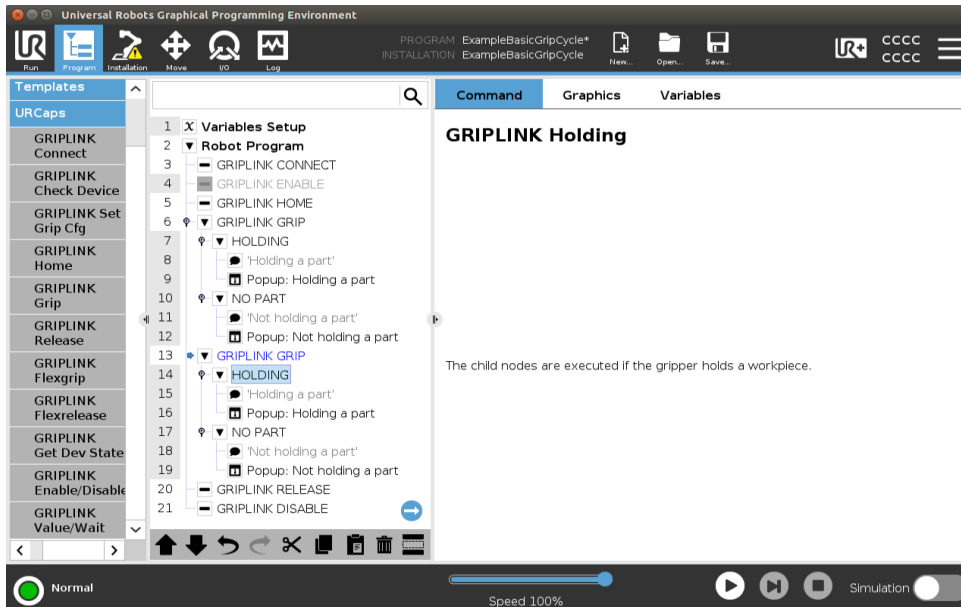


Abbildung 27: Child-Node „HOLDING“ - Ausführen der untergeordneten Aktionen, wenn das Gerät ein Werkstück hält.

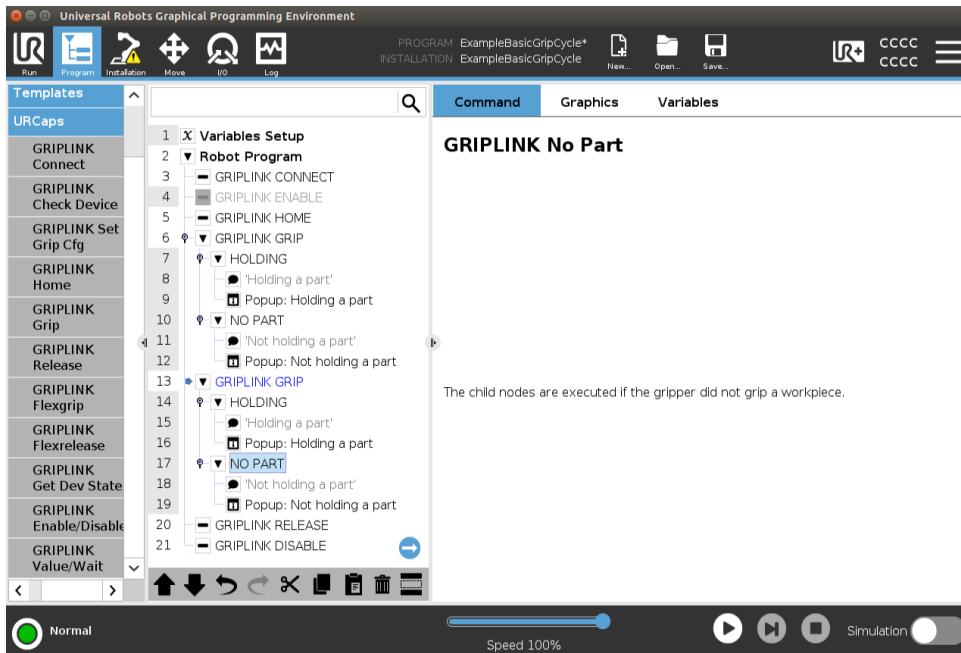


Abbildung 28: Child-Node „NO PART“ - Ausführen der untergeordneten Aktionen, wenn kein Werkstück gegriffen wurde.

7.7 Flexibles Greifen, Freigeben und Vorpositionieren - GRIPLINK Flexgrip/Flexrelease

Die Greifmodule der WPG-Serie, INTRAPAL-Serie, IEG PLUS-Serie, CRG-Serie und STERIGRIP-Serie unterstützen flexibles Greifen und Freigeben. Hier können Position, Greifkraft und Bewegungsparameter unabhängig von Presets vorgegeben werden. Dadurch sind 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 Programmknoten „GRIPLINK Flexgrip“ können Sie:

- Port des gewünschten Greifers auswählen
- Zielposition in mm vorgeben
- Gewünschte Greifkraft in N einstellen
- Geschwindigkeit im Ethernet-Betrieb in mm/s oder im Tool-I/O-Betrieb in % vorgeben
- Beschleunigung im Ethernet-Betrieb in mm/s² oder im Tool-I/O-Betrieb in % vorgeben

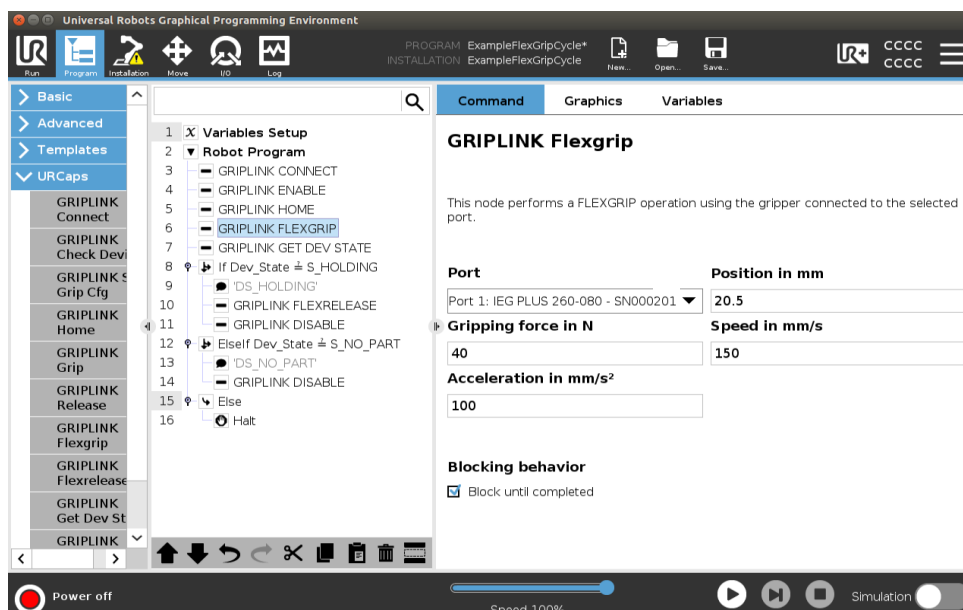


Abbildung 29: GRIPLINK FLEXGRIP - Flexibles Greifen mit Parametern (Position, Greifkraft, Geschwindigkeit, Beschleunigung; optional blockierend).

Im Programmknoten „GRIPLINK Flexrelease“ können Sie:

- Port des gewünschten Greifers auswählen
- Zielposition in mm vorgeben
- Geschwindigkeit im Ethernet-Betrieb in mm/s oder im Tool-I/O-Betrieb in % vorgeben
- Beschleunigung im Ethernet-Betrieb in mm/s² oder im Tool-I/O-Betrieb in % vorgeben

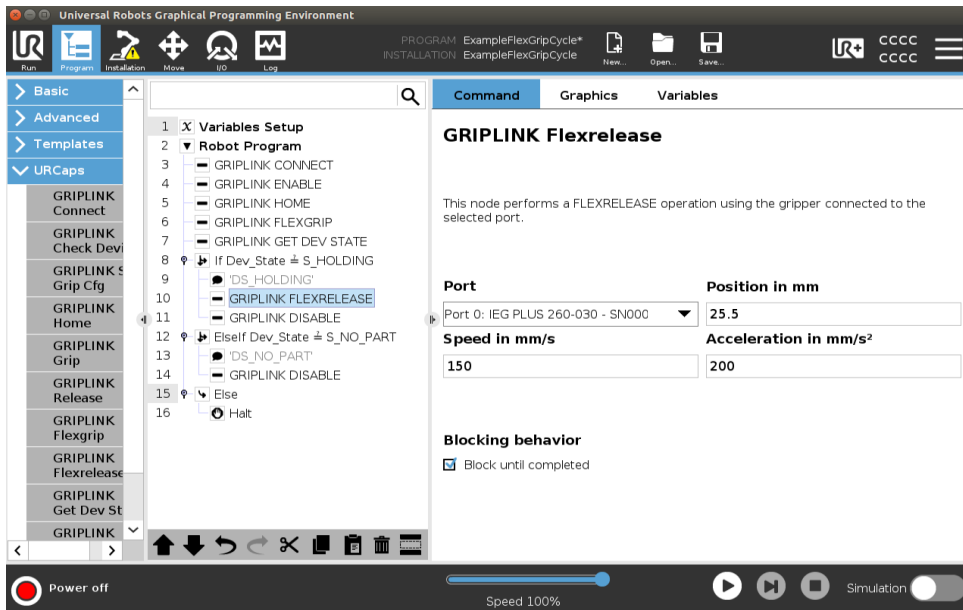


Abbildung 30: GRIPLINK FLEXRELEASE - Flexibles Freigeben mit Parametern (Position, Geschwindigkeit, Beschleunigung; optional blockierend).

In beiden Knoten steht außerdem die Checkbox „Blocking behavior“ zur Verfügung:

Blockierend (Checkbox mit Haken : `do_block = 1`)

- Der Befehl wird an den GRIPLINK-Controller gesendet und erst bestätigt, wenn die Bewegung abgeschlossen ist.
- Aus Sicht des Roboterprogramms bleibt der Knoten blockierend, bis der Vorgang beendet ist.
- Bei **FLEXRELEASE** wird intern zusätzlich geprüft, ob die Zielposition erreicht wurde (über `GL_WAITVAL`).

Nicht blockierend (Checkbox ohne Haken : `do_block = 0`)

- Die Bewegung wird gestartet und sofort bestätigt.
- Das Roboterprogramm läuft weiter, während die Greif- bzw. Freigabebewegung im Hintergrund ausgeführt wird.
- Eine nicht blockierende Ausführung kann z. B. verwendet werden, um mit mehreren Greifern parallele Greifprozesse durchzuführen.



Setzen Sie die Checkbox „Block until completed (Blocking behavior)“ nur dann auf nicht blockierend, wenn das Sicherheits- und Bewegungskonzept der Anlage dies zulässt.



Bei GRIPKIT EASY-Greifmodulen mit einer Firmware-Version unter 3.0.0 wird nur die blockierende Ausführung unterstützt. Nicht blockierende Aufrufe sind für diese Geräte nicht zulässig.



Befindet sich ein Gerät bereits im Zustand „**HOLDING**“ oder „**NO_PART**“, ist es im GRIPLINK-Protokoll nicht zulässig, den Befehl erneut mit dem Programmknoten „GRIPLINK Flexgrip“ bzw. im URScript-Code mit `GL_FLEXGRIP()` auszuführen.



Wird der Programmknoten „GRIPLINK Flexgrip“ in einem dieser Zustände (S_HOLDING oder S_NO_PART) dennoch aufgerufen, führt dies zu einer Meldung: „Service unavailable within current state“.

Befehlsaufruf mit URScript-Code

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

Parameter	Typ	Bedeutung
<PORT>	Integer	Port-Index (0...31)
<POSITION>	Double	Zielposition in mm
<GRIPPING_FORCE>	Double	Greifkraft in N (nur für GL_FLEXGRIP)
<SPEED>	Double	Ethernet-Schnittstelle: Geschwindigkeit in mm/s Tool-I/O-Schnittstelle: Geschwindigkeit in %
<ACCELERATION>	Double	Ethernet-Schnittstelle: Beschleunigung in mm/s ² Tool-I/O-Schnittstelle: Beschleunigung in %
<DO_BLOCK>	Integer	0: nicht blockierende Ausführung 1: blockierende Ausführung
<SOCKET_NAME>	String	Name des zu verwendenden Sockets, z. B. "sock_griplink" Der Socket-Name wird in allen nachfolgenden Knoten im Roboterprogramm verwendet!

Beispiel:

```
# Prepositioning with WPG at port 0 to 50.0 mm with optimum speed
# and acceleration
# with blocking until flexrelease ends
GL_FLEXRELEASE(0,50,0,0,1,"sock_griplink")

# Grip part with No Part Limit 45.0 mm with 100 N and optimum speed
# and acceleration
# with blocking until flexgrip ends
GL_FLEXGRIP(0,45,100,0,0,1,"sock_griplink")
```

7.8 Zustandsabfrage - GRIPLINK Get Dev State

Der Programmknoten „GRIPLINK Get Dev State“ wird verwendet, um den aktuellen Gerätezustand eines Geräts abzufragen. Der ausgelesene Gerätezustand wird als Ganzzahl (Integer) in eine vom Benutzer ausgewählte Ergebnisvariable geschrieben. Im Knoten können folgende Parameter eingestellt werden:

Port (Combobox)

Auswahl des Ports, an dem sich das Gerät befindet, dessen Zustand abgefragt werden soll.

Ergebnisvariable (Combobox)

Auswahl einer numerischen Variable, in die der Gerätezustand geschrieben wird. So können Zustände zentral definiert und wiederverwendet werden. Zur Auswahl stehen alle numerischen Variablen, die im Reiter Installation → General → Variables erstellt wurden.

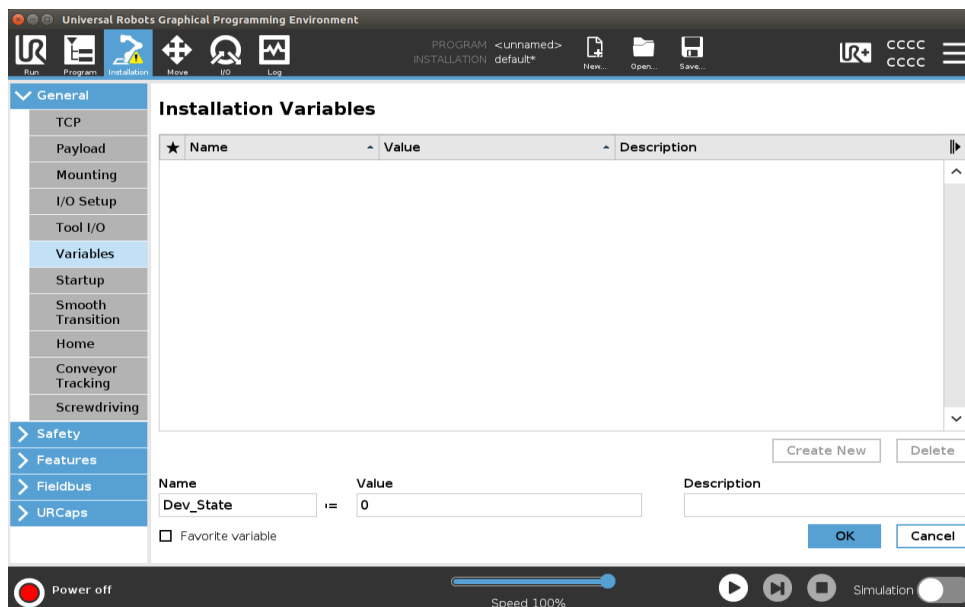


Abbildung 31: PolyScope 5 - Installation/ General/Variables: Installationsvariable anlegen.

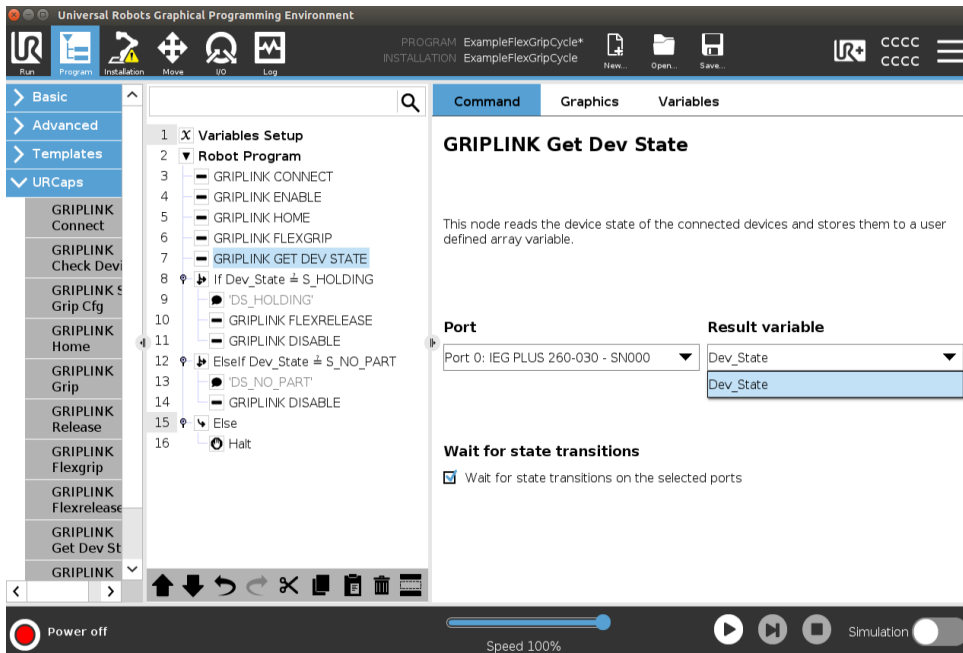


Abbildung 32: GRIPLINK Get Dev State - Gerätezustand am ausgewählten Port auslesen und in der Variable Dev_State speichern (optional mit „Wait for state transitions“).

Der Knoten verfügt über die Checkbox „Wait for state transitions on the selected ports“:

Checkbox ohne Haken

- Der aktuelle Gerätezustand wird sofort mit `GL_DEVSTATE (. . .)` gelesen.
- Das Ergebnis wird unmittelbar in die Ergebnisvariable geschrieben.
- Typischer Einsatz: einfaches periodisches Status-Polling im laufenden Programm.

Checkbox mit Haken

- Der Knoten wartet zunächst intern mit `GL_WSTR (. . .)` auf einen Zustandsübergang des Geräts am ausgewählten Port.
- Erst nach einem erkannten Zustandswechsel wird der aktuelle Gerätezustand mit `GL_DEVSTATE (. . .)` gelesen und in die Ergebnisvariable geschrieben.
- Typischer Einsatz: nach einem vorherigen Befehl (z. B. Grip, Release, Enable, Disable), wenn ausgewertet werden soll, welcher Zustand tatsächlich erreicht wurde, sobald sich das Gerät sichtbar verändert hat.



Die möglichen Gerätezustände sind in Anhang A aufgeführt!



Erfolgt über einen längeren Zeitraum kein Zustandswechsel, kann es bei aktivierter Option „Wait for state transitions on the selected ports“ zu einem Timeout kommen.

Der ausgelesene Zustand kann anschließend mit `If-Else`-Konstrukten verarbeitet werden. Im folgenden Beispiel wird der Zustand eines Greifers an Port 0 ausgewertet. Je nach Zustand wird entweder zur Ablageposition verfahren oder ein alternativer Ablauf ausgeführt.

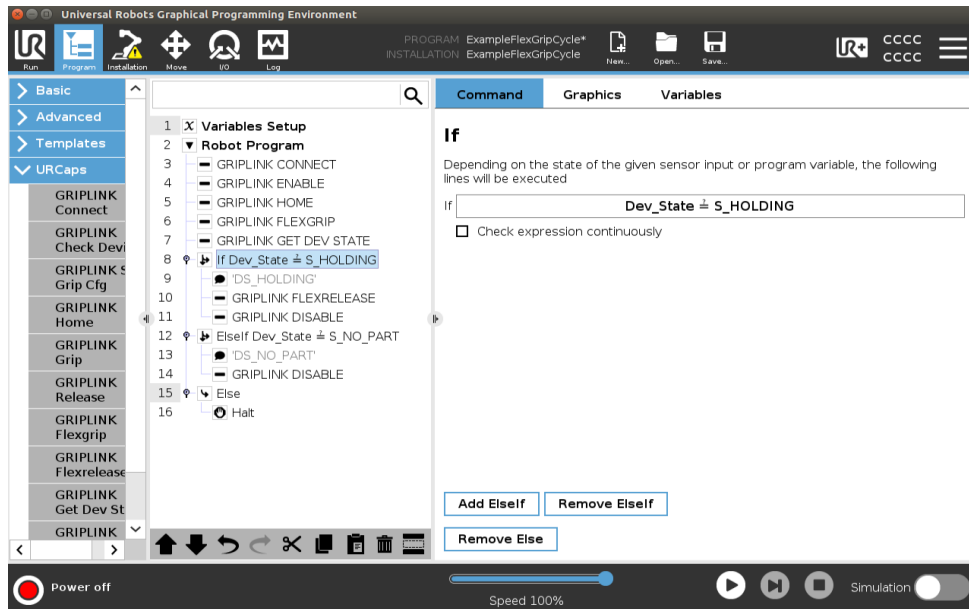


Abbildung 33: If-Bedingung - Auswertung von `Dev_State` (z. B. `Dev_State == S_HOLDING`) zur Ablaufverzweigung.

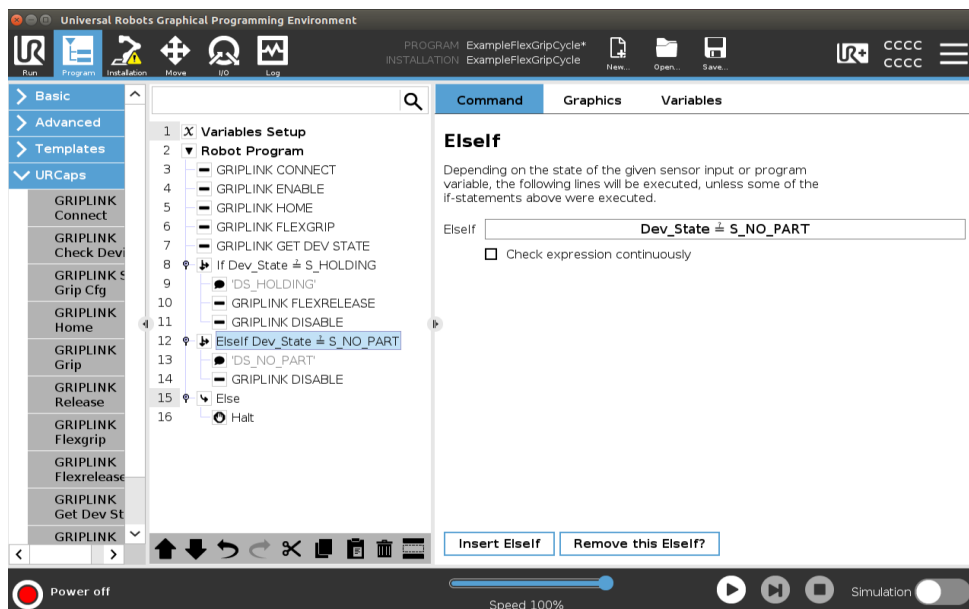


Abbildung 34: Elseif-Bedingung - Alternative Zustandsauswertung von `Dev_State` (z. B. `Dev_State == S_NO_PART`).

Befehlsaufruf mit URScript-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, z. B. "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 1 - Zustand direkt anfragen (ohne WSTR):

```
# Grip with gripper connected to port 0 and preset index 3  
# Execute the command WITH blocking  
GL_GRIP(0,3,1,"sock_griplink")  
  
# Read the current device state immediately  
gripper0_state = GL_DEVSTATE(0,"sock_griplink")  
  
# When state is HOLDING, move to place position  
if (gripper0_state == S_HOLDING):  
    # Part was successfully gripped  
    # Move to place position  
    # ...  
else:  
    # Handle other states here (e.g. S_NO_PART, S_FAULT, ...)  
end
```

Beispiel 2 - Zustand nach Zustandsübergang auswerten (mit WSTR):

```
# Grip with gripper connected to port 0 and preset index 3
# Execute the command WITHOUT blocking
GL_GRIP(0,3,0,"sock_griplink")

# Wait for a state transition on port 0
GL_WSTR(0,"sock_griplink")

# Read the current device state after the transition
gripper0_state = GL_DEVSTATE(0,"sock_griplink")

# When state is HOLDING, move to place position
if (gripper0_state == S_HOLDING):
    # Part was successfully gripped
    # Move to place position
    # ...
else:
    # Handle other states here (e.g. S_NO_PART, S_FAULT, ...)
end
```

7.9 Gerätewerte auslesen und warten - GRIPLINK Value/Wait

Der Programmknoten „GRIPLINK Value/Wait“ dient dazu, einen Gerätewert auszulesen und optional zu warten, bis dieser einen vorgegebenen Zielwert erreicht hat.

Die Arbeitsweise des Programmknotens wird über die Checkbox „**Wait device value to reach the given value within the given window**“ gesteuert:

- Checkbox ohne Haken : Gerätewert lesen
- Checkbox mit Haken : auf Gerätewert im Fenster warten, dann Gerätewert lesen

Geräte wie Sensoren liefern Daten in Form indizierter Werte. Diese Werte können ausgelesen werden; je nach Gerät stehen unterschiedlich viele Indizes zur Verfügung.

7.9.1 Gerätewert lesen

Im Programmknoten können folgende Parameter eingestellt werden:

Port (Combobox):

Auswahl des Ports, an dem der gewünschte Greifer oder Sensor angeschlossen ist.

Value index (Combobox):

Auswahl, welcher Gerätewert gelesen werden soll. Die verfügbaren Indizes hängen vom Gerätetyp ab.

Beispiel (IEG PLUS Greifer):

- Index 0: Öffnungsweite (mm)
- Index 1: Werkstückbreite (mm)
- Index 2: Werkstückanzahl
- Index 3: Gerätetemperatur (°C)

Ergebnisvariable:

Name der numerischen Variable, in die der gelesene Gerätewert geschrieben wird. Als Ergebnisvariable können alle numerischen Variablen ausgewählt werden, die im Reiter **Installation → General → Variables** definiert wurden.



Bei Greifmodulen entspricht der Wert mit Index 0 der Öffnungsweite in Millimetern.

Verhalten

- Der aktuelle Gerätewert wird einmalig gelesen (über `GL_VALUE(...)`).
- Das Ergebnis wird in die angegebene Ergebnisvariable geschrieben.
- Das Programm läuft ohne Wartezeit weiter.

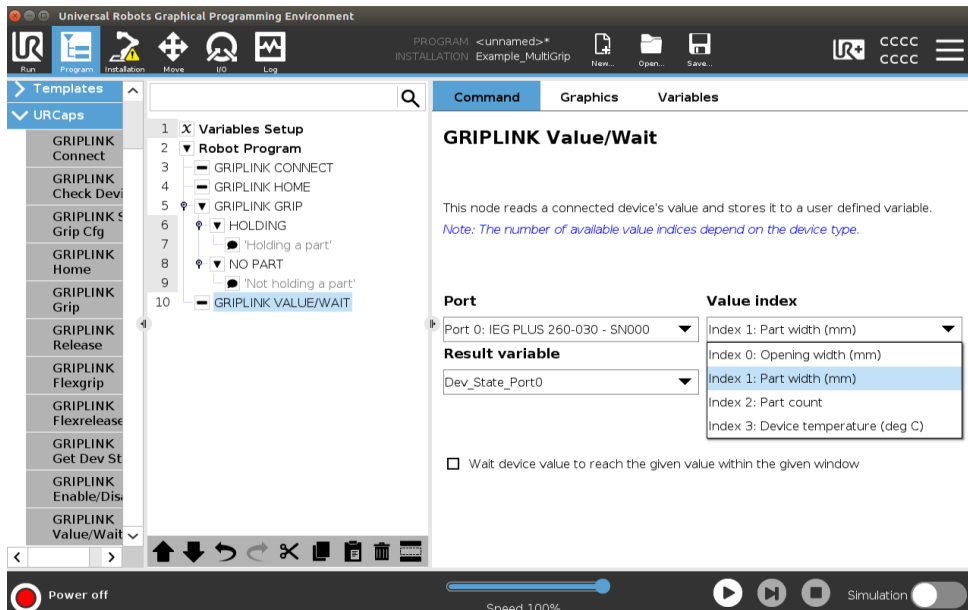


Abbildung 35: GRIPLINK VALUE/WAIT - Gerätewert (Value Index) auslesen und in eine Variable schreiben (z.B. Dev_State_Port0).



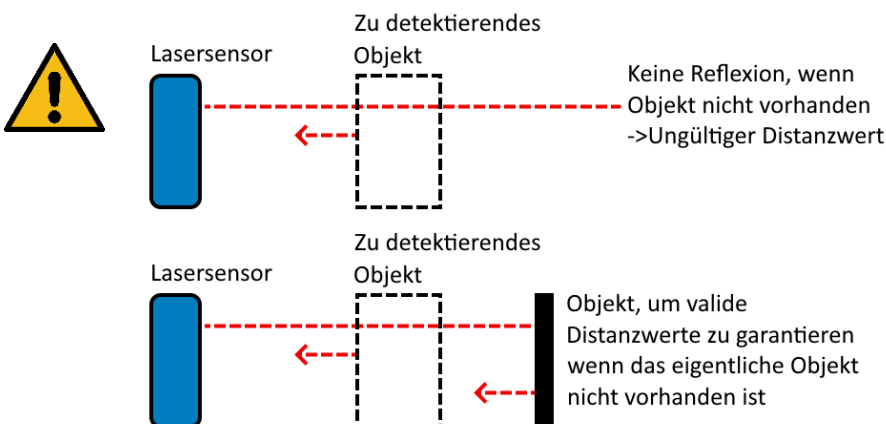
Es sind maximal acht auslesbare Werte möglich (Index 0...7), je nach Gerät können es allerdings auch weniger sein.

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 vermeiden, 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:



Befehlsaufruf mit URScript-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, z. B. "sock_griplink" Der Socket-Name wird in allen nachfolgenden Knoten im Roboterprogramm verwendet!

Typ Rückgabewert	Bedeutung
Integer	Wert des ausgelesenen Gerätewertes

Beispiel:

```

# Read value 0 (position) of a gripper connected to port 0
gripper0_position = GL_VALUE(0,0,"sock_griplink")

```

7.9.2 Gerätewert warten und lesen

Nach dem Setzen eines Gerätewertes kann es erforderlich sein, zu warten, bis dieser Wert tatsächlich erreicht wurde. Bei Aktoren kann das beispielsweise eine Zielposition sein.

Im **Wartemodus** (Checkbox mit Haken) sorgt der Knoten dafür, dass

- auf einen bestimmten **Schwellwert** gewartet wird und
- dieser innerhalb eines **Toleranzfensters** und
- innerhalb einer vorgegebenen Zeit in ms (**Timeout**) erreicht werden muss.

Wird der Zielbereich nicht innerhalb der angegebenen Zeit erreicht, wird ein **Timeout-Fehler** erzeugt. Im Wartemodus stehen zusätzlich folgende Einstellmöglichkeiten zur Verfügung:

Value threshold

Schwellwert, auf den der Gerätewert zulaufen soll (z. B. 20 mm Werkstückbreite).

Window size

Größe des Toleranzfensters um den Zielwert. Beispiel:

- Window size = 1,5 mm
- Gültiger Bereich: 20 mm ± 0,75 mm

Timeout in ms

Maximale Wartezeit in Millisekunden. Wird der Zielbereich innerhalb dieser Zeit nicht erreicht, erzeugt der Knoten einen Timeout-Fehler.

Ablauf beim Warten und Lesen des Gerätewertes

1. Der Knoten wartet mit `GL_WAITVAL (. . .)` darauf, dass der Gerätewert in das definierte Fenster um Schwellwert eintritt.
2. Danach wird der aktuelle Gerätewert mit `GL_VALUE (. . .)` gelesen und in die Ergebnisvariable geschrieben.

Typische Anwendungen

- Warten auf eine Sensor-Distanz, um sicherzustellen, dass ein Objekt wirklich vorhanden ist.
- Überwachung, ob ein Wert innerhalb einer definierten Toleranz bleibt (z. B. Öffnungsweite, Werkstückbreite).

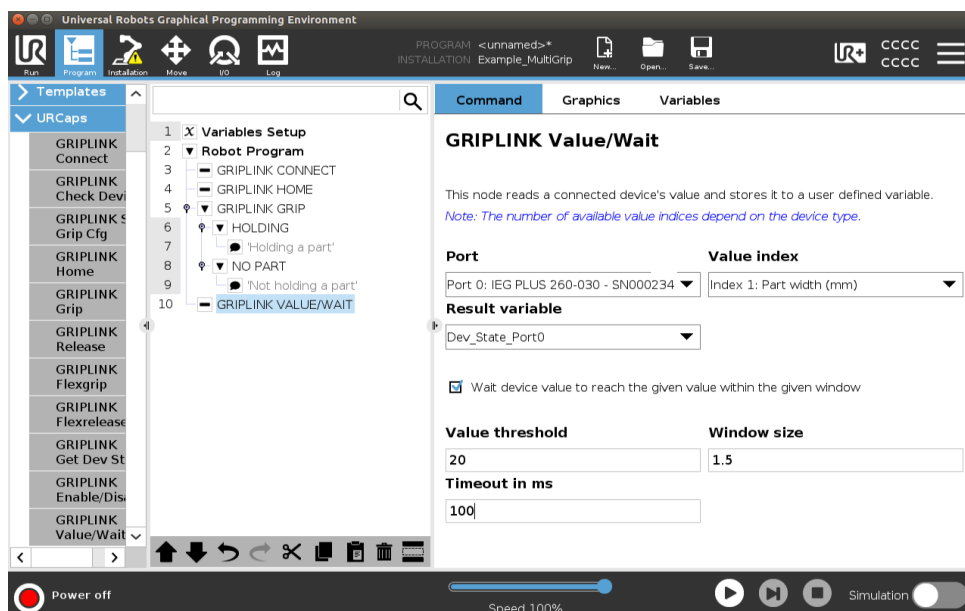


Abbildung 36: GRIPLINK VALUE/WAIT - Auf einen Gerätewert warten und anschließend den Wert auslesen und in einer Variable speichern.



Diese Funktion ist nicht auf allen Geräten verfügbar.



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

Befehlsaufruf mit URScript-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>	Double	Schwellwert, der erreicht werden soll
<WINDOW_SIZE>	Double	Fenstergröße, die um den Schwellwert zentriert den Wertebereich vorgibt, innerhalb dessen der WAITVAL-Befehl ohne Fehler abschließt
<TIMEOUT_MS>	Integer	Maximale Dauer in ms, für die auf das Erreichen des Zielwertes gewartet wird
<SOCKET_NAME>	String	Name des zu verwendenden Sockets, z. B. "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 index 0
# Value threshold 4000
# Window size 500
# 15 s => 15,000 ms
GL_WAITVAL(0,0,4000,500,15000,"sock_griplink")
```

```
# Read value 0 afterwards  
gripper0_position = GL_VALUE(0,0,"sock_griplink")
```

7.10 Konfigurieren eines Grip-Presets - GRIPLINK Set Grip Config

Abhängig vom Greifmodul können bis zu acht frei konfigurierbare Grip-Presets ausgeführt werden. Mit dem Programmknoten „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 diesem Programmknoten können der Port des gewünschten Greifers und der Index des Grip-Presets ausgewählt werden. Über die Combobox „Value Interpretation“ kann die Darstellung der Werte angepasst werden, je nachdem, welche Bedeutung die einzelnen Parameter haben. Für bestimmte Geräte werden nicht-relevante Parameter ausgeblendet.

Im Programmknoten können folgende Parameter eingestellt werden:

Port (Combobox)

Auswahl des Ports des gewünschten Greifers.

Preset Index (Combobox)

Auswahl des zu konfigurierenden Presets (0...7).

Grip Preset Tag (Texteingabe)

Freier Text, mit dem das Preset beschrieben wird (z. B. `workpiece1`). Der Tag erleichtert die Zuordnung des Grip-Presets zu einem Werkstück.

Value Interpretation (Combobox)

Festlegung, wie die einzelnen Parameterfelder interpretiert und beschriftet werden.

WEISS ROBOTICS IO-Link Gripper

Parameter werden als No Part Limit (mm), Release Limit (mm) und Force Factor (%) angezeigt.

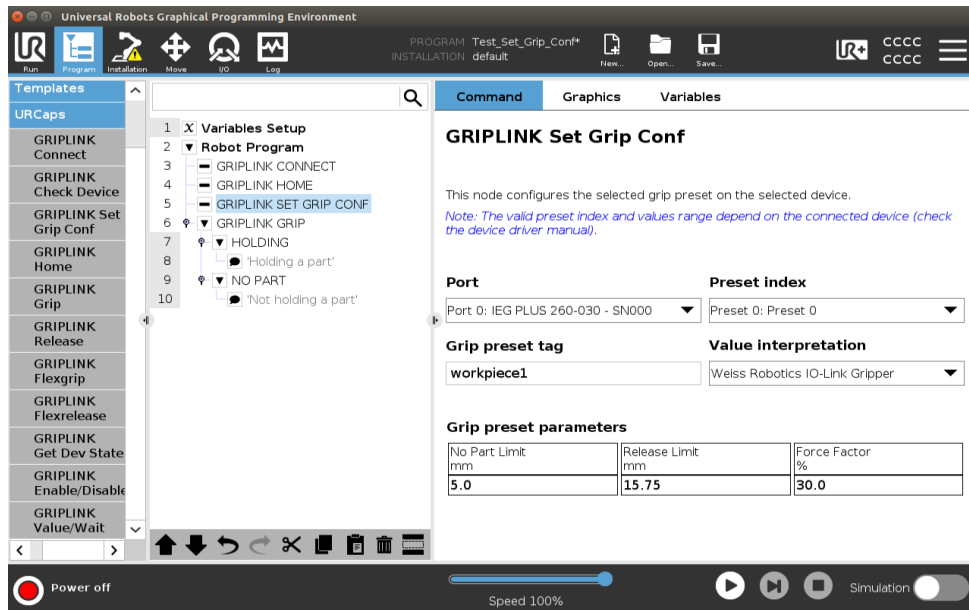


Abbildung 37: GRIPLINK SET GRIP CONF - Value Interpretation: WEISS ROBOTICS IO-Link Gripper.

WEISS ROBOTICS WPG/INTRAPAL Gripper

Parameter werden als No Part Limit (mm), Release Limit (mm), Force Factor (%), Grip Velocity (mm/s), Grip Acceleration (mm/s²), Release Velocity (mm/s) und Release Acceleration (mm/s²) angezeigt.

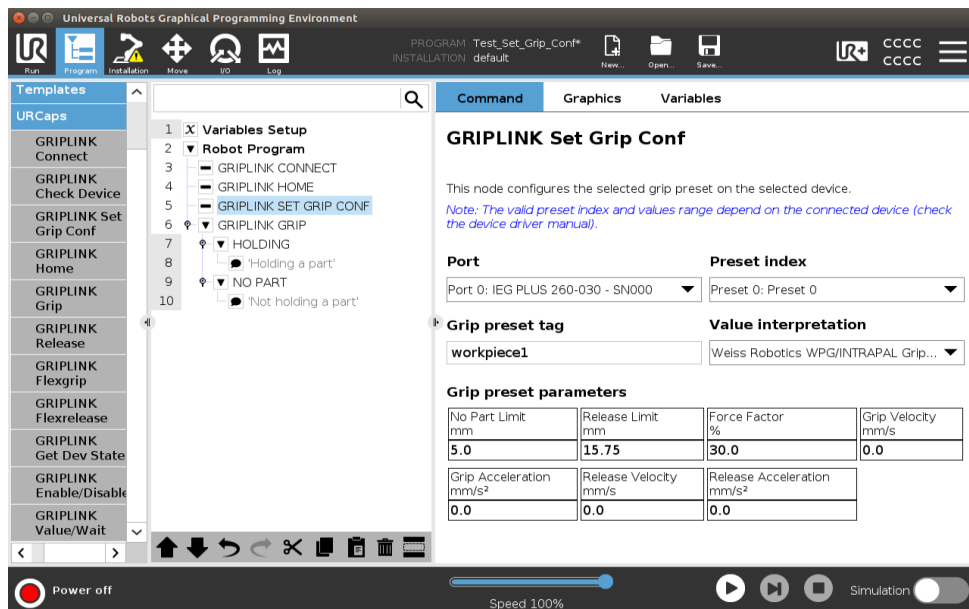


Abbildung 38: GRIPLINK SET GRIP CONF - Value Interpretation: WEISS ROBOTICS WPG/INTRAPAL Gripper.

Generic Vacuum Picker

Parameter werden als Part Present Setpoint (kPa), Energy Saving Setpoint (kPa) und Blow-off Duration (ms) dargestellt.

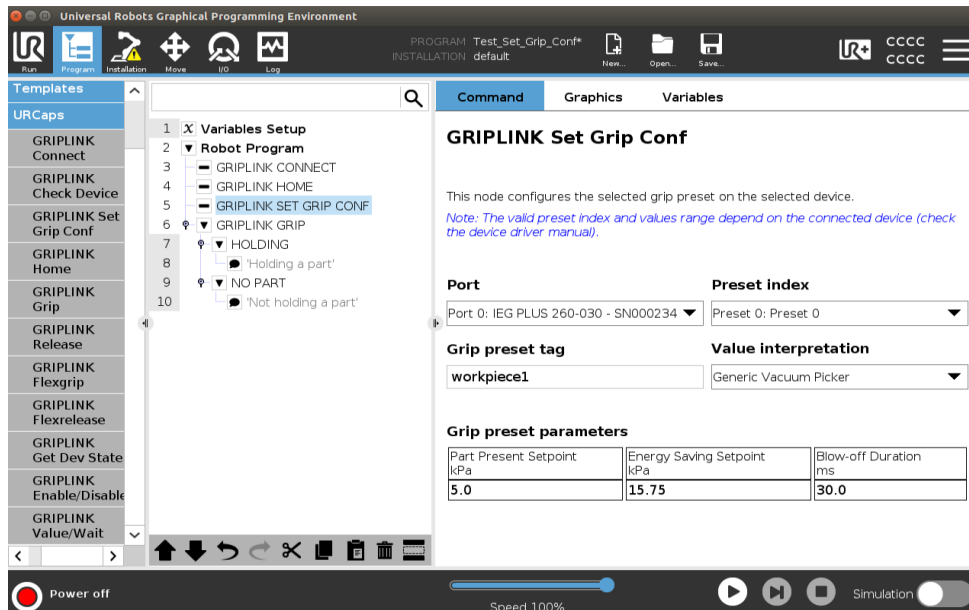


Abbildung 39: GRIPLINK SET GRIP CONF - Value Interpretation: Generic Vacuum Picker.

Generic

Alle bis zu acht verfügbaren Parameter werden neutral als Value-0 bis Value-7 angezeigt.

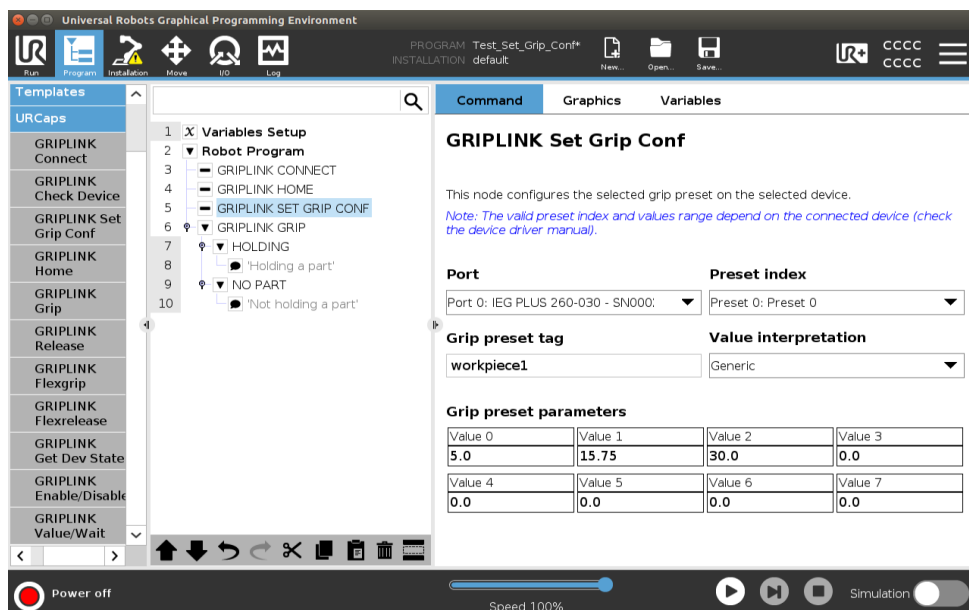


Abbildung 40: GRIPLINK SET GRIP CONF - Value Interpretation: Generic.



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



Beachten Sie, dass abhängig vom Greifmodul nicht immer alle acht konfigurierbaren 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.

Befehlsaufruf mit URScript-Code

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

Parameter	Typ	Bedeutung
<PORT>	Integer	Port-Index (0...31).
<PRESET_INDEX>	Integer	Preset-Index (0...7).
<PRESET_TAG>	String	Preset-Tag - Frei wählbarer Tag für das Preset (z. B. "workpiece1").
<PRESET_PARAMS>	Liste von Double-Werten	Liste der numerischen Parameterwerte für dieses Preset. Die Liste umfasst mindestens ein und maximal 8 Elemente.
<SOCKET_NAME>	String	Name des zu verwendenden Sockets, z. B. "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 = [5,15.75,30]

# Set grip preset 3 with the tag "workpiece1" for the device connected
to port 0
GL_SETGRIPCFG(0,3,"workpiece1",preset_params,"sock_griplink")
```

7.11 Gerätewert setzen - GRIPLINK Set Value

Einige Geräte unterstützen das Setzen von Gerätewerten, z. B. für Positionen.



Diese Funktion ist nicht auf allen Geräten verfügbar.



Das Setzen von Gerätewerten erfolgt ausschließlich über die URScript-Funktion `GL_SETVAL(...)`, z. B. in einem Script-Node.



Das Setzen von Gerätewerten kann bei Aktoren zu Bewegungen führen.

Befehlsaufruf mit URScript-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>	Double	Wert, der gesetzt werden soll
<SOCKET_NAME>	String	Name des zu verwendenden Sockets, z. B. "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  
GL_SETVAL(3,2,12,"sock_griplink")
```

7.12 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.



Die LED-Funktion ist nur bei kompatiblen Geräten verfügbar!



Die Ansteuerung des LED-Leuchtrings erfolgt ausschließlich über die URScript-Funktion `GL_LED(...)`, z. B. in einem Script-Node.

Befehlsaufruf mit URScript-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, z. B. "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")
```

7.13 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 Greifvorgänge abzuschalten und das Bauteil dennoch weiterhin sicher zu halten.



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



Die Klemmung wird ausschließlich über die URScript-Funktion `GL_CLAMP(...)` gesteuert.



Diese Funktion ist nicht auf allen Geräten verfügbar.

Befehlsaufruf mit URScript-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, z. B. "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 with preset 0 without blocking.  
GL_GRIP(0,0,0,"sock_griplink")  
  
# Hold part for longer time  
#...
```

8 Fehlersuche

Das GRIPLINK-Plugin gibt im Betrieb Fehlermeldungen aus. Im Folgenden werden typische Meldungen, mögliche Ursachen und empfohlene Maßnahmen beschrieben. Die beschriebenen Hinweise unterstützen bei Inbetriebnahme, Diagnose und Service.

8.1 Kein Zugriff auf die Tool-I/O-Schnittstelle

Mögliche Ursache:

Im Tool-I/O-Menü des Roboters wurde unter „**Ansteuerung durch**“ nicht die Option „**GRIPLINK for UR**“ ausgewählt.

Empfohlene Maßnahme:

Prüfen Sie die Einstellungen des Roboters unter „**Installation / Tool I/O**“ und wählen Sie dort „**GRIPLINK for UR**“ aus. Kontrollieren Sie anschließend im Installationsknoten des GRIPLINK-Plugins, ob der Tool-I/O-Zugriff den Status „**GRANTED**“ anzeigt.

8.2 Gerät wird im Tool-I/O-Betrieb nicht erkannt

Mögliche Ursache:

Das angeschlossene Gerät ist nicht kompatibel, nicht korrekt verdrahtet, nicht mit dem Original-Tool-I/O-Kabel angeschlossen oder es wurde noch kein erneuter Gerätescan durchgeführt.

Empfohlene Maßnahme:

Prüfen Sie Verdrahtung, Spannungsversorgung und Gerätekompatibilität. Stellen Sie sicher, dass das Original-Tool-I/O-Kabel verwendet wird. Führen Sie anschließend im Installationsknoten des GRIPLINK-Plugins die Aktion „**Rescan devices**“ aus und prüfen Sie, ob das Gerät in der Ergebnisanzeige erscheint.

8.3 Timeout bei aktivierter Option „Wait for state transitions“

Mögliche Ursache:

Der erwartete Zustandsübergang ist nicht eingetreten, weil sich das Gerät bereits im Zielzustand befand, der vorausgehende Befehl keinen Zustandswechsel ausgelöst hat oder das Gerät nicht im zulässigen Ausgangszustand war.

Empfohlene Maßnahme:

Prüfen Sie den Programmablauf und den Gerätezustand vor dem Befehl. Aktivieren Sie die Wartefunktion nur dann, wenn tatsächlich ein Zustandsübergang zu erwarten ist. Kontrollieren Sie bei Bedarf den aktuellen Gerätezustand zunächst mit **GRIPLINK Get Dev State**.

8.4 Nicht blockierende Ausführung bei GRIPKIT EASY Firmware-Versionen unter 3.0.0

Mögliche Ursache:

Für GRIPKIT EASY-Geräte mit einer Firmware-Version unter **3.0.0** ist die nicht blockierende Ausführung von Greif- und Freigabebefehlen nicht zulässig.

Empfohlene Maßnahme:

Verwenden Sie bei diesen Geräten ausschließlich die blockierende Ausführung. Prüfen Sie die eingesetzte Firmware-Version und passen Sie bestehende Programme entsprechend an.

Anhang A Gerätezustand

Die folgende Tabelle listet die möglichen Zustandswerte angeschlossener Geräte auf. Die in der rechten Spalte angegebenen Konstanten können im Roboterprogramm verwendet werden (siehe Abschnitt 7.8).

Gerätezustand	Wert	Bedeutung	Name der URScript-Konstanten
NOT CONNECTED	0	Greifmodul nicht verbunden	S_NOT_CONNECTED
NOT INITIALIZED	1	Greifmodul nicht initialisiert	S_NOT_INITIALIZED
DISABLED	2	Antrieb inaktiv Finger können manuell verschoben werden. Gerät ist 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 ist aktiviert.	S_ENABLED
FAULT	7	Fehlerzustand	S_FAULT



Weisen Sie den URScript-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!

Anhang B Statuscodes

Tritt während der Ausführung eines GRIPLINK-Befehls ein Fehler auf, wird ein entsprechender Statuscode ausgegeben. Dieser beschreibt die Ursache des Fehlers und kann zur Diagnose im Roboterprogramm oder während der Inbetriebnahme herangezogen werden. Die folgende Tabelle enthält eine Übersicht der möglichen Statuscodes sowie deren Bedeutung.

Status code	Identifier	Description
0	E_SUCCESS	No error. Command successfully executed.
1	E_OVERRUN	Data overrun
2	E_RANGE_ERROR	Value out of range
3	E_NOT_AVAILABLE	Function or data not available
4	E_NOT_INITIALIZED	Device not initialized
5	E_TIMEOUT	Timeout
6	E_INSUFFICIENT_RESOURCES	Not enough memory available
7	E_CHECKSUM_ERROR	Checksum error
8	E_ACCESS_DENIED	Access denied
9	E_INVALID_HANDLE	Invalid handle
10	E_INVALID_PARAMETER	Invalid parameter
11	E_INDEX_OUT_OF_BOUNDS	Index out of bounds
12	E_IO_ERROR	Generic I/O error
13	E_READ_ERROR	Read error
14	E_WRITE_ERROR	Write error
15	E_NOT_FOUND	Resource not found
16	E_NOT_OPEN	File or device not open
17	E_EXISTS	Resource already exists
18	E_NO_COMM	Connection error
19	E_STATE_CONFLICT	Invalid state
20	E_NOT_SUPPORTED	Command or function not supported

21	E_INCONSISTENT_DATA	Data inconsistent
22	E_CMD_SYNTAX	Syntax error
23	E_CMD_UNKNOWN	Unknown command
24	E_CMD_ABORTED	Command aborted
25	E_CMD_FAILED	Command failed
26	E_AXIS_BLOCKED	Axis is blocked
27	E_PENDING	Pending action / Not yet finished

Anhang C Tool-I/O-spezifische Statuscodes

Die Statuscodes ab 100 sind Tool-I/O-spezifische Fehlercodes des URCaps und unterstützen die Diagnose und Fehlerbehandlung in der Tool-I/O-Schnittstelle.

Statuscode	Bedeutung
100	Zugriff auf die Tool-I/O-Schnittstelle wurde nicht erteilt.
101	Ein zuvor erteilter Zugriff auf die Tool-I/O-Schnittstelle wurde entzogen.
102	Ein Gerätescan läuft bereits. Bitte dessen Abschluss abwarten.
103	Die Tool-I/O-Schnittstelle wird gerade initialisiert.
104	Die Tool-I/O-Schnittstelle wird gerade beendet.

Anhang D Versionsverlauf

Datum	Version	Typ	Beschreibung	Autor
06.04.2026	3.0.0	Hinzugefügt	Unterstützung von GRIPKIT-Geräten und Geräten der INTRAPAL-Serie über Tool-I/O (Modbus RTU).	H. E.
06.04.2026	3.0.0	(BREAKING)	Ethernet-basierte Kommunikation (TCP/IP) und Tool-I/O-basierte Kommunikation (Modbus RTU) wurden in einem gemeinsamen Plugin zusammengeführt. Das bisherige FlexGrip-Plugin wird abgekündigt und durch das GRIPLINK-Plugin ersetzt.	H. E.
06.04.2026	3.0.0	Hinzugefügt	Türkische Lokalisierung (TR).	H. E.
16.12.2025	2.2.0	Hinzugefügt	Demo-Programme für PolyScope	H. E.
16.12.2025	2.2.0	Hinzugefügt	Port-Konfiguration und Logging-Konfiguration.	H. E.
16.12.2025	2.2.0	Hinzugefügt	GRIPLINK Toolbar (IGRIP/PGRIP/VALUE).	H. E.
16.12.2025	2.2.0	Geändert	Überarbeitung der Programmknoten: <ul style="list-style-type: none"> • Port-Combobox mit Online-Infos. • Gemeinsamer Enable/Disable-Knoten • DO_BLOCK für GRIP/RELEASE und FLEXGRIP/FLEXRELEASE • Gleitkomma-Unterstützung für FLEXGRIP/FLEXRELEASE und VALUE/WAIT • Variablenauswahl für VALUE/WAIT und Get Dev State sowie Zusammenführung von WAITVAL und VALUE in einem Knoten. 	H. E.
16.12.2025	2.2.0	Entfernt	Programmknoten für MULTIGRIP, MULTIRELEASE und Get Position	H. E.

© 2026 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.