



GRIPLINK-PLUGIN FÜR UNIVERSAL ROBOTS POLYSCOPE X

Version 1.0.0

Juli 2026



Inhaltsverzeichnis

1 Technische Übersicht und Sicherheit.....	3
1.1 Einführung.....	3
1.1.1 Notation und Symbole.....	3
1.1.2 Bestimmungsgemäße Verwendung.....	3
1.1.3 Systemvoraussetzungen.....	4
1.1.4 Lizenzbestimmungen.....	4
1.1.5 Demo-Programme.....	5
1.2 Plugin-Struktur im Überblick.....	6
1.3 Verfügbare Kommunikationsschnittstellen.....	7
1.4 Versions- und Migrationshinweise.....	7
1.5 Not-Halt.....	7
2 Installation und Inbetriebnahme.....	8
2.1 Installation.....	8
2.2 Inbetriebnahme: Ethernet.....	9
2.3 Inbetriebnahme: Tool I/O.....	10
3 Applikationsknoten und Setup.....	13
4 Tool-I/O-Integration.....	16
5 Manuelle Bedienung über Sidebar und Smart Skills.....	21
5.1 Sidebar.....	21
5.2 Smart Skills.....	27
6 Programmieren mit Programmknoden.....	33
6.1 Wie die Befehlsreferenz zu lesen ist.....	33
6.2 Programmknoden-Befehlsreferenz.....	34
6.3 Verbindung aufbauen – CONNECT.....	36
6.4 Verbindung trennen – DISCONNECT.....	37
6.5 Ein angeschlossenes Gerät prüfen – ASSERTDEVICE.....	37
6.6 Gerät referenzieren – HOME.....	37
6.7 Gerät aktivieren – ENABLE.....	37
6.8 Gerät deaktivieren – DISABLE.....	38
6.9 Werkstück greifen – GRIP.....	38
6.10 Werkstück freigeben – RELEASE.....	38
6.11 Werkstück über Parameter greifen – FLEXGRIP.....	39
6.12 Werkstück über Parameter freigeben – FLEXRELEASE.....	39
6.13 Pick / Place.....	40
6.14 Greifkrafterhaltung steuern – CLAMP.....	41
6.15 Ansteuerung der Status-LED – LED.....	41

6.16	Aktuellen Gerätezustand abfragen – DEVICESTATE.....	41
6.17	Gerätewert abfragen – VALUE.....	42
6.18	Warten auf eine Änderung des Gerätezustands – WSTR.....	43
6.19	Warten auf einen Gerätewert – WAITVAL.....	43
6.20	Gerätewert setzen – SETVALUE.....	44
6.21	Parametrierung eines Griff-Presets – SGRIPCFG.....	44
7	URScripting.....	47
7.1	Verbindung aufbauen – GL_CONNECT.....	47
7.2	Verbindung trennen – GL_DISCONNECT.....	48
7.3	Ein angeschlossenes Gerät prüfen – GL_DEVASSERT.....	49
7.4	Gerät referenzieren – GL_HOME.....	49
7.5	Gerät aktivieren – GL_ENABLE.....	50
7.6	Gerät deaktivieren – GL_DISABLE.....	50
7.7	Werkstück greifen – GL_GRIP.....	51
7.8	Werkstück freigeben – GL_RELEASE.....	52
7.9	Werkstück über Parameter greifen – GL_FLEXGRIP.....	53
7.10	Werkstück über Parameter freigeben – GL_FLEXRELEASE.....	53
7.11	Greifkrafterhaltung steuern – GL_CLAMP.....	54
7.12	Ansteuerung der Status-LED – GL_LED.....	55
7.13	Aktuellen Gerätezustand abfragen – GL_DEVSTATE.....	56
7.14	Warten auf eine Änderung des Gerätezustands – GL_WSTR.....	56
7.15	Gerätewert abfragen – GL_VALUE.....	57
7.16	Warten auf einen Gerätewert – GL_WAITVAL.....	58
7.17	Gerätewert setzen – GL_SETVAL.....	59
7.18	Parametrierung eines Griff-Presets – GL_SETGRIPCFG.....	59
8	Fehlersuche und Diagnose.....	61
8.1	Fehlersuche-Matrix.....	61
8.2	Kein Zugriff auf die Tool-I/O-Schnittstelle.....	63
8.3	Gerät wird im Tool-I/O-Betrieb nicht erkannt.....	64
8.4	Timeout bei aktivierter Option „Auf Zustandsübergang warten“.....	64
8.5	Nicht blockierende Ausführung bei GRIPKIT-EASY-Firmware-Versionen unter 3.0.0.....	65
9	Anhänge.....	66
9.1	Gerätezustand.....	66
9.2	Statuscodes.....	66
9.3	Tool-I/O-spezifische Statuscode-Aktionen.....	68
9.4	Versionsverlauf.....	68

1 Technische Übersicht und Sicherheit

1.1 Einführung

Mit der GRIPLINK-Technologie können Automationskomponenten von WEISS ROBOTICS über eine standardisierte Kommunikationsschnittstelle in Robotersysteme integriert werden. Das GRIPLINK-Plugin für Universal Robots PolyScope X ist das steuerungsseitige Bindeglied zu Geräten von WEISS ROBOTICS und zu kompatiblen Drittanbietergeräten aus dem GRIPLINK-Ökosystem, etwa Sensoren und Aktoren. Es bindet diese Geräte an Robotersysteme von Universal Robots mit PolyScope X an und bedient sie.

Das GRIPLINK-Plugin unterstützt sowohl Geräte, die das GRIPLINK-Protokoll nativ über TCP/IP verwenden (**Ethernet**), als auch kompatible Modbus-RTU-Geräte, die über die **Tool-I/O**-Schnittstelle der Steuerung von Universal Robots 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 GRIPLINK Controllers entnehmen Sie der Betriebsanleitung des jeweiligen Moduls. Diese finden Sie online unter www.griplink.de.

1.1.1 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.

1.1.2 Bestimmungsgemäße Verwendung

Die Software „GRIPLINK-Plugin“ ist zur Kommunikation zwischen dem GRIPLINK Controller von WEISS ROBOTICS und einer Robotersteuerung bestimmt. Sie ermöglicht die Integration und Steuerung von GRIPLINK-kompatiblen Geräten über die standardisierte GRIPLINK-Kommunikationsschnittstelle.

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.

1.1.3 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. Das Plugin ist ein **URCapX**-Paket, das Nachfolgeformat des ursprünglichen URCap auf PolyScope 5.

Zum Betrieb wird eine der folgenden Universal Robots Robotersteuerungen benötigt:

- Control Box CB5.6 mit Softwarestand 10.14 (oder höher)

Schnittstelle	Unterstützte Geräte / Firmware	Lizenz
Ethernet (TCP/IP)	GRIPLINK-ET4 ab FW 5.2.0; WPG-Serie ab FW 2.4.0; INTRAPAL-Serie ab FW 5.2.0	–
Tool I/O (Modbus RTU)	GRIPKIT EASY ab FW 2.1.1; GRIPKIT LS ab FW 1.1.0; INTRAPAL-Serie ab FW 5.3.0	OPT-GKEASY-MB für GRIPKIT EASY; OPT-IPL-MB für INTRAPAL-Serie



OPT-GKEASY-MB und OPT-IPL-MB sind Geräteoptionen, die die Modbus-RTU-Schnittstelle (Tool I/O) auf dem jeweiligen Gerät freischalten. Sie werden bei WEISS ROBOTICS bestellt; Details siehe Betriebsanleitung des Geräts.



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.



Wichtiger Hinweis: Eine Aktualisierung des URCapX 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 den Versionsverlauf.



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

1.1.4 Lizenzbestimmungen

Das GRIPLINK-Plugin ist urheberrechtlich geschützt. Die jeweils gültigen Lizenzbestimmungen liegen dem Softwarepaket bei und sind zudem im Applikationsknoten unter **License Agreement** verfügbar. Mit der Installation akzeptieren Sie diese Lizenzbestimmungen.

1.1.5 Demo-Programme

Das Softwarepaket enthält eine Reihe von Beispielprogrammen, die die häufigsten Integrationsmuster demonstrieren. Sie sind ausschließlich für Testzwecke und als Ausgangspunkt für die eigene Anwendung gedacht. Die Beschreibungen verwenden die GRIPLINK-Befehle und Gerätezustände aus den Programmier-Kapiteln und dem Anhang Gerätezustand; diese Liste dient als Überblick und lohnt einen zweiten Blick nach der Lektüre dieser Kapitel.

Jedes Beispiel liegt als natives PolyScope-X-Programm vor (URProgram/* .urpx, aufgebaut aus GRIPLINK-Programmknotten) und - mit Ausnahme von pick_and_place, dessen Ergebniszweige nur als Programmknotten existieren - zusätzlich als URScript-Datei (URScript/* .script). Beide Formen führen dieselbe Abfolge aus, daher gelten die folgenden Beschreibungen für beide.

simple_grip_release

Ein einzelner blockierender Greif- und Freigabezyklus mit einem gespeicherten Preset. Das Programm liest den Gerätezustand, initialisiert das Gerät bei Bedarf (Home und Enable), öffnet die Finger, greift und klassifiziert das Ergebnis als HOLDING, NO PART oder Fehler; anschließend gibt es frei und prüft den Zustand RELEASED. Der empfohlene Ausgangspunkt für eine neue Integration.

griplink_demo

Führt eine konfigurierbare Anzahl aufeinanderfolgender blockierender Greif-/Freigabezyklen aus und wertet nach jedem Greifen den Gerätezustand aus. Ein Zyklus, der HOLDING nicht erreicht, zeigt eine Warnung mit Zyklus und Zustand und bricht die Schleife ab.

flexgrip_cycle

Ein Greif-/Freigabezyklus, bei dem die Bewegungsparameter (Position, Kraft, Geschwindigkeit, Beschleunigung) über FLEXGRIP und FLEXRELEASE in jedem Aufruf explizit angegeben werden statt über ein gespeichertes Preset. Nützlich, wenn dasselbe Programm ohne vorkonfigurierte Presets unterschiedliche Teile handhaben soll.

async_preposition

Überlappt die Fingerbewegung mit der Roboterfahrt, um die Zykluszeit zu verkürzen: Ein nicht blockierendes FLEXRELEASE setzt die Finger in Richtung einer Zielposition in Bewegung, während der Roboter fährt; anschließend wartet WAITVAL, bis die Fingerposition (Wert-Index 0) das Zielfenster erreicht. WAITVAL wird statt WSTR verwendet, weil es auf ein Wertefenster statt auf einen Zustandsübergang wartet und damit auch dann abschließt, wenn sich das Gerät bereits am Ziel befindet.

check_workpiece

Greift ein Werkstück und nutzt die Fingerposition als Messgröße, um es anzunehmen oder abzuweisen. Nach dem Greifen bricht das Programm bei NO PART oder FAULT ab; andernfalls

liest es die Fingerposition über VALUE-Index 0 und vergleicht sie mit einem Toleranzband, meldet das Teil als ANGENOMMEN (mit dem Messwert) oder ABGEWIESEN und gibt es dann frei.

multi_grip_release

Greift und gibt die Geräte an einem Portbereich gleichzeitig frei, mit nicht blockierenden Befehlen und Synchronisierung je Port: Alle Greifbefehle werden nicht blockierend abgesetzt, danach bestätigt ein WSTR an jedem belegten Port den Zustand HOLDING; dasselbe Muster wird für die Freigabe wiederholt. Ein Port, der den erwarteten Zustand nicht erreicht, wird protokolliert.

fault_clearing_via_disable

Erkennt Fehlerstatus und quittiert ihn mit DISABLE gefolgt von HOME; anschließend wird der Zustand erneut gelesen, um zu bestätigen, ob der Fehler behoben wurde. Auch die Fälle „nicht verbunden“ und „nicht im Fehlerzustand“ werden gemeldet.

pick_and_place

Eine Pick-and-Place-Sequenz, aufgebaut aus den GRIPLINK-Programmknotten **Pick** und **Place**: Pick greift das Teil und verzweigt nach HOLDING / NO PART, Place gibt es frei und verzweigt nach RELEASED / NOT RELEASED. Wie bei den übrigen Beispielen ist die Roboterfahrt als Platzhalter-Wartezeit abstrahiert.

1.2 Plugin-Struktur im Überblick

Komponente	Navigation	Hauptzweck
Applikationsknoten	Anwendung-Tab → URCaps → WEISS ROBOTICS GRIPLINK	Plugin-Einstellungen.
Sidebar	Seitenleiste von PolyScope X	Manuelle Live-Steuerung für Inbetriebnahme, Test, Diagnose.
Smart Skills	Bewegen-Ansicht (Smart-Skills-Tab)	Wiederverwendbare manuelle Aktionen: GRIP, RELEASE, HOME.
Programmknotten	Programm-Ansicht → Commands → URCaps-Gruppe	Grafische Programmierung mit dem vereinheitlichten GRIPLINK-Knoten und den Pick-/Place-Knoten.
URScript-Funktionen	Skriptknoten / erzeugte Programm-Präambel	Erweiterte Programmierung über integrierte URScript-Aufrufe.

Demo-Programme	Softwarepaket	Acht Beispielprogramme, vom einfachen Greifzyklus bis zum vollständigen Pick-and-Place.
----------------	---------------	---

1.3 Verfügbare Kommunikationsschnittstellen

Schnittstelle	Geräte
Ethernet	Geräte mit GRIPLINK Controller, z. B. GRIPLINK-ET4, INTRAPAL-Serie und WPG-Serie.
Tool I/O	Geräte mit FLEXGRIP-Protokoll, z. B. GRIPKIT EASY und GRIPKIT LS-40-050.

1.4 Versions- und Migrationshinweise

Bestehende PolyScope-5-Projekte in URScript lassen sich nach PolyScope X migrieren, da die GRIPLINK-Bibliothek in PolyScope X dieselben Aufrufe bereitstellt wie in PolyScope 5. UI-basierte Programme (URProgram) aus PolyScope 5 erfordern eine Migration. Der Funktionsumfang des PolyScope-5-Plugins wird auch vom PolyScope-X-Plugin abgedeckt. Das Plugin unterstützt unter PolyScope X sowohl das GRIPLINK- als auch das FLEXGRIP-Protokoll.

1.5 Not-Halt

Ein Not-Halt trennt sofort die Spannungsversorgung des Roboterarms. Ob dabei auch das angeschlossene Gerät die Spannung verliert, hängt von seiner Spannungsversorgung ab:

Spannungsversorgung	Verhalten bei Not-Halt
Externe Versorgung	Das Gerät nutzt eine eigene externe Spannungsversorgung. Ob diese bei einem Not-Halt abgeschaltet wird oder eingeschaltet bleibt, ist die Entscheidung des Anlagenplaners.
Versorgung über den Tool Connector	Die Spannungsversorgung über den Tool Connector wird bei einem Not-Halt abgeschaltet; damit ist auch das Gerät spannungsfrei.



Das Plugin meldet für ein spannungsloses Gerät - wie für jede andere Unterbrechung der Tool-I/O-Verbindung - Statuscode 18 (NO_COMM). Um die Kommunikation wiederherzustellen, sobald die Spannung zurückkehrt, drücken Sie **Rescan** im Anwendung-Tab.

2 Installation und Inbetriebnahme

2.1 Installation

Das GRIPLINK-Plugin wird als **URCapX**-Paket ausgeliefert (Dateiendung .urcapx).

Plugin installieren

1. griplink-1.0.0.urcapx in das Stammverzeichnis eines USB-Sticks kopieren (oder die Datei PolyScope X anderweitig bereitstellen).
2. **Hamburger-Menü → System Manager → URCaps** öffnen. Dieser Bereich ist passwortgeschützt; für die Installation das Admin-Passwort eingeben.
3. **+ URCap** drücken und die Datei griplink-1.0.0.urcapx auswählen.

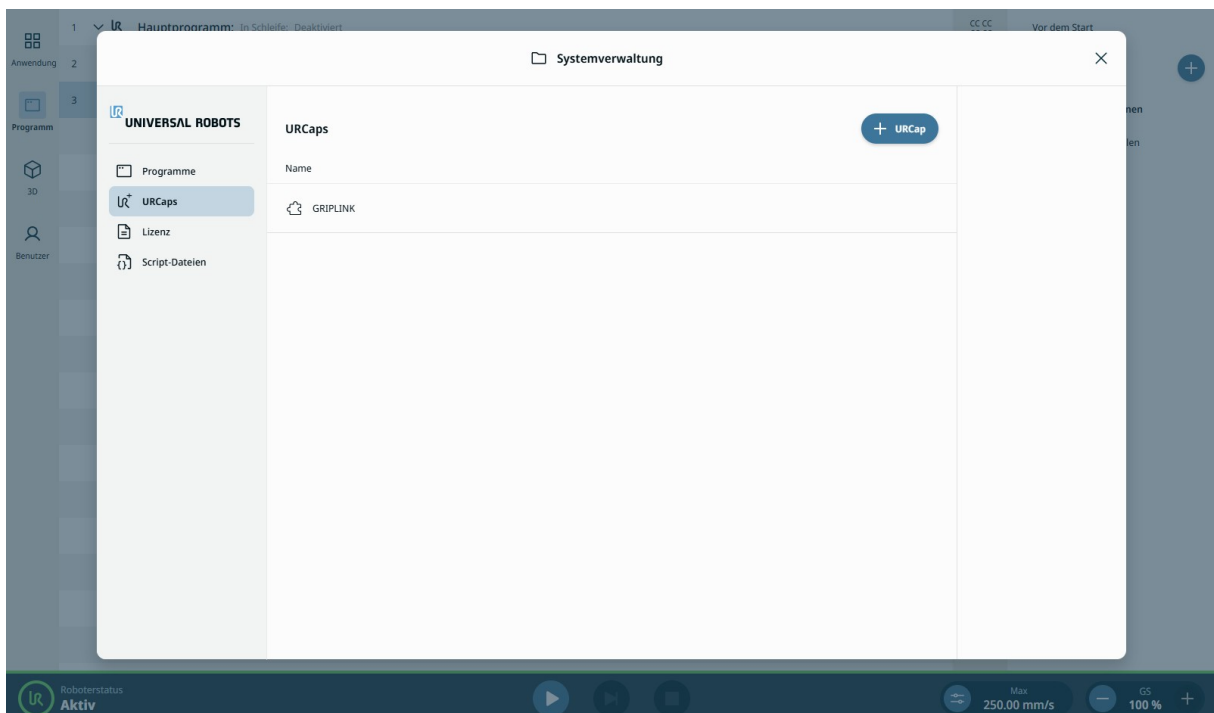


Abbildung 1: System Manager - URCaps: GRIPLINK URCapX installieren.

4. Die Installation bestätigen und den Roboter bei Aufforderung neu starten.

Software deinstallieren

1. **Hamburger-Menü → System Manager → URCaps** öffnen und mit dem Admin-Passwort entsperren.
2. GRIPLINK auswählen.
3. Die Schaltfläche **Entfernen** drücken.

4. Den Roboter neu starten.

2.2 Inbetriebnahme: Ethernet

1. Robotersteuerung und GRIPLINK Controller an dasselbe Netzwerk anschließen (gleiches Subnetz).
2. Die Netzwerkeinstellungen des Roboters konfigurieren: das **Hamburger-Menü** oben links in der Kopfzeile antippen und **Settings** wählen. Die Seitenleiste öffnen und den Tab **Connection** auswählen, dann **Network** aufrufen. Auf eine statische IP-Adresse umstellen und die gewünschte IP-Adresse des Roboters sowie die Subnetzmaske setzen.
3. **Anwendung-Tab → URCaps → WEISS ROBOTICS GRIPLINK → Setup** öffnen.
4. **Interface** auf **Ethernet** stellen.
5. Die Controller-IP-Adresse eingeben (Standard 192 . 168 . 1 . 40, falls unverändert).
6. **GRIPLINK-Controller prüfen** drücken. Als Rückmeldung erscheint ein Popup mit der eingestellten IP-Adresse oder der Meldung, dass der Controller nicht erreichbar ist.



Die IP-Adresse des GRIPLINK Controllers muss im selben Subnetz liegen wie die Control Box. Die IP-Adresse des GRIPLINK Controllers kann über dessen Weboberfläche geändert werden; die Werkseinstellung ist 192 . 168 . 1 . 40.

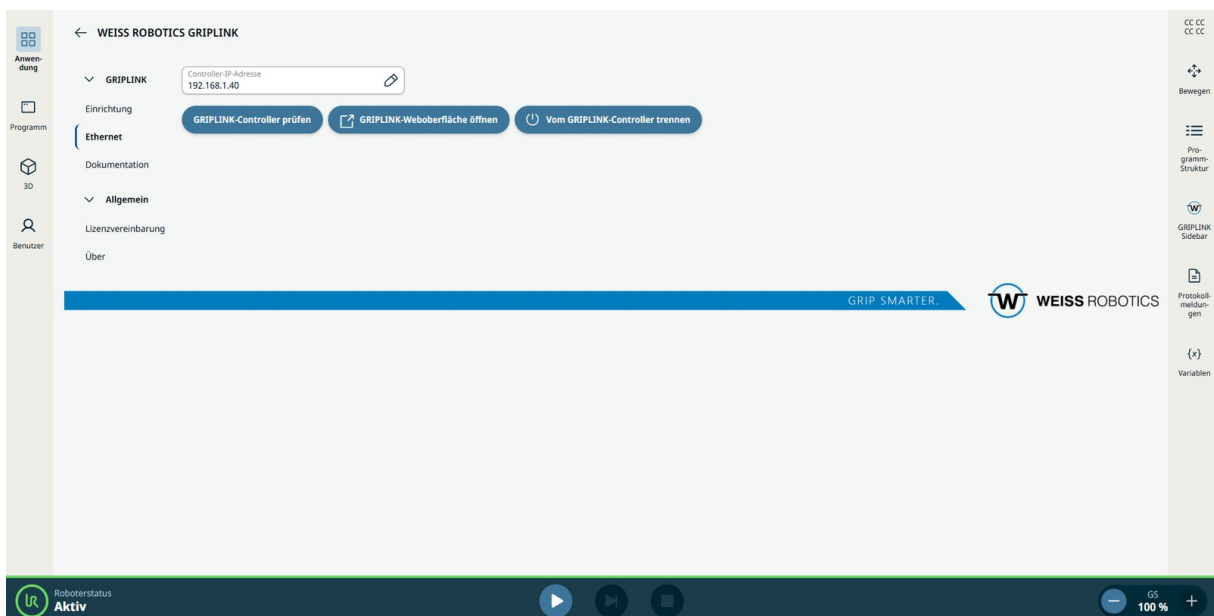


Abbildung 2: Ethernet-Inbetriebnahme im GRIPLINK-Setup - Controller IP Address und Aktionen.

2.3 Inbetriebnahme: Tool I/O

1. Die Tool-I/O-Schnittstelle vor dem Anschließen des Geräts spannungsfrei schalten (0 V); siehe die Hotplug-Warnung am Ende dieses Abschnitts.
2. Das Gerät montieren und das originale Tool-I/O-Kabel anschließen.
3. **Anwendung-Tab → URCaps → WEISS ROBOTICS GRIPLINK → Setup** öffnen.
4. **Interface** auf **Tool I/O** stellen. Das Plugin konfiguriert 24 V und Tool-I/O-Kommunikation automatisch.
5. **Anwendung-Tab → Communication → Robot → Tool I/O** öffnen und **Controlled By** von **User** auf **GRIPLINK** umstellen.
6. **Rescan** drücken. Die erkannten Geräte erscheinen mit Seriennummer und Firmware.

Zur Wiederherstellung der Tool-I/O-Schnittstelle die Schaltfläche **Restart Interface** verwenden und erneut scannen. Werden keine Geräte gefunden, die Schritte 1 bis 5 erneut durchführen. Montage, Verkabelung sowie das An- und Abstecken von Geräten sind im Kapitel Tool-I/O-Integration ausführlich beschrieben.

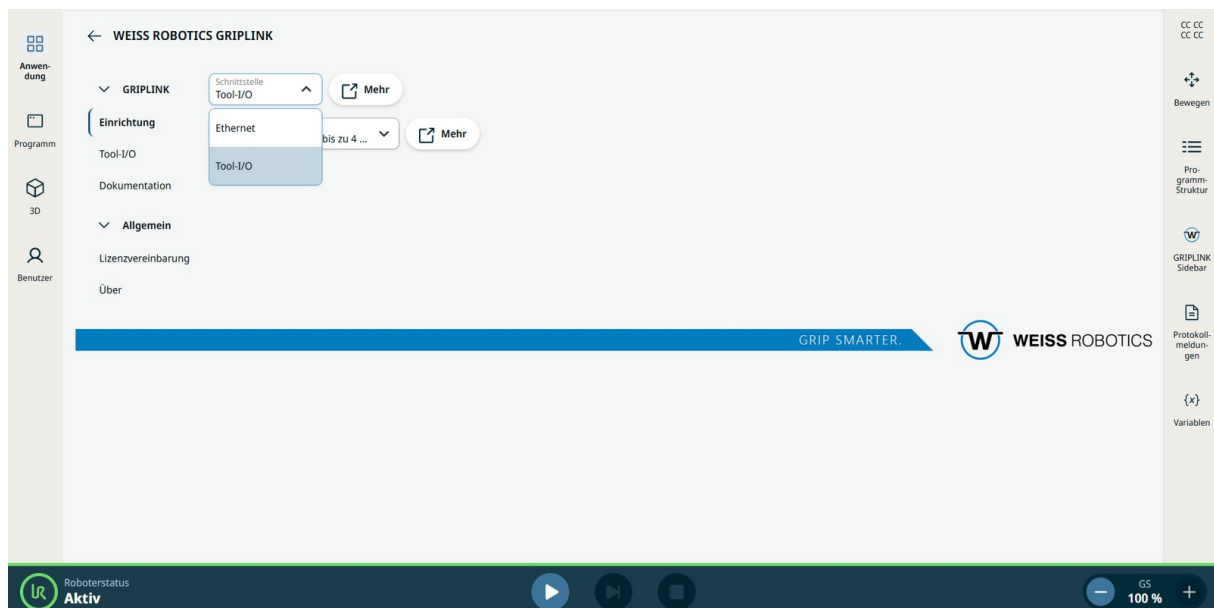


Abbildung 3: Interface-Auswahl im GRIPLINK-Setup - Tool I/O wählen.

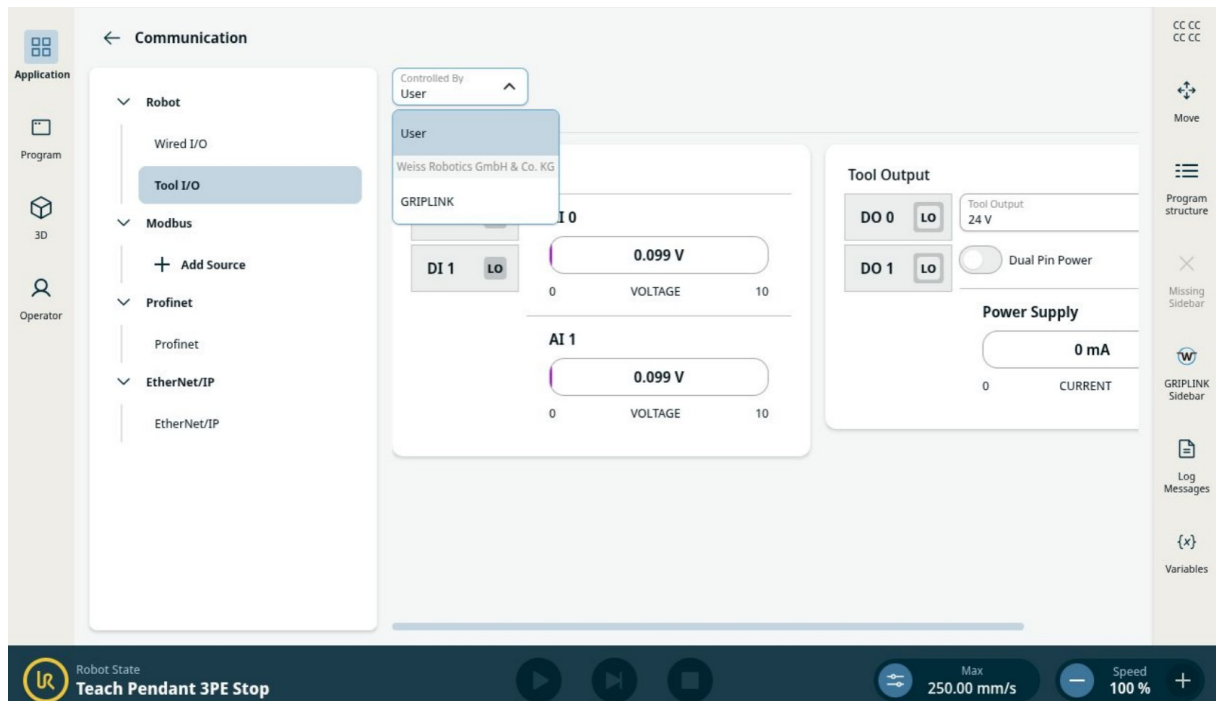


Abbildung 4: Robotereinstellungen Communication → Robot → Tool I/O - Controlled By auf GRIPLINK stellen.

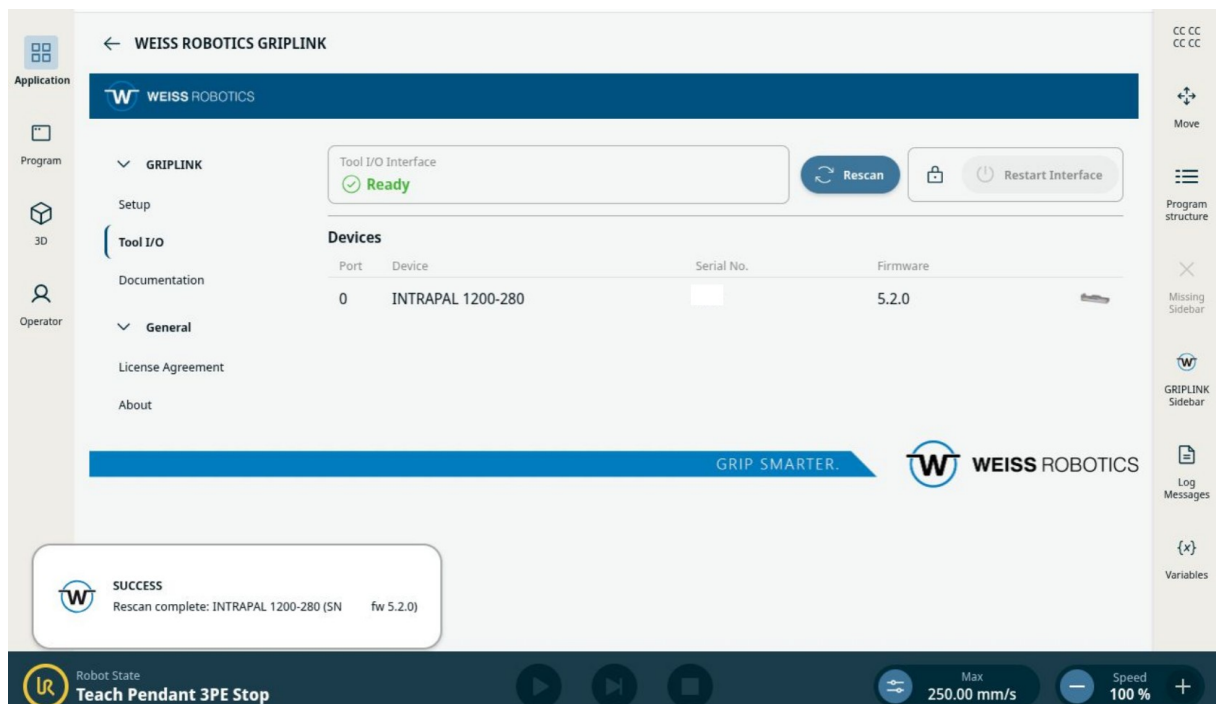


Abbildung 5: Erfolgreicher Rescan - das erkannte Gerät erscheint mit Seriennummer und Firmware.



Werkzeuge oder Tool-Kabeladapter niemals „hotpluggen“ (anschießen oder trennen), während der Tool Connector mit Strom versorgt ist. Dies kann den internen

Werkzeugflansch-Anschluss des Roboters oder den Endeffektor dauerhaft beschädigen.
Den Roboter vor dem Anbringen oder Abnehmen von Werkzeugen immer ausschalten.

Verhalten im Fehlerfall

Tritt im Plugin oder bei der Gerätekommunikation ein Fehler auf, stoppt das laufende Programm, jede Roboterbewegung stoppt, und eine Fehlermeldung erscheint; das Programm läuft nicht weiter, solange der Fehler ansteht. Typische Ursachen: eine unterbrochene Controller-Verbindung (spannungsloser GRIPLINK Controller, nicht angeschlossenes Ethernet-Kabel, falsche Netzwerkkonfiguration), ein ungültiger Parameter oder ein Gerät, das einen Fehler meldet, einschließlich eines Geräts, das sich bereits im Zustand FAULT befindet oder in diesen wechselt. Das Plugin zeigt ein Popup mit dem fehlgeschlagenen Befehl und einer **Retry**-Option.

3 Applikationsknoten und Setup

Konfigurieren Sie die Geräteverbindung im Applikationsknoten **WEISS ROBOTICS GRIPLINK: Anwendung-Tab → URCaps → WEISS ROBOTICS GRIPLINK**. Die Seite zeigt links die Navigationsleiste und rechts den Konfigurationsinhalt.

GRIPLINK fügt einen Eintrag unter **Anwendung → URCaps** hinzu.

Gruppe	Einträge
GRIPLINK	Setup der Kommunikationsschnittstelle (Ethernet oder Tool I/O), Dokumentation (öffnet dieses Handbuch als PDF).
General	License Agreement , About (Plugin-Version, Beschreibung und unterstützte Befehle).

Setup

Die Setup-Seite enthält nur zwei Auswahlfelder, **Interface** und **Port Configuration**, und legt fest, wie das Plugin das Gerät erreicht. Wählen Sie zuerst **Interface**: Je nach gewähltem Modus erscheint ein passender Eintrag **Ethernet** oder **Tool I/O** in der Navigationsleiste, mit den Feldern dieses Modus auf einer eigenen Seite.

Interface	Bedeutung
Ethernet	GRIPLINK kommuniziert über das Roboternetzwerk mit einem externen GRIPLINK Controller unter der konfigurierten IP-Adresse.
Tool I/O	Das Gerät ist direkt an die Tool-I/O-Schnittstelle des Roboters angeschlossen; es ist kein externer GRIPLINK-Controller nötig.

Das Auswahlfeld **Port Configuration** (in beiden Modi vorhanden) legt fest, wie viele Ports adressierbar sind:

Port Configuration	Bedeutung
Standalone Controller	Die Robotersteuerung kommuniziert mit einem einzelnen GRIPLINK Controller; Ports 0-3 sind in den Programmknoten wählbar.
Proxy Controller	Stellt einen GRIPLINK Controller mit bis zu 32 Ports bereit; Ports 0-31 sind adressierbar.

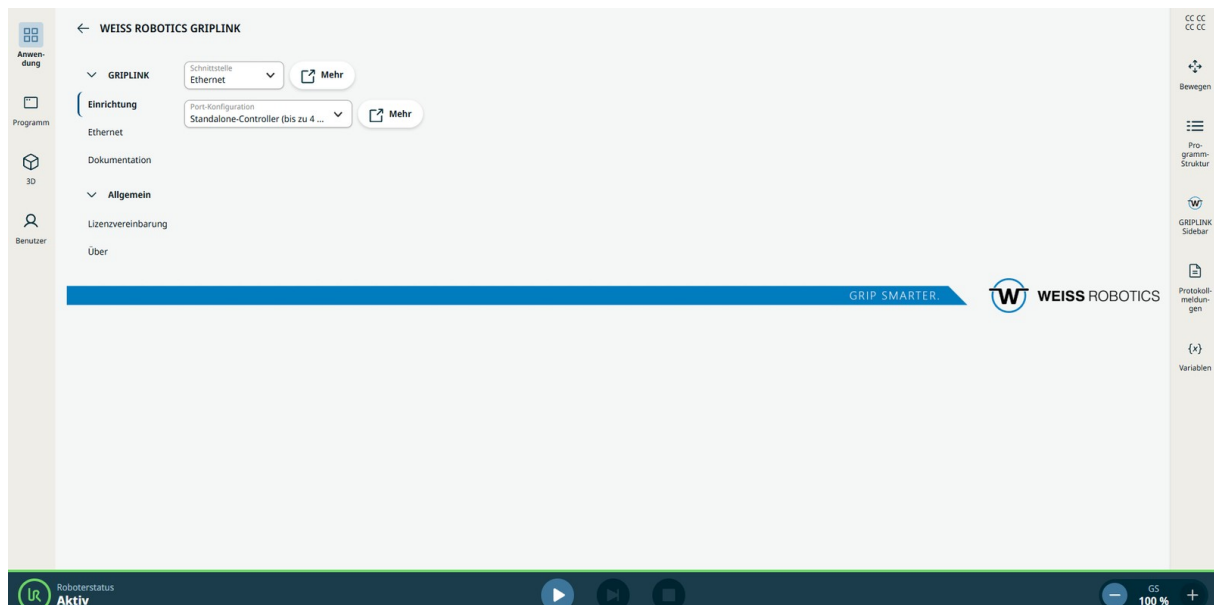


Abbildung 6: Die Setup-Seite - Auswahlfelder Interface und Port Configuration.

Ethernet-Betrieb

Ist **Interface** auf **Ethernet** gestellt, erscheint in der Navigationsleiste eine eigene **Ethernet**-Seite mit dem Feld **Controller IP Address** (Standard 192.168.1.40) sowie den Aktionen **GRIPLINK-Controller prüfen**, **GRIPLINK-Weboberfläche öffnen** und **Vom GRIPLINK-Controller trennen**.

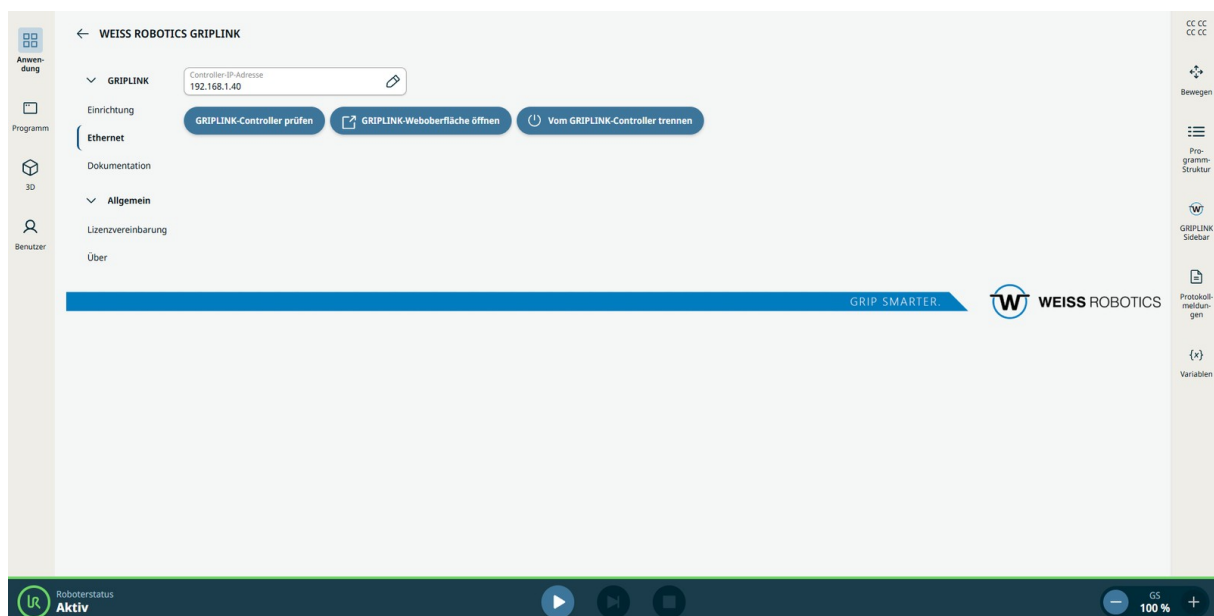


Abbildung 7: Ethernet-Seite - Controller-IP-Adresse und Aktionen.

Aktion	Beschreibung
GRIPLINK-Controller prüfen	Prüft, ob der Controller erreichbar ist, und

	meldet Typ, Seriennummer und Firmware. Als Rückmeldung erscheint ein Popup mit der eingestellten IP-Adresse oder der Meldung, dass der Controller nicht erreichbar ist.
GRIPLINK-Weboberfläche öffnen	Öffnet die Weboberfläche des Controllers im Chromium-Browser; verfügbar, sobald der Controller erreichbar ist.
Vom GRIPLINK-Controller trennen	Schließt den offenen GRIPLINK-Socket beim nächsten Programmlauf, nach einer Bestätigung „Disconnect from GRIPLINK?“.



Konfigurieren Sie den GRIPLINK Controller selbst über seine Weboberfläche, über die Schaltfläche **GRIPLINK-Weboberfläche öffnen** auf der Ethernet-Seite oder direkt im Browser unter seiner IP-Adresse (Standard 192 . 168 . 1 . 40).

Tool-I/O-Betrieb

Ist **Interface** auf **Tool I/O** gestellt, erscheint in der Navigationsleiste eine eigene **Tool I/O**-Seite mit dem Schnittstellenzustand, der Aktion **Rescan** und der Gerätetabelle. Diese werden zusammen mit Flanschverkabelung und Stromversorgung im Kapitel Tool-I/O-Integration behandelt.

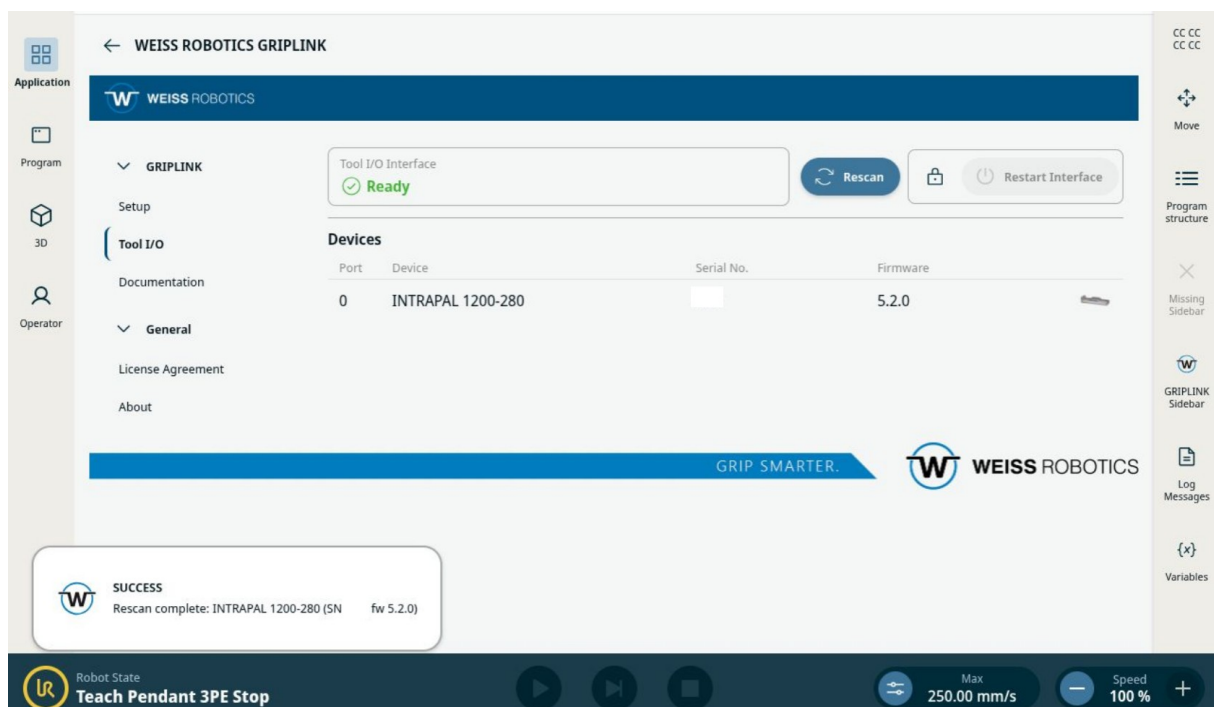


Abbildung 8: Tool-I/O-Seite mit erkanntem Gerät - Zustand Ready, INTRAPAL 1200-280 an Port 0 nach erfolgreichem Rescan.

4 Tool-I/O-Integration

Dieses Kapitel behandelt den Tool-I/O-Betrieb: Montage und Stromversorgung des Geräts am Werkzeugflansch, die Zuordnung der angeschlossenen Geräte zu Ports, die auf der Tool-I/O-Seite angezeigten Schnittstellenzustände sowie Rescan und Restart Interface.

Montage am Roboterflansch

Montieren Sie das Gerät gemäß der eigenen Betriebsanleitung des Geräts am Roboterflansch.

Gerät anschließen und trennen

Gehen Sie wie folgt vor, um ein Gerät am Werkzeugflansch anzuschließen oder zu trennen.

Gerät anschließen

1. Das Gerät am Werkzeugflansch montieren und das Tool-I/O-Kabel anschließen.
2. Auf der Setup-Seite im GRIPLINK-Anwendungs-Tab **Interface** auf **Tool I/O** stellen. Das Plugin versorgt den Flansch automatisch mit 24 V und aktiviert die Tool-I/O-Kommunikation.
3. **Anwendung-Tab → Communication → Robot → Tool I/O** öffnen und **Controlled By** von **User** auf **GRIPLINK** umstellen.
4. **Rescan** drücken; neu angeschlossene Geräte werden ausschließlich über einen Rescan erkannt.
5. Prüfen, dass das Gerät mit Seriennummer und Firmware-Version in der Gerätetabelle erscheint.

Gerät trennen

1. Das Roboterprogramm stoppen. Hält das Gerät ein Teil, dieses zuerst freigeben.
2. **Anwendung-Tab → Communication → Robot → Tool I/O** öffnen und **Controlled By** von **GRIPLINK** auf **User** umstellen.
3. Den Roboter spannungsfrei schalten, sodass die Tool-I/O-Schnittstelle 0 V erreicht (siehe die Warnhinweise unten).
4. Das Kabel trennen und das Gerät vom Flansch entfernen.

Stromversorgung



Die Stromversorgung muss für die angeschlossenen Geräte ausreichend dimensioniert sein. Siehe die Betriebsanleitung des Geräts für die Anforderungen an die Stromversorgung.



Führen Sie Installations-, Demontage- und Verdrahtungsarbeiten nur im spannungsfreien Zustand durch. Stellen Sie vor dem Anschließen, Trennen oder

Austauschen eines Geräts sicher, dass die Tool-I/O-Schnittstelle bei 0 V liegt.

Ports und Gerätezuordnung

Im Tool-I/O-Betrieb kommuniziert das Gerät über die Tool-I/O-Schnittstelle des Roboters, angeschlossen am Tool Connector des Werkzeugflansches. Das Plugin ordnet jedes angeschlossene Gerät einem logischen Port zu; es gelten dieselben Portnummern wie in Programmknoten und Sidebar. Ist **Setup → Interface** auf Tool I/O gestellt, konfiguriert das Plugin die Tool-I/O-Schnittstelle (24-V-Versorgung und Kommunikationseinstellungen) automatisch.

Beispiel: Ist am Tool Connector nur ein Greifer angeschlossen, belegt er **Port 0**.

Jedes Gerät belegt einen Block von vier Ports: Der Greifer selbst ist der erste Port seines Blocks, seine Finger-Module (z. B. bei einem modularen INTRAPAL-Aufbau) folgen auf den nächsten Ports desselben Blocks. Insgesamt sind bis zu 4 Greifer und 16 Ports möglich (dieses 16-Port-Schema gilt spezifisch für die Tool-I/O-Schnittstelle):

Port	Gerät
0	Greifer 1
1-3	Finger-Module von Greifer 1
4	Greifer 2
5-7	Finger-Module von Greifer 2
8	Greifer 3
9-11	Finger-Module von Greifer 3
12	Greifer 4
13-15	Finger-Module von Greifer 4

Rescan zeigt, welche Ports tatsächlich belegt sind; Programmknoten und Sidebar verwenden genau diese Portnummern.



In welchem Block ein Greifer erscheint, bestimmt die am Gerät konfigurierte Stationsadresse (Adresse 1 → Ports 0-3, Adresse 2 → Ports 4-7, ...). Mit der Werkseinstellung 1 ist ein einzelner Greifer immer Port 0. Zum Ändern der Stationsadresse siehe die Betriebsanleitung des Geräts.



Die zulässige Stromaufnahme der Tool-I/O-Schnittstelle des Roboters darf durch angeschlossene Geräte nicht überschritten werden; dies kann zu Störungen oder Schäden führen.



Betreiben Sie die Tool-I/O-Schnittstelle nur mit dem Original-Kabel des jeweiligen Geräts. Andere Kabel oder Verdrahtungen können die Datenverbindung

beeinträchtigen und Schäden verursachen.



Während des Betriebs darf ausschließlich das GRIPLINK-Plugin auf die Tool-I/O-Schnittstelle zugreifen. Andere URCapXs mit Zugriff auf die Tool-I/O-Schnittstelle können zu Konflikten, Kommunikationsfehlern oder unzuverlässigem Betrieb führen.



Die gerätespezifische Parametrierung des GRIPKIT EASY kann bei Bedarf mit der Anwendung „GRIPKIT Configurator“ vorgenommen werden (siehe <<https://weiss-robotics.com/de/gripkit/gripkit-easy/>>).

Schnittstellenzustände

Das Feld **Tool I/O Interface** auf der Tool-I/O-Seite meldet den Zustand der Schnittstelle:

Zustand	Bedeutung
Ready	Betriebsbereit, mindestens ein Gerät verfügbar.
Waiting for Tool I/O access	PolyScope X hat dem Plugin die Tool-I/O-Kontrolle noch nicht erteilt.
Scanning devices...	Die Tool-I/O-Schnittstelle wird nach Geräten durchsucht (typischerweise 2-5 Sekunden).
Waiting for device	Zugriff erteilt, kein Gerät erkannt; Gerät anschließen, mit Spannung versorgen, dann Rescan drücken.
Tool I/O interface error	Die Schnittstelle hat einen Fehler gemeldet; Restart Interface zur Wiederherstellung verwenden.
Starting...	Die Schnittstelle startet.
Interface offline	Die Schnittstelle ist vorübergehend nicht verfügbar.

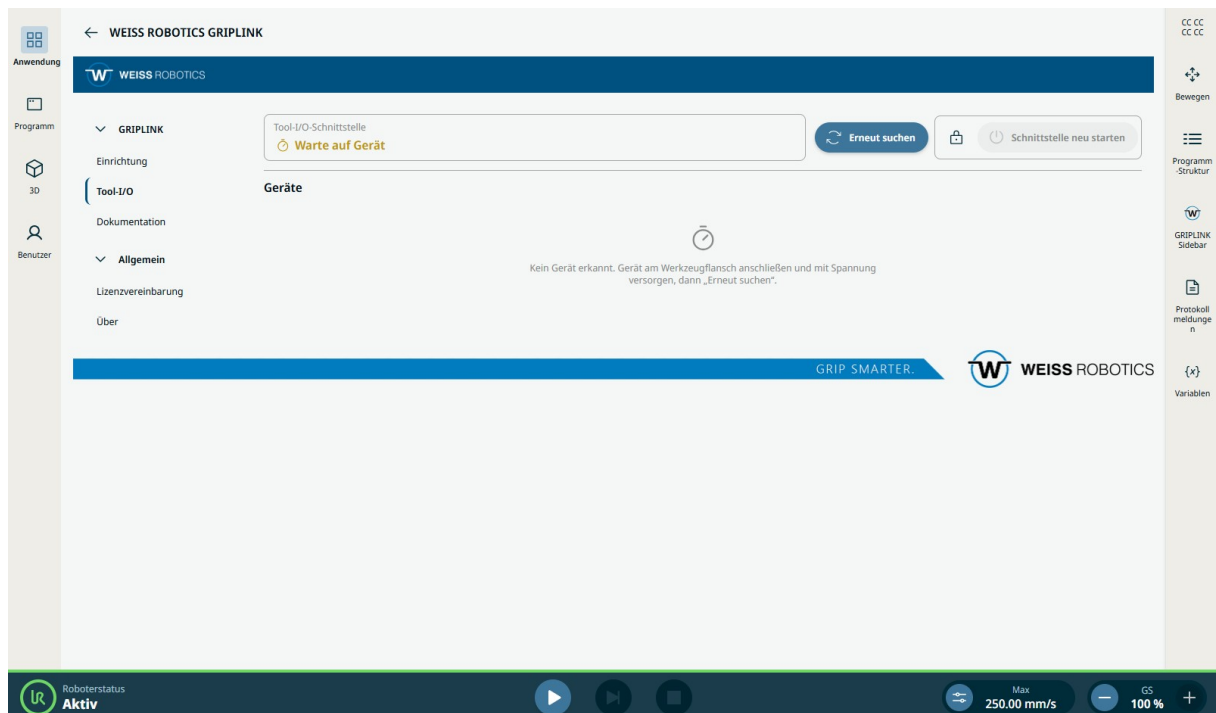


Abbildung 9: Tool-I/O-Seite - Schnittstellenzustand, Rescan und Gerätetabelle.

Rescan und Restart

Aktion	Beschreibung
Rescan	Durchsucht die Tool-I/O-Schnittstelle nach Geräten und aktualisiert die Gerätetabelle. Nach dem Anschließen oder Einschalten eines Geräts drücken. Nicht verfügbar, solange ein Roboterprogramm die Schnittstelle nutzt.
Restart Interface	Stoppt und startet die Tool-I/O-Schnittstelle neu. Zeigt einen Bestätigungsdialog; normalerweise gesperrt, muss vor der Verwendung entsperrt werden. Verwenden, wenn die Schnittstelle in einem Fehlerzustand verbleibt.

Gerätetabelle

Für jeden Port listet die Gerätetabelle **Device**, **Serial No.** und **Firmware**.



Trennen Sie die Verbindung nicht, während ein Roboterprogramm läuft: Verletzungs- und Beschädigungsgefahr durch herabfallende Teile.



Während eine Verbindung auf der Tool-I/O-Schnittstelle aktiv ist, steht **Rescan** nicht zur Verfügung; **Restart Interface** bleibt zur Wiederherstellung verfügbar.

Tool-I/O-spezifische Statuscodes

Diese Statuscodes beschreiben den Zustand der Tool-I/O-Schnittstelle. Sie sind von den allgemeinen GRIPLINK-Protokoll-Statuscodes im Anhang Statuscodes zu unterscheiden.

Statuscode	Name	Beschreibung
100	TOOL_IO_NOT_GRANTED	Der Tool-I/O-Zugriff wurde von PolyScope X nicht erteilt.
101	TOOL_IO_REVOKED	Der Tool-I/O-Zugriff wurde von PolyScope X entzogen.
102	TOOL_IO_CONNECTING	Ein Gerätescan an der Tool-I/O-Schnittstelle läuft.
103	DAEMON_BOOTING	Die Tool-I/O-Schnittstelle startet.
104	DAEMON_SHUTDOWN	Die Tool-I/O-Schnittstelle wird beendet.

5 Manuelle Bedienung über Sidebar und Smart Skills

Für die manuelle Bedienung ohne Roboterprogramm stellt das Plugin zwei Oberflächen bereit: die **GRIPLINK Sidebar** für Live-Steuerung und Diagnose sowie **Smart Skills** für vorkonfigurierte Aktionen, die mit einem Fingertipp ausgelöst werden. Beide sprechen das Gerät direkt an; keine von beiden erzeugt URScript oder verändert das Roboterprogramm.

5.1 Sidebar

Die **GRIPLINK Sidebar** bietet eine komfortable Bedienoberfläche für die im aktuellen Setup erkannten angeschlossenen Geräte. Sie wird über die Seitenleiste von PolyScope X geöffnet und unterstützt insbesondere folgende Aufgaben:

- Inbetriebnahme
- Diagnose
- Greifen und Freigeben beim Einrichten und Testen
- Einsehen von Presets und deren Parametern
- Beobachten wichtiger Gerätedaten

Die Sidebar zeigt die Daten an und stellt die unterstützten Funktionen für den aktuell ausgewählten Port bereit. Bedienelemente werden je nach Gerätezustand automatisch aktiviert oder deaktiviert.

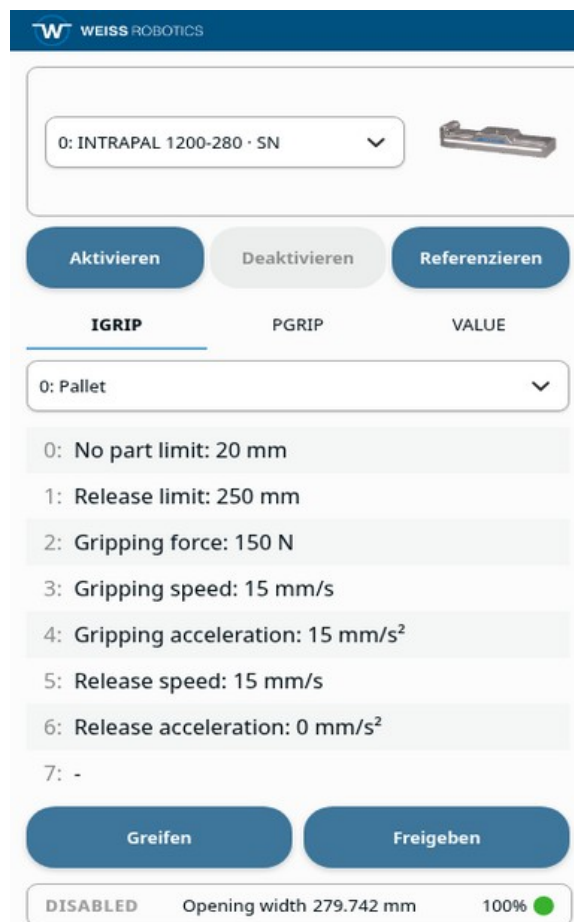


Abbildung 10: GRIPLINK Sidebar (IGRIP-Tab).

Aufbau der Sidebar

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

Bedienelement	Beschreibung
Geräteauswahl (Dropdown-Menü)	Dropdown zur Auswahl des gewünschten Geräts, z. B. 0: INTRAPAL 1200-280 · SN... Angezeigt werden Portnummer, Gerätetyp und Seriennummer, zusammen mit einem Bild des angeschlossenen Geräts.
Aktivieren / Deaktivieren (Buttons)	Aktiviert bzw. deaktiviert das ausgewählte Gerät. Der resultierende Zustand wird in der Statuszeile angezeigt, z. B. DISABLED.
Referenzieren (Button)	Führt für das ausgewählte Gerät eine Referenzfahrt aus.

Die Statuszeile am unteren Rand der Sidebar zeigt zyklisch aktualisierte Daten des ausgewählten Geräts:

Anzeige	Beschreibung
Zustand	Der aktuelle Gerätezustand (z. B. DISABLED, RELEASED, HOLDING, FAULT).
Telemetriedaten	Der primäre Messwert des Geräts, z. B. die Öffnungsweite (in mm) bei Greifmodulen.
Condition (Statusanzeige)	Die Zustandsbewertung („Condition“) des Geräts als Prozentwert mit farbiger Statusanzeige. Grün kennzeichnet ein betriebsbereites Gerät; Orange und Rot kennzeichnen einen fehlerhaften Zustand.

IGRIP-Tab: Arbeiten mit Griff-Presets

IGRIP („Index GRIP“) greift oder gibt mit den für das ausgewählte Gerät gespeicherten Grip-Presets frei. Der Tab zeigt die konfigurierten Presets und deren Parameter (siehe Abbildung 10).

Bedienelement	Beschreibung
---------------	--------------

Preset-Auswahl (Dropdown-Menü)	Dropdown mit den verfügbaren Grip-Presets 0–7. Jede Zeile zeigt den Index und den Tag, z. B. 0: Pallet.
Preset-Parameter-Anzeige	Listet die im ausgewählten Preset gespeicherten Parameter auf. Je nach Gerät werden u. a. angezeigt: No part limit (mm), Release limit (mm), Greifkraft (N), Greif- und Freigabegeschwindigkeit (mm/s) sowie Beschleunigung (mm/s ²).

Greifen oder Freigeben anhand eines Presets

1. Ein Preset von 0 bis 7 auswählen.
2. **Greifen** oder **Freigeben** drücken.

Typische Verwendung

- Testen eines neu konfigurierten Presets.
- Einsehen von Presets und deren eingestellten Parametern.
- Schnelles Einlernen: Preset auswählen, über die Sidebar mehrmals greifen und das Verhalten beobachten.
- Kontrolle, ob ein Teil mit den eingestellten Limits zuverlässig erkannt wird.

PGRIP-Tab: Flexibles Greifen / Freigeben

PGRIP („Parameter GRIP“) ermöglicht ein parametrisiertes, flexibles Greifen und Freigeben mit expliziten Werten statt eines gespeicherten Presets (analog zu den Befehlen FLEXGRIP und FLEXRELEASE).

Bedienelement	Beschreibung
Greifen - Position (Schieberegler)	Zielposition zum Greifen in mm.
Greifen - Kraft (Schieberegler)	Greifkraft in N.
Greifen - Geschwindigkeit (Schieberegler)	Greifgeschwindigkeit; im Ethernet-Betrieb in mm/s, im Tool-I/O-Betrieb als prozentualer Faktor (%) der maximalen Gerätegeschwindigkeit. Mit Automatische Geschwindigkeit wird die Standardgeschwindigkeit des Geräts verwendet.
Freigeben - Position (Schieberegler)	Zielposition zum Freigeben in mm.
Freigeben - Geschwindigkeit (Schieberegler)	Freigabegeschwindigkeit, mit eigener Checkbox Automatische Geschwindigkeit . Das Freigeben verwendet nur Position und Geschwindigkeit.

Greifen oder Freigeben mit expliziten Parametern

1. Position, Kraft und Geschwindigkeit über die Schieberegler einstellen (oder **Automatische Geschwindigkeit** aktivieren).
2. **Greifen** oder **Freigeben** drücken.

Einsatzfälle

- Ermitteln einer passenden Greifposition und Greifkraft für ein neues Teil.
- 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.



Die Tabs **PGRIP** und **VALUE** erfordern ein angeschlossenes und aktiviertes Gerät; ohne Gerät sind beide Tabs deaktiviert.

The screenshot displays the WEISS ROBOTICS GRIPLINK interface. At the top, a dropdown menu shows '0: INTRAPAL 1200-280 · SN' next to a small robot arm icon. Below this are three buttons: 'Aktivieren' (highlighted in blue), 'Deaktivieren' (greyed out), and 'Referenzieren' (blue). Underneath these buttons are three tabs: 'IGRIP', 'PGRIP' (which is selected and underlined), and 'VALUE'. The main content area is divided into two sections: 'Greifen' (Gripping) and 'Freigeben' (Releasing). The 'Greifen' section includes a 'Position: 10.00 mm' label with a slider, a 'Kraft: 300 N' label with a slider, a 'Geschwindigkeit:' label with a slider, and a checked checkbox for 'Automatische Geschwindigkeit'. The 'Freigeben' section includes a 'Position: 272.00 mm' label with a slider, a 'Geschwindigkeit:' label with a slider, and a checked checkbox for 'Automatische Geschwindigkeit'. At the bottom of the main area are two large blue buttons: 'Greifen' and 'Freigeben'. Below these buttons is a status bar showing 'DISABLED', 'Opening width 279.742 mm', and '100%' with a green circle.

Abbildung 11: GRIPLINK Sidebar (PGRIP-Tab) - flexible Greif- und Freigabeparameter.

VALUE-Tab: Gerätedaten beobachten

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

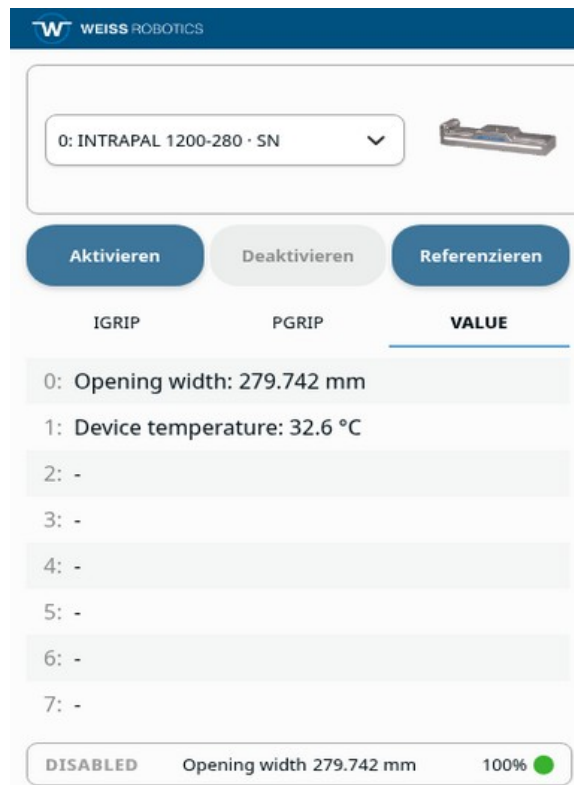


Abbildung 12: GRIPLINK Sidebar (VALUE-Tab) - Live-Gerätetelemetrie (Öffnungsweite, Temperatur).

Beispiel für die INTRAPAL-Serie:

Index	Wert
0	Öffnungsweite (mm)
1	Gerätetemperatur (°C)
2-7	Nicht verwendet bei der INTRAPAL-Serie

Typische Verwendung

- Kontrolle, ob ein Teil korrekt erkannt wurde.
- Überwachen der Öffnungsweite beim Einrichten und Testen.
- Überprüfen der Gerätetemperatur bei langen Zyklen.
- Unterstützung bei der Wahl sinnvoller Grenzwerte (No part limit, Release limit usw.).

Hinweise zur Verwendung der Sidebar



Sidebar- und Smart-Skill-Aktionen schließen sich mit einer offenen URScript-Verbindung (GL_CONNECT) gegenseitig aus, selbst wenn das Programm nur pausiert ist. Sidebar- und Smart-Skill-Aktionen stehen wieder zur Verfügung, sobald die Verbindung geschlossen ist (Programm beendet oder DISCONNECT ausgeführt).



Bei mehrfachen schnellen Tastendrücken werden die Schaltflächen vorübergehend deaktiviert und automatisch wieder aktiviert, um Kommunikations- und

Robotersteuerungs-Überlastung zu vermeiden.



Die Sidebar verändert keine bestehenden Programmknoten; verwenden Sie sie nur für manuelle Bedienung, Diagnose und Inbetriebnahme. Dauerhafte Prozessänderungen im Roboterprogramm vornehmen.

5.2 Smart Skills

Smart Skills sind vorkonfigurierte manuelle Aktionen, die mit einem Fingertipp ausgelöst werden. Ein Skill wird einmal unter **Anwendung** → **Smart Skills** konfiguriert; der Bediener löst ihn anschließend im Tab Smart Skills der **Bewegen**-Ansicht aus, ohne ein Programm oder die Sidebar zu öffnen. Ein Skill spricht das Gerät direkt an, über denselben Live-Kanal wie die Sidebar; er erzeugt kein URScript und verändert das Roboterprogramm nicht.

GRIPLINK verfügt über drei Smart Skills:

Icon	Titel	Beschreibung
	GRIPLINK Grip	Greifen mit dem konfigurierten Preset oder den Parametern am gewählten Port.
	GRIPLINK Release	Freigeben mit dem konfigurierten Preset oder den Parametern am gewählten Port.
	GRIPLINK Home	Greifer am gewählten Port referenzieren (Home).

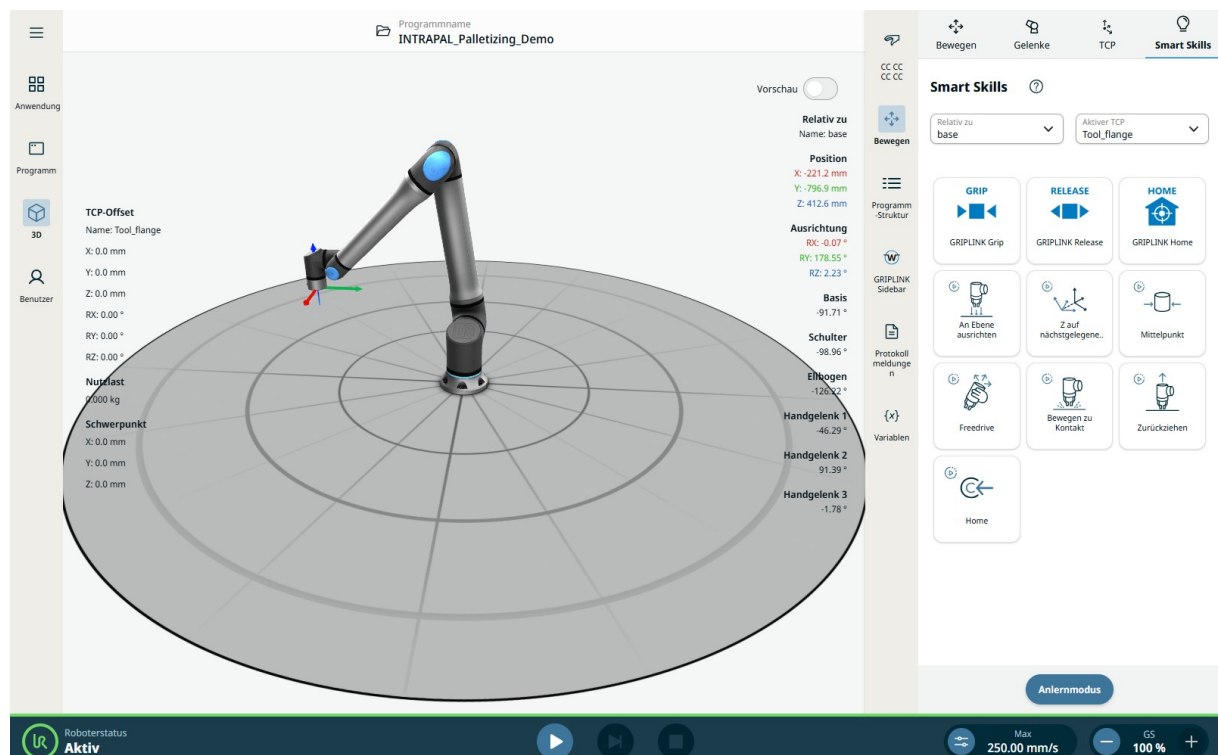


Abbildung 13: Der Tab Smart Skills in der Bewegen-Ansicht - GRIPLINKs drei Skills neben den eingebauten Skills von PolyScope X.



Verwenden Sie Smart Skills wie die Sidebar für manuelle Bedienung und Inbetriebnahme; sie verändern das Roboterprogramm nicht. Soll eine Greifer-Aktion Teil des Programms werden, verwenden Sie stattdessen einen GRIPLINK-Programmknoden.

Skill-Instanzen verwalten

Konfigurieren Sie jeden Smart Skill unter **Anwendung → Smart Skills**. Die Liste links trennt **aktivierte** Skills (im Tab Smart Skills der Bewegen-Ansicht sichtbar) von **deaktivierten** (konfiguriert, aber ausgeblendet). **Smart Skill erstellen** dupliziert einen vorhandenen Skill oder beginnt einen neuen von Grund auf.

Aktion	Beschreibung
Umbenennen	Den Anzeigenamen der Instanz ändern.
Duplizieren	Eine zweite, unabhängig konfigurierte Instanz desselben Skills anlegen, z. B. eine „Grip“-Instanz für Port 0 und eine separate für Port 1.
Deaktivieren / Löschen	Aus der Bewegen-Ansicht entfernen. GRIPLINKs drei mitgelieferte Skills bleiben im Katalog erhalten und werden deaktiviert; eine duplizierte Instanz kann zusätzlich gelöscht werden.

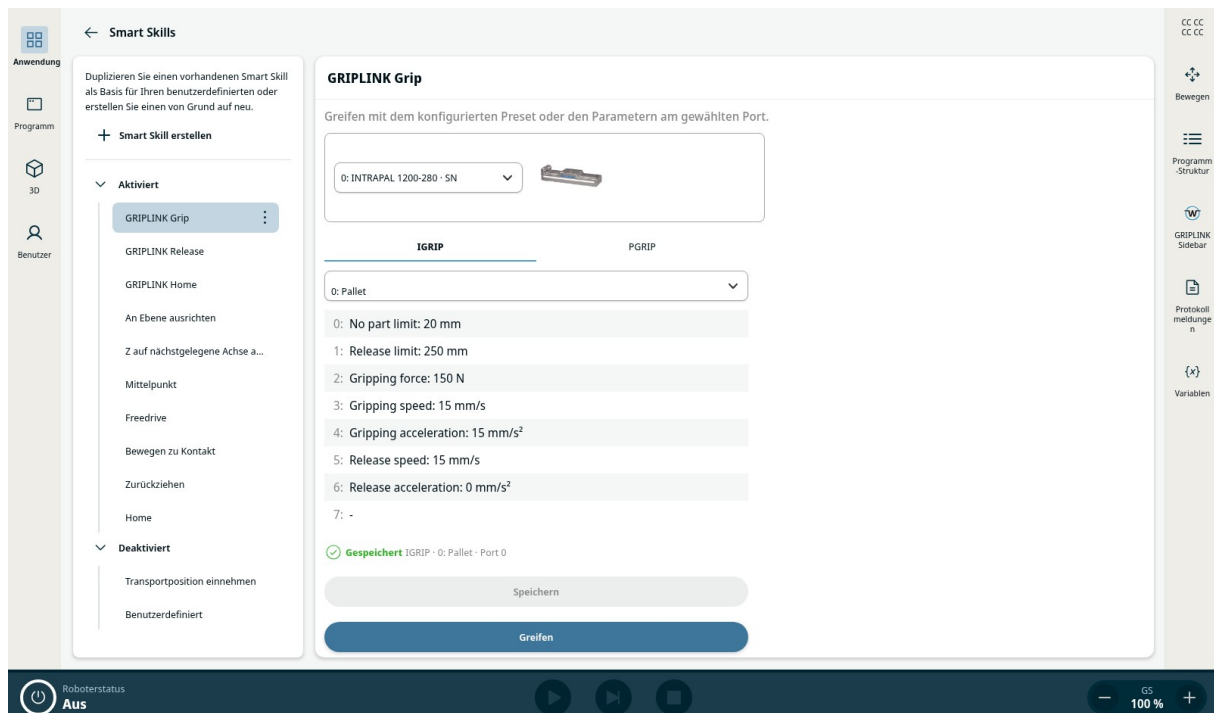


Abbildung 14: Konfiguration des GRIPLINK-Grip-Skills, geöffnet aus dem Smart-Skills-Katalog.

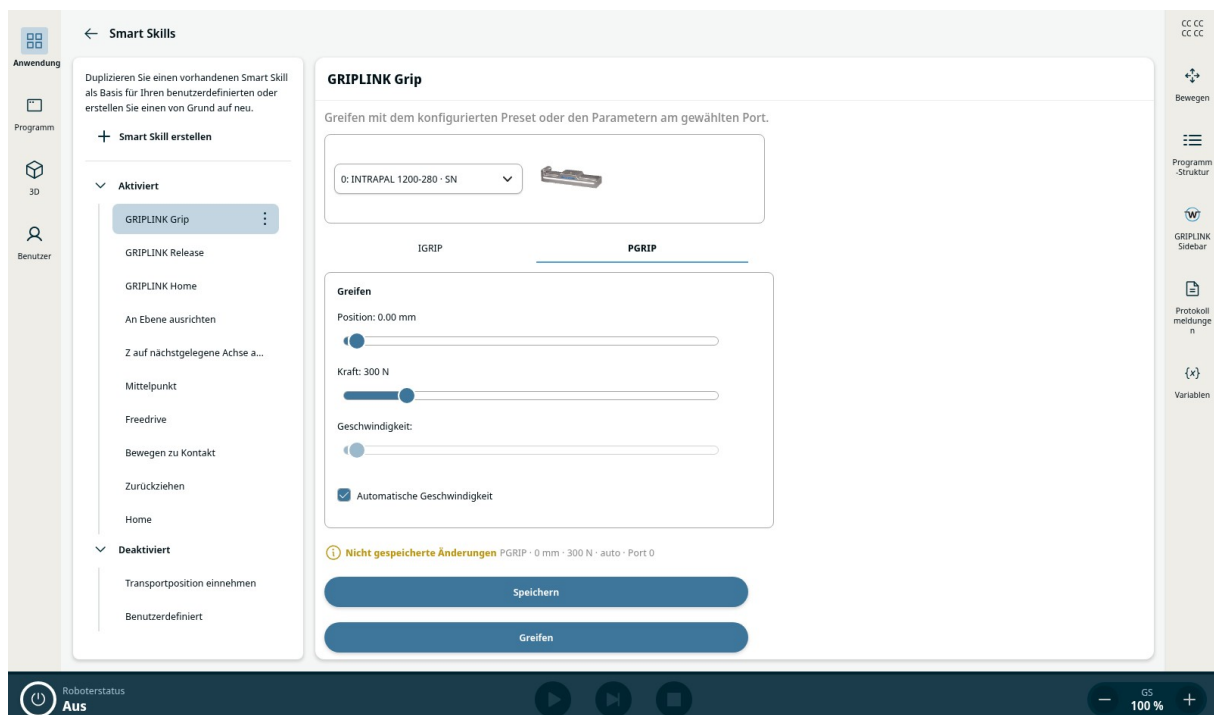


Abbildung 15: GRIPLINK-Grip-Skill (PGRIP-Tab) - Greifen mit expliziter Position, Kraft und Geschwindigkeit.

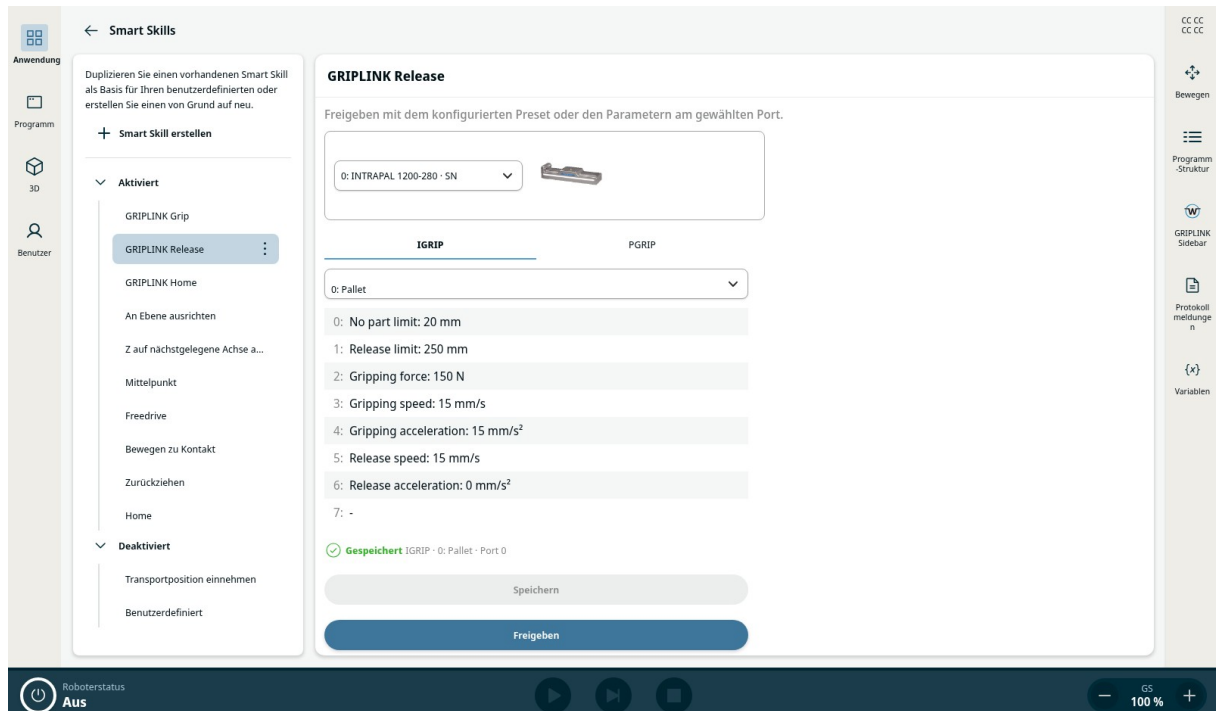


Abbildung 16: GRIPLINK-Release-Skill (IGRIP-Tab) - Freigeben per gespeichertem Preset.

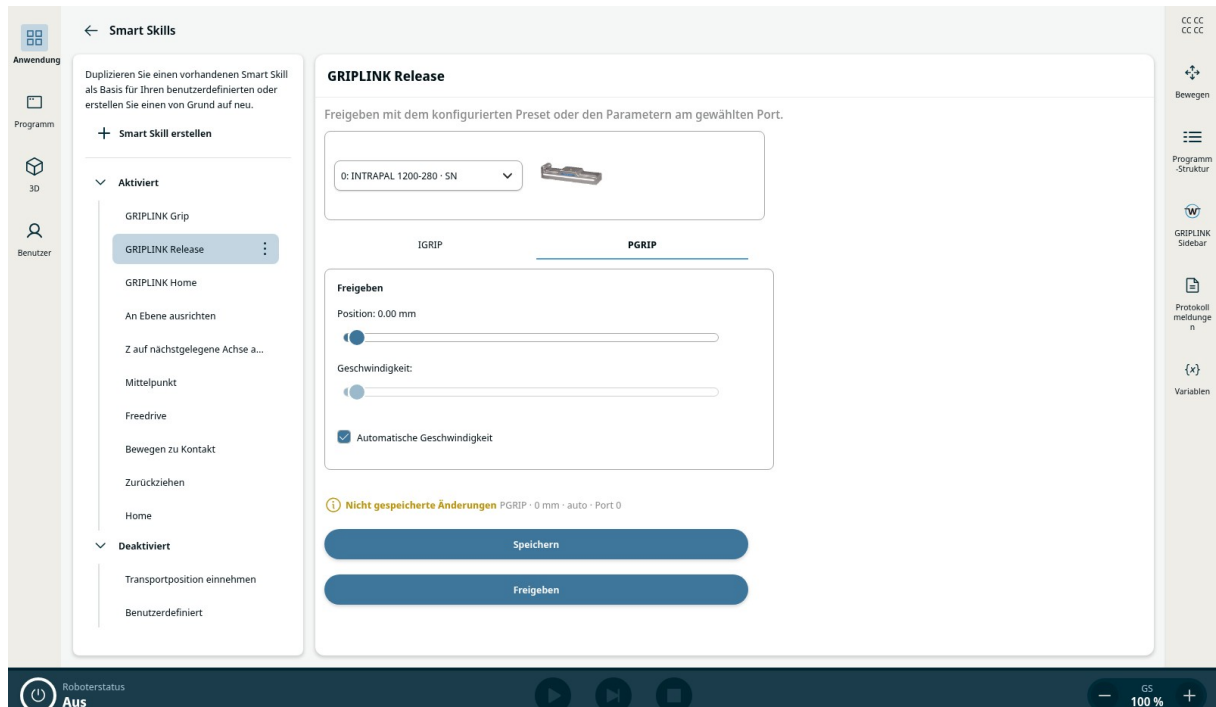


Abbildung 17: GRIPLINK-Release-Skill (PGRIP-Tab) - Freigeben mit expliziter Position und Geschwindigkeit.

Einen Skill konfigurieren: Port, Modus und Preset

Beim Öffnen eines Skills erscheinen dieselben Bedienelemente wie im entsprechenden Sidebar-Tab, bezogen auf diese eine Instanz:

Bedienelement	Beschreibung
Port	Der Controller-Port, auf den diese Skill-Instanz wirkt, mit Live-Bild des angeschlossenen Geräts.
IGRIP / PGRIP (nur Grip und Release)	Ein gespeichertes Preset wählen (dieselbe Acht-Slot-Tabelle wie im IGRIP-Tab der Sidebar) oder Position, Kraft und Geschwindigkeit direkt einstellen (dieselben Schieberegler und die Checkbox Automatische Geschwindigkeit wie im PGRIP-Tab der Sidebar, einschließlich der modusabhängigen Geschwindigkeitseinheit).

GRIPLINK Home benötigt nur den Port.

Jede Skill-Instanz behält ihren eigenen Port, ihren Modus und ihre Werte, unabhängig von der Sidebar und jeder anderen Instanz.

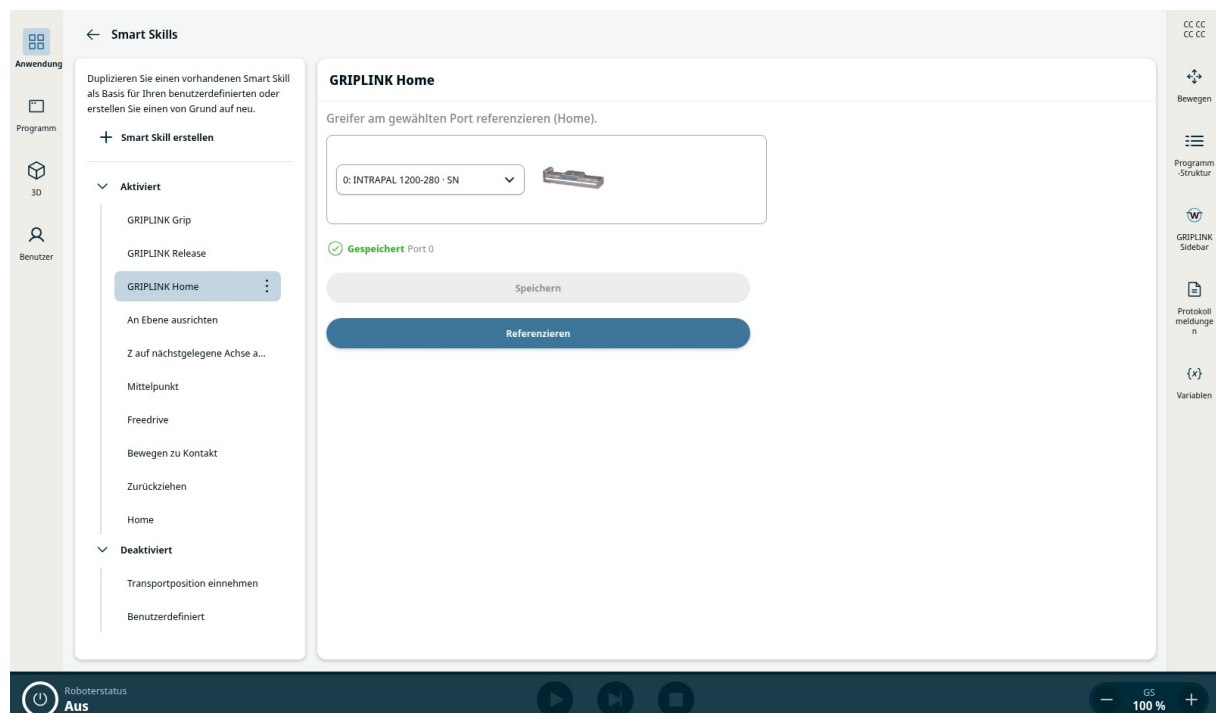


Abbildung 18: Konfiguration des GRIPLINK-Home-Skills - der einfachste Skill, nur der Port.

Entwurf, Speichern und die Anzeige nicht gespeicherter Änderungen

Das Ändern von Port, Modus, Preset oder eines Schiebereglerwerts bearbeitet zunächst einen **Entwurf**; wirksam wird er erst mit dem Speichern. Eine Statuszeile zeigt den Zustand des Panels:

Status	Bedeutung
Nicht gespeicherte Änderungen	Der Entwurf enthält nicht gespeicherte Änderungen. Eine Zusammenfassung zeigt jederzeit den aktuellen Entwurf (z. B. „IGRIP · 0: Pallet · Port 0“ oder „PGRIP · 40 mm · 25 N · auto · Port 0“), sodass die auszuführende Aktion auf einen Blick sichtbar ist.
Gespeichert	Der Entwurf entspricht dem gespeicherten Stand und bleibt beim Verlassen der Ansicht erhalten.

Speichern übernimmt den Entwurf und ist nur aktiv, solange nicht gespeicherte Änderungen vorliegen.

Verfügbarkeit und Validierung

Die Aktionsschaltfläche eines Skills ist nur aktiv, wenn dieselbe Sidebar-Schaltfläche im aktuellen Gerätezustand ebenfalls aktiv wäre, und nur, wenn kein anderer Steuerkanal (etwa eine offene Verbindung durch ein Roboterprogramm) das Gerät gerade steuert.

6 Programmieren mit Programmknoten

6.1 Wie die Befehlsreferenz zu lesen ist

Grundsequenz für Roboterprogramme

Verwenden Sie diese Reihenfolge als Standardmuster; überspringen Sie optionale Schritte nur, wenn Gerätetyp und Anwendung dies erlauben.

Schritt	Befehl / Aktion	Zweck	Entscheidung der Anwendung
1	CONNECT / GL_CONNECT	Kommunikation zum GRIPLINK Controller oder zur Tool-I/O- Schnittstelle öffnen.	Einmal vor Gerätebefehlen ausführen.
2	ASSERTDEVICE / GL_DEVASSERT	Erwartetes Gerät am ausgewählten Port prüfen.	Beim Start oder nach einem Werkzeugwechsel verwenden.
3	HOME / GL_HOME	Servogreifer referenzieren, die dies benötigen.	Nur überspringen, wenn keine Referenzierung nötig oder bereits erfolgt ist.
4	ENABLE / GL_ENABLE	Gerät für den Betrieb aktivieren.	Vor Greifen, Freigeben oder parametrisierten Bewegungen ausführen.
5	GRIP / RELEASE / FLEXGRIP / FLEXRELEASE	Prozessaktion ausführen.	Preset-Befehle für einfache Zyklen; FLEX- Befehle mit Position, Kraft, Geschwindigkeit.
6	DEVICESTATE / WSTR / VALUE / WAITVAL	Ergebnis auswerten oder Zustände/Werte überwachen.	Auf HOLDING, NO PART, RELEASED, FAULT oder Timeout verzweigen.
7	DISABLE / DISCONNECT	Gerät deaktivieren oder Verbindung trennen.	Werkzeugwechsel.

Ebene	Verwenden, wenn...	Dokumentiert in
Programmknoten	Die Applikation visuell im Programmbaum aufgebaut werden soll.	Dieses Kapitel.
Pick-/Place-Knoten	Das Programm automatisch auf HOLDING oder NO PART verzweigen soll.	Pick / Place, dieses Kapitel.
URScript-Funktionen	Das Programm Logik auf Skript-Ebene, Schleifen, Variablen oder generierten Code benötigt.	Kapitel URScripting.
Anhänge	Das Programm Zustandswerte oder Diagnosen auswertet.	Gerätezustand und Statuscodes.

Beispiel: Minimalprogramm

Das kürzeste vollständige Greiferprogramm - verbinden, das Gerät vorbereiten, dann greifen und freigeben:

```
# Ethernet: Controller-IP-Adresse; Tool I/O: 127.0.0.1
GL_CONNECT("192.168.1.40", "sock_griplink")
GL_HOME(0, "sock_griplink")
GL_ENABLE(0, True, "sock_griplink")
GL_GRIP(0, 0, 1, "sock_griplink")
# ... Roboterbewegung ...
GL_RELEASE(0, 0, 1, "sock_griplink")
```

Dieselbe Sequenz gilt für Programmknoten: CONNECT, HOME und ENABLE im Initialisierungsabschnitt (z. B. BeforeStart), GRIP und RELEASE im Hauptprogramm.

6.2 Programmknoten-Befehlsreferenz

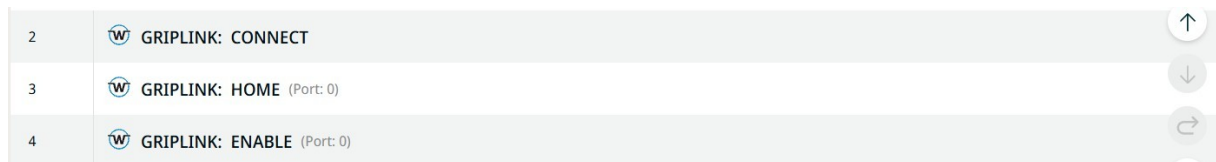
GRIPLINK-Befehle werden über die GRIPLINK-Programmknoten in den Programmbaum eingefügt. Der zentrale Knoten ist der vereinheitlichte **GRIPLINK**-Knoten, der jeden Befehl über das Auswahlfeld **Aktion** bereitstellt. Die strukturierten Knoten **GRIPLINK Pick** und **GRIPLINK Place** (siehe Pick / Place in diesem Kapitel) bauen auf denselben GRIP-/RELEASE-Aktionen auf und ergänzen automatische Ergebniszweige.

Jeder Knoten bietet einen **Mehr**-Dialog mit einer Befehlsbeschreibung und einer URScript-Vorschau des erzeugten Aufrufs. Das Kapitel URScripting dokumentiert die zugrundeliegende GL_*-Funktion jeder Aktion mit genauem Parametertyp und Wertebereich.

Grundlegender Programmablauf

Initialisierungsabschnitt einrichten (z. B. BeforeStart)

1. Einen **GRIPLINK CONNECT**-Knoten hinzufügen, um die Verbindung zum Controller zu öffnen.
2. Einen **GRIPLINK HOME**-Knoten für jedes Gerät hinzufügen, das referenziert werden muss.
3. Einen **GRIPLINK ENABLE**-Knoten für jedes Gerät hinzufügen, das dies benötigt.



2	 GRIPLINK: CONNECT	↑
3	 GRIPLINK: HOME (Port: 0)	↓
4	 GRIPLINK: ENABLE (Port: 0)	↺

Abbildung 19: GRIPLINK-Programmknoten im Initialisierungsabschnitt (BeforeStart).

Verwenden Sie im Hauptprogramm **GRIPLINK GRIP / RELEASE** (oder die Knoten **GRIPLINK Pick / Place**), um Teile zu handhaben, und werten Sie das Ergebnis zur Verzweigung aus.



CONNECT, ENABLE und HOME im Initialisierungsabschnitt stellen sicher, dass jeder Programmlauf von einem definierten Gerätezustand ausgeht.

Das Auswahlfeld „Aktion“

Fügen Sie einen **GRIPLINK**-Knoten ein und wählen Sie den Befehl im Auswahlfeld **Aktion**:

Aktion (UI-Bezeichnung)	Funktion
GRIPLINK CONNECT	Verbindung zum Controller öffnen.
GRIPLINK DISCONNECT	Verbindung trennen.
GRIPLINK ASSERTDEVICE	Gerätetyp an einem Port prüfen (Vendor / Product ID).
GRIPLINK HOME	Gerät referenzieren.
GRIPLINK ENABLE	Gerät aktivieren (optional auf Gerätezustandsübergang warten).
GRIPLINK DISABLE	Gerät deaktivieren (optional warten).
GRIPLINK GRIP	Werkstück über Griff-Preset greifen.
GRIPLINK RELEASE	Werkstück über Griff-Preset freigeben.
GRIPLINK FLEXGRIP	Werkstück über Parameter greifen (Position, Kraft, Geschwindigkeit, Beschleunigung).

GRIPLINK FLEXRELEASE	Werkstück über Parameter freigeben (Position, Geschwindigkeit, Beschleunigung).
GRIPLINK CLAMP	Greifkrafterhaltung ein-/ausschalten.
GRIPLINK LED	Anzeige-LED auf ein Preset setzen.
GRIPLINK DEVICESTATE	Aktuellen Gerätezustand abfragen (in eine Variable).
GRIPLINK VALUE	Gerätewert abfragen (in eine Variable).
GRIPLINK WSTR	Auf eine Änderung des Gerätezustands warten.
GRIPLINK WAITVAL	Warten, bis ein Gerätewert ein Ziel erreicht.
GRIPLINK SETVALUE	Gerätewert setzen.
GRIPLINK SGRIPCFG	Griff-Preset parametrieren.

Übliche Parameterbereiche: **Port** 0–3 (Standalone) bzw. 0–31 (Proxy); **Preset** 0–7.

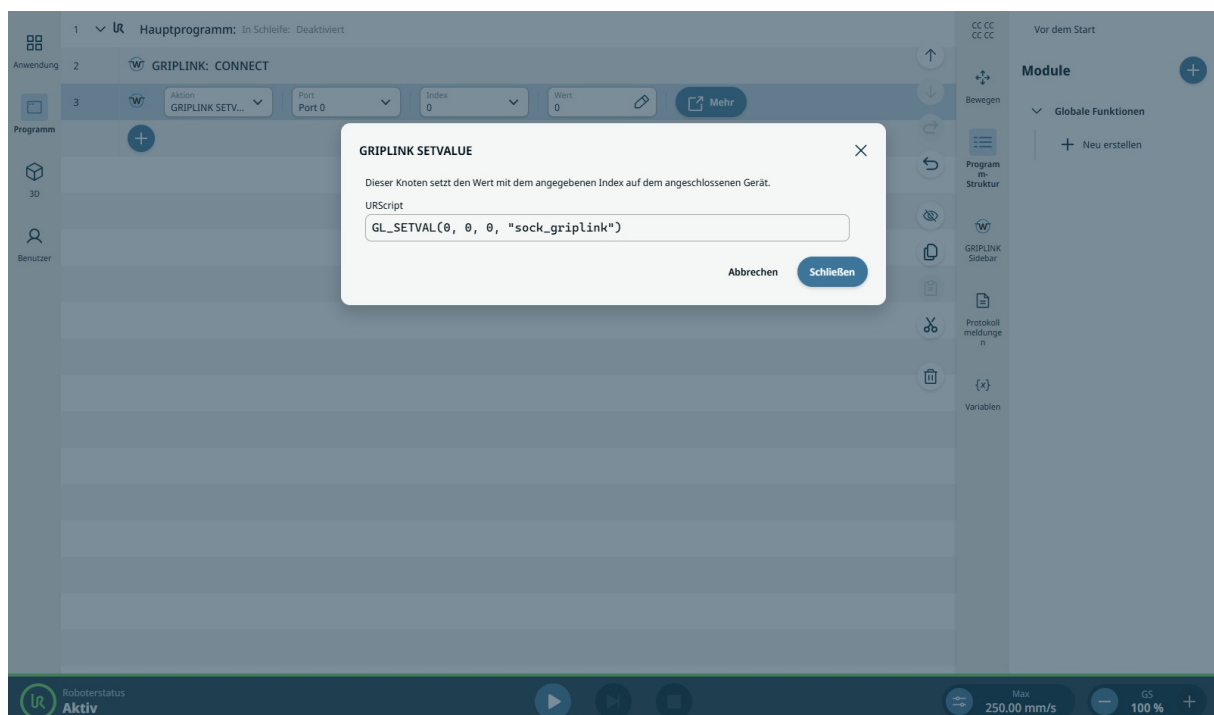


Abbildung 20: Der Mehr-Dialog (hier für GRIPLINK SETVALUE) - Beschreibung und eine URScript-Vorschau des erzeugten Aufrufs.

6.3 Verbindung aufbauen – CONNECT

Der Befehl CONNECT stellt die Verbindung zum konfigurierten GRIPLINK Controller her. IP-Adresse und Socket-Name stammen aus dem Applikationsknoten; der Knoten hat keine weiteren Felder.



Abbildung 21: GRIPLINK CONNECT - keine weiteren Felder außer dem Auswahlfeld Aktion.

6.4 Verbindung trennen – DISCONNECT

Der Befehl DISCONNECT trennt die von CONNECT aufgebaute Verbindung, etwa wenn ein Programm den Controller vor dem Programmende oder vor einem erneuten Verbindungsaufbau freigeben muss. Der Knoten hat keine weiteren Felder.



Abbildung 22: GRIPLINK DISCONNECT - keine weiteren Felder außer dem Auswahlfeld Aktion.

6.5 Ein angeschlossenes Gerät prüfen – ASSERTDEVICE

Der Befehl ASSERTDEVICE prüft, ob das Gerät am ausgewählten Port den erwarteten Werten **Hersteller-ID** und **Produkt-ID** entspricht. Beide IDs stammen aus der IODD-Datei des Geräts (IO-Link Device Description) und sind editierbar; **Übernehmen** liest sie direkt vom angeschlossenen Gerät, statt sie manuell einzutragen.

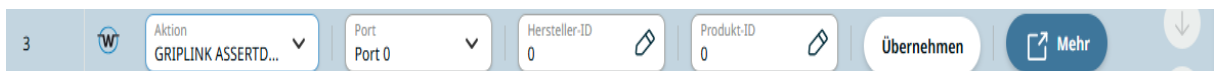


Abbildung 23: GRIPLINK ASSERTDEVICE - Port, Vendor ID, Product ID und Apply.



Siehe die Betriebsanleitung des jeweiligen Geräts für Vendor- und Product-IDs.

6.6 Gerät referenzieren – HOME

Der Befehl HOME referenziert das Gerät am ausgewählten Port. **Port** ist das einzige Feld.



Abbildung 24: GRIPLINK HOME - Port ist das einzige Feld.

6.7 Gerät aktivieren – ENABLE

Der Befehl ENABLE aktiviert das Gerät am ausgewählten Port. Mit aktivierter Option **Auf Zustandsübergang warten** wartet der Knoten, bis das Gerät den Zustand ENABLED meldet.



Abbildung 25: GRIPLINK ENABLE - Port und der Schalter Auf Zustandsübergang warten.



Manche Geräte müssen aktiviert werden, bevor sie verwendet werden können.

6.8 Gerät deaktivieren – DISABLE

Der Befehl DISABLE deaktiviert das Gerät am ausgewählten Port und versetzt es in einen sicheren, spannungsfreien Zustand. Der Knoten hat dieselben Felder wie ENABLE: **Port** und **Auf Zustandsübergang warten**.



Abbildung 26: GRIPLINK DISABLE - Port und der Schalter Auf Zustandsübergang warten.



Ist das Gerät bereits DISABLED, wenn dies mit aktivierter Option Auf Zustandsübergang warten ausgeführt wird, kann ein Timeout-Fehler auftreten.



Das Demo-Programm `fault_clearing_via_disable` quittiert einen Gerätefehler mit DISABLE gefolgt von HOME.

6.9 Werkstück greifen – GRIP

Der Befehl GRIP führt einen Griff mit dem gewählten Griff-Preset (**Preset**, 0–7) am ausgewählten Port aus. **Blockierend** ist standardmäßig aktiv, sodass der Knoten wartet, bis der Greifvorgang abgeschlossen ist; deaktiviert kehrt der Knoten sofort zurück.



Abbildung 27: GRIPLINK GRIP - Port, Preset und Blockierend (standardmäßig aktiv).



Siehe die Betriebsanleitung des Greifmoduls für Greifparameter.



Die Demo-Programme `simple_grip_release` und `griplink_demo` zeigen vollständige Greif-/Freigabezyklen mit einem gespeicherten Griff-Preset (siehe Demo-Programme).

6.10 Werkstück freigeben – RELEASE

Der Befehl RELEASE gibt das Werkstück mit dem gewählten Griff-Preset (**Preset**, 0–7) am ausgewählten Port frei. Der Knoten hat dieselben Felder wie GRIP: Port, Preset und Blockierend.



Abbildung 28: GRIPLINK RELEASE - Port, Preset und Blockierend.

6.11 Werkstück über Parameter greifen – FLEXGRIP

Der Befehl FLEXGRIP führt einen Griff mit den Parametern **Position**, **Kraft**, **Geschwindigkeit** und **Beschleunigung** statt eines gespeicherten Griff-Presets aus. Die Parameter werden in einem separaten **Konfigurieren**-Dialog („Flex-Parameter“) eingestellt; die Knotenzeile selbst zeigt nur **Port**, **Konfigurieren** und **Blockierend**.



Abbildung 29: GRIPLINK FLEXGRIP - Port, Konfigurieren und Blockierend.

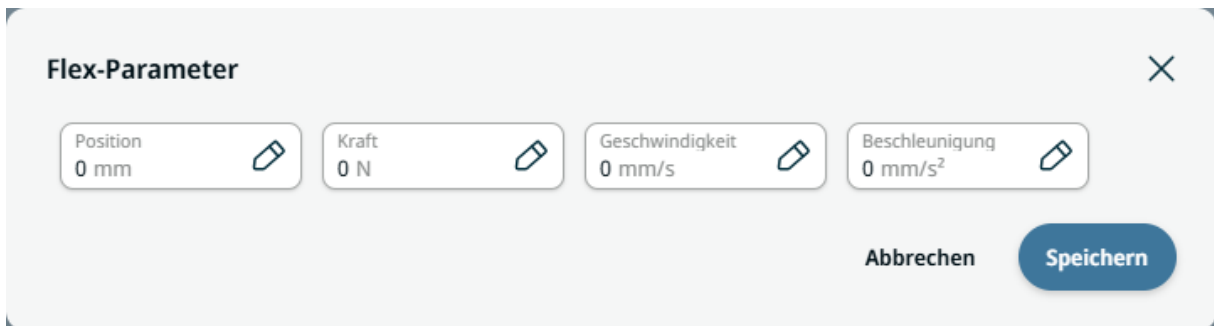


Abbildung 30: Der FLEXGRIP-Konfigurieren-Dialog - Position, Kraft, Geschwindigkeit, Beschleunigung.



Siehe die Betriebsanleitung des Greifmoduls für Bewegungsparameter.



Die Bedeutung von **Geschwindigkeit** und **Beschleunigung** hängt vom Anschlussmodus ab: physikalische mm/s bzw. mm/s² im Ethernet-Betrieb; Override-Prozentwerte im Tool-I/O-Betrieb (Geschwindigkeit 0–200 %, Beschleunigung 0–100 %, 100 % = Nennwert).



Das Demo-Programm flexgrip_cycle zeigt einen Greif-/Freigabezyklus mit expliziten Bewegungsparametern.

6.12 Werkstück über Parameter freigeben – FLEXRELEASE

Der Befehl FLEXRELEASE gibt das Werkstück mit den Parametern **Position**, **Geschwindigkeit** und **Beschleunigung** frei. Im Tool-I/O-Betrieb ist **Geschwindigkeit** ein Override von 10–100 % und **Beschleunigung** ein Override von 0–100 %; im Ethernet-Betrieb sind beide physikalisch (mm/s, mm/s²).

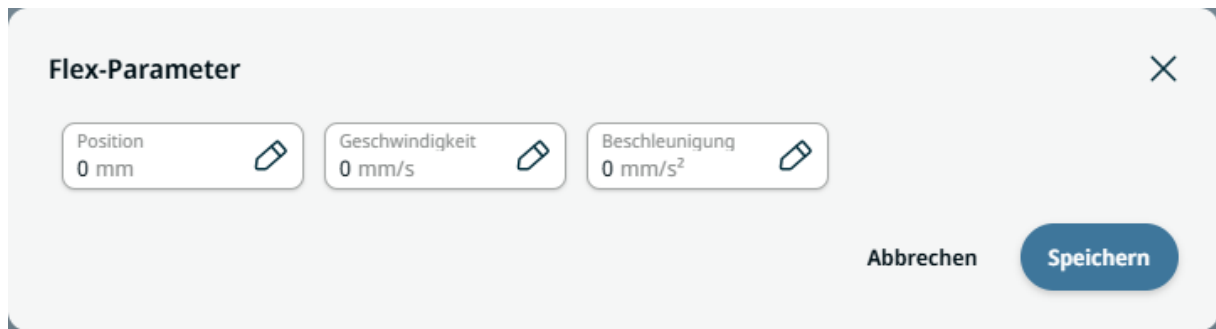


Abbildung 31: Der FLEXRELEASE-Konfigurieren-Dialog - Position, Geschwindigkeit, Beschleunigung.



Das Demo-Programm `async_preposition` nutzt ein nicht blockierendes FLEXRELEASE, um die Finger vorzupositionieren, während der Roboter verfährt.

6.13 Pick / Place

Für die Teilehandhabung stellt GRIPLINK zwei strukturierte Programmknoten bereit, die ein Greifen bzw. Freigeben mit automatischen Ergebniszweigen kombinieren, sodass das Programm keine separate DEVICESTATE-Abfrage und keinen If-Knoten benötigt.

GRIPLINK Pick

Führt ein Greifen aus (GRIP oder FLEXGRIP, je nach Konfiguration) und erstellt zwei feste Ergebniszweige:

Zweig	Läuft, wenn
HOLDING	Der Greifer nach dem Greifen ein Teil hält.
NO PART	Der Greifer ohne Teil geschlossen hat.

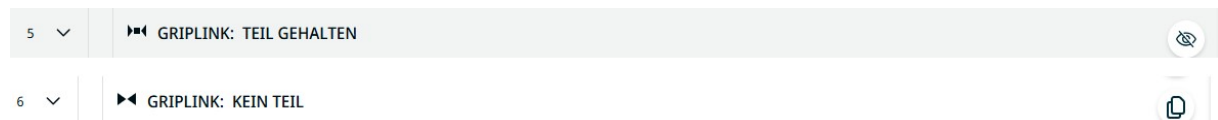


Abbildung 32: GRIPLINK Pick mit seinen beiden Ergebniszweigen - HOLDING (Teil gegriffen) und NO PART (kein Teil gegriffen).

GRIPLINK Place

Führt ein Freigeben aus (RELEASE oder FLEXRELEASE) und erstellt:

Zweig	Läuft, wenn
-------	-------------

RELEASED	Der Greifer das Teil erfolgreich freigegeben hat.
NOT RELEASED	Der Greifer RELEASED nicht erreicht hat (Teil noch gehalten, oder ein Fehler ist aufgetreten).

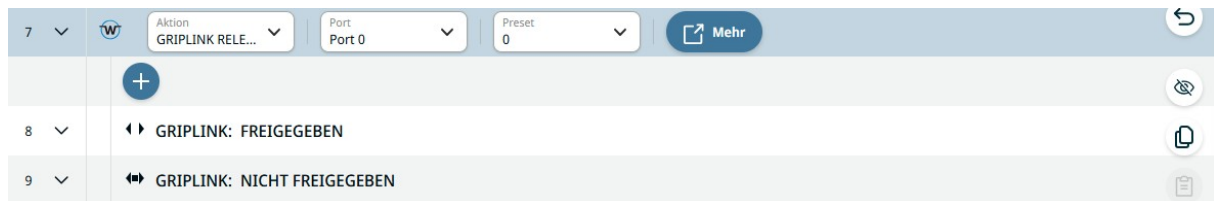


Abbildung 33: GRIPLINK Place mit den Zweigen RELEASED / NOT RELEASED.



Pick und Place warten immer (blockierend), bis der Vorgang abgeschlossen ist, bevor der Ergebnisweig ausgewertet wird; nur ein abgeschlossener Vorgang liefert ein eindeutiges Ergebnis.



Das Demo-Programm pick_and_place ist eine vollständige Pick-and-Place-Sequenz mit diesen Knoten (siehe Demo-Programme).

6.14 Greifkrafterhaltung steuern - CLAMP

Der Befehl CLAMP schaltet die Greifkrafterhaltung (PERMAGRIP) für das Gerät am ausgewählten Port ein oder aus; der Knoten hat dafür einen einzelnen **Clamp**-Schalter. Verfügbar auf kompatiblen Geräten (z. B. CRG-Serie).



Abbildung 34: GRIPLINK CLAMP - Port und der Clamp-Schalter.

6.15 Ansteuerung der Status-LED - LED

Der Befehl LED setzt die Status-LED des Geräts am ausgewählten Port auf eines von acht Preset-Mustern. Verfügbar auf kompatiblen Geräten (z. B. CRG-Serie).



Abbildung 35: GRIPLINK LED - nur Port und Preset.

6.16 Aktuellen Gerätezustand abfragen - DEVICESTATE

Der Befehl DEVICESTATE fragt den aktuellen Zustand des Geräts am ausgewählten Port ab und legt ihn in einer **Rückgabe**-Variablen ab, zur späteren Prüfung mit einem If-Knoten; mögliche Werte siehe

Anhang Gerätezustand. **Rückgabe** hat keinen Standardwert; der Knoten zeigt eine Validierungswarnung, bis eine Variable gewählt ist.



Abbildung 36: GRIPLINK DEVICESTATE - Port und Rückgabe (eine Variable muss ausgewählt werden).

Rückgabe listet Anwendungsvariablen vom Typ **Integer**. Eine solche Variable wird auf der Seite **Anwendungsvariablen** der Ansicht **Anwendung** angelegt: **Variable erstellen** drücken, einen Namen vergeben (z. B. DeviceState) und **Typ** auf **Integer** setzen. Die Variable bleibt zwischen den Sitzungen bestehen; Name und Typ sind mit der Erstellung festgelegt.

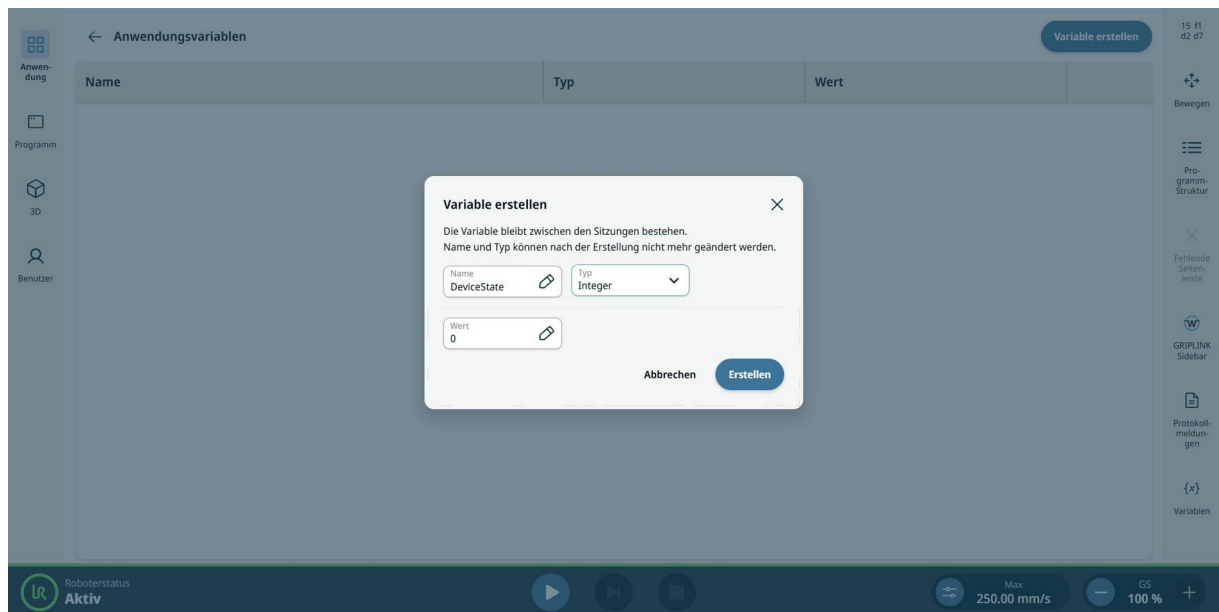


Abbildung 37: Der Dialog „Variable erstellen“ - eine Integer-Variable für das Feld Rückgabe von DEVICESTATE.



Dieser Knoten liest den aktuellen Zustand. Um zuerst auf die nächste Änderung zu warten, einen WSTR-Knoten unmittelbar davor verwenden.

6.17 Gerätewert abfragen - VALUE

Der Befehl VALUE fragt einen numerischen Wert (**Index** 0–7) des Geräts am ausgewählten Port ab und speichert das Ergebnis in einer **Rückgabe**-Variablen. Der Variablenwähler funktioniert wie bei DEVICESTATE, mit einem Unterschied: Er listet Variablen vom Typ **Float**, passend zum Datentyp des Werts.



Abbildung 38: GRIPLINK VALUE - Port, Index und Rückgabe.

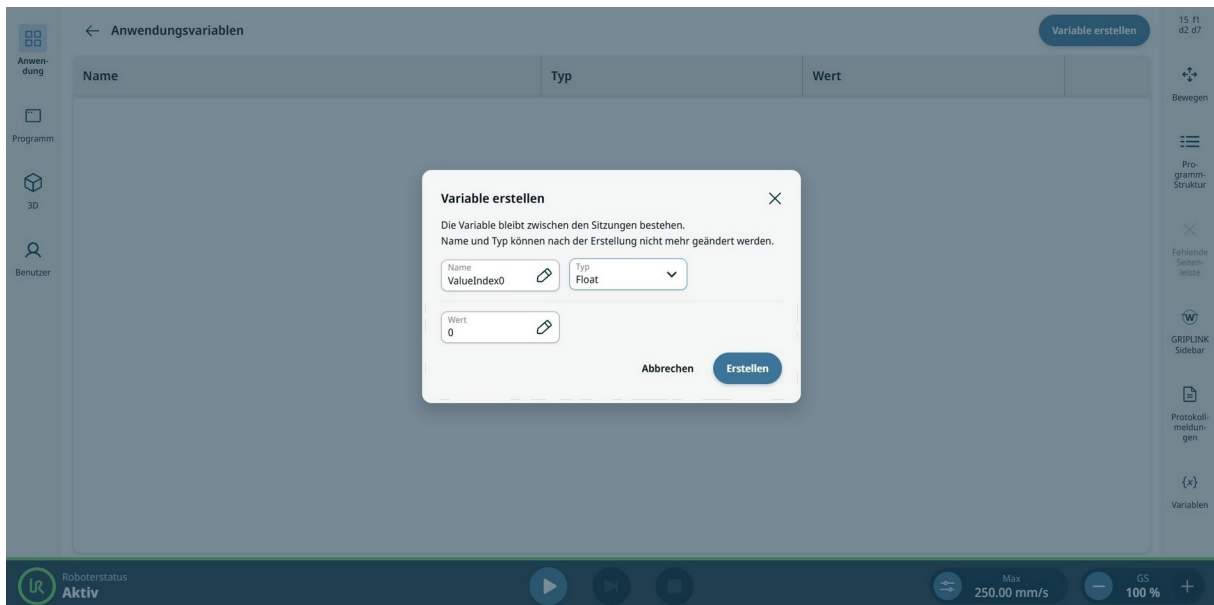


Abbildung 39: Der Dialog „Variable erstellen“ - eine Float-Variable für das Feld Rückgabe von VALUE.



Das Demo-Programm check_workpiece nutzt VALUE, um das gegriffene Teil anhand der Fingerposition zu prüfen.

6.18 Warten auf eine Änderung des Gerätezustands – WSTR

Der Befehl WSTR („Wait for State Transition“) wartet, bis das Gerät am ausgewählten Port seinen Zustand ändert, ohne eine weitere Aktion auszuführen. **Port** ist das einzige Feld, gleiches Layout wie HOME. Nützlich nach einem nicht blockierenden GRIP/RELEASE/ENABLE/DISABLE oder unmittelbar vor einer DEVICESTATE-Abfrage, die einen aktuellen Zustand benötigt.



Abbildung 40: GRIPLINK WSTR - Port ist das einzige Feld.



Das Demo-Programm multi_grip_release nutzt WSTR, um nicht blockierende Griffe über mehrere Ports zu synchronisieren.

6.19 Warten auf einen Gerätewert – WAITVAL

Der Befehl WAITVAL wartet, bis ein Wert einen Zielwert erreicht oder ein Timeout abläuft. Die Knotenzeile zeigt nur **Port** und **Konfigurieren**; die Parameter liegen in einem separaten Dialog.



Abbildung 41: GRIPLINK WAITVAL - Port und Konfigurieren.

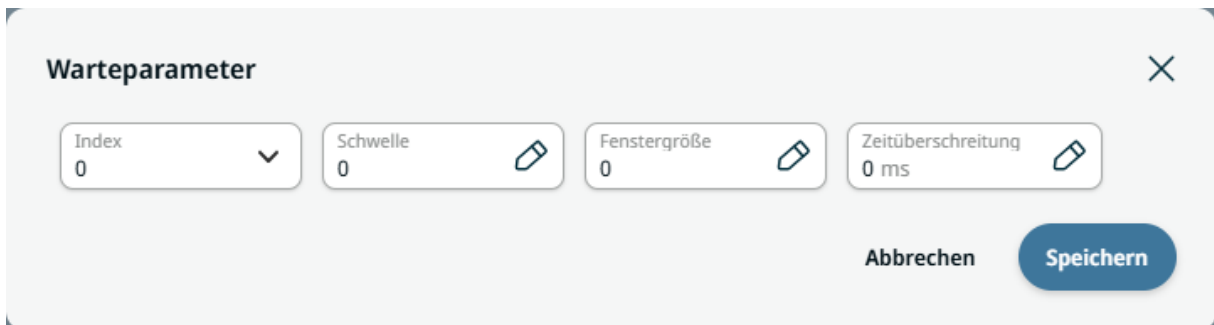


Abbildung 42: Der WAITVAL-Konfigurieren-Dialog - Index, Threshold, Window size, Timeout.

Parameter

Index	Der zu beobachtende Wert (0-7).
Threshold	Der Zielwert.
Window size	Das Toleranzband um Threshold, das als „erreicht“ zählt.
Timeout	Wie lange gewartet wird (ms), bevor aufgegeben wird.



Das Demo-Programm `async_preposition` nutzt WAITVAL, um zu warten, bis die Fingerposition das Zielfenster erreicht.

6.20 Gerätewert setzen – SETVALUE

Der Befehl SETVALUE setzt einen Wert in einem Wertkanal (**Index**) des Geräts am ausgewählten Port. Die Felder sind Port, Index und ein editierbares **Value**-Feld. Die Unterstützung dieses Befehls hängt vom Gerätetyp ab.



Abbildung 43: GRIPLINK SETVALUE - Port, Index und Value.



Das Setzen von Gerätewerten kann Aktoren bewegen.

6.21 Parametrierung eines Griff-Presets – SGRIPCFG

Der Befehl SGRIPCFG dient zur Parametrierung eines Griff-Presets. Die Knotenzeile zeigt Port, ein **Preset**-Auswahlfeld (0-7) und **Konfigurieren**, das den Dialog zur Programmierung dieses Presets

öffnet: ein **Tag** zur Benennung der Konfiguration und eine **Value interpretation**-Vorlage, die **P0–P7** für die gewählte Geräteklasse umbenennt und nur die von dieser Klasse verwendeten Parameter zeigt.

Konfigurieren ✕

Bezeichnung

Werteinterpretation

☒ Generisch

☐ Weiss Robotics IO-Link Greifer

☐ Weiss Robotics WPG/INTRAPAL Greifer

☐ Generischer Vakuumgreifer

P0 0 P1 0 P2 0 P3 0

P4 0 P5 0 P6 0 P7 0

Abbrechen Speichern

Abbildung 44: Der SGRIPCFG-Konfigurieren-Dialog - Tag, Value interpretation template und P0–P7.

Vorlage	P0	P1	P2	P3	P4	P5	P6	P7
Generic	P0 (ohne Einheit)	P1	P2	P3	P4	P5	P6	P7
Weiss Robotics IO-Link Gripper	No Part Limit (mm)	Release Limit (mm)	Force Factor (%)	–	–	–	–	–
Weiss Robotics WPG/INTRAPAL Gripper	No Part Limit (mm)	Release Limit (mm)	Force Factor (%)	Grip Velocity (mm/s)	Grip Acceleration (mm/s ²)	Release Velocity (mm/s)	Release Acceleration (mm/s ²)	–
Generic Vacuum Picker	Part Present Setpoint (kPa)	Energy Saving Setpoint (kPa)	Blow-off Duration (ms)	–	–	–	–	–



Das Wechseln der Value-interpretation-Vorlage und Speichern setzt jeden von der neuen Vorlage nicht verwendeten Parameter auf 0 zurück. Der Wechsel von WPG/INTRAPAL (7 Parameter) zu IO-Link (3 Parameter) löscht z. B. P3-P6.

7 URScripting

Jeder GRIPLINK-Befehl steht auch als reine URScript-Funktion (GL_*) zur Verfügung, direkt aus einem Skriptknoten aufrufbar; es sind dieselben Funktionen, die auch die grafischen Knoten intern aufrufen. Dieses Kapitel ist die Referenz für die fortgeschrittene Programmierung.

Der Applikationsknoten erzeugt einmal pro Programm eine GL_*-Präambel; Skriptknoten rufen diese Funktionen direkt auf.

Wie Fehler gemeldet werden

Schlägt ein Aufruf fehl, zeigt das Plugin ein Popup mit dem Namen des fehlgeschlagenen Befehls und einer **Retry**-Option; das Programm pausiert, bis der Bediener wiederholt oder stoppt; siehe Installation → Verhalten im Fehlerfall.

Beispiel

Ein Programm ruft GL_GRIP(0, 2, 1) auf, doch der GRIPLINK Controller hat gerade die Spannung verloren. Das Programm pausiert an dieser Zeile; ein Popup nennt GL_GRIP als fehlgeschlagenen Befehl und den Statuscode.

7.1 Verbindung aufbauen – GL_CONNECT

Dieser Befehl stellt die Verbindung zwischen dem GRIPLINK Controller und der Robotersteuerung her. Er muss der erste GRIPLINK-Aufruf im Programm sein; jede andere GL_*-Funktion setzt voraus, dass die Verbindung bereits aufgebaut ist. Der Befehl prüft weiterhin, ob der angeschlossene GRIPLINK Controller die Mindestanforderungen des GRIPLINK-Protokolls erfüllt (siehe Systemvoraussetzungen).

Die Verbindung verwendet immer TCP-Port 10001 am GRIPLINK Controller. Der Befehl schließt vor dem Öffnen stets eine eventuell bestehende Verbindung unter <SOCKET_NAME>, sodass ein erneuter Aufruf mitten im Programm (z. B. um mit anderer IP neu zu verbinden) ohne vorheriges GL_DISCONNECT sicher ist.



Die Tool-I/O-Schnittstelle wird über die IP-Adresse 127.0.0.1 angesprochen. Beim Aufruf von GL_CONNECT ist im Tool-I/O-Betrieb daher die IP-Adresse 127.0.0.1 zu verwenden. Der GRIPLINK-Socket ist für beide Schnittstellen identisch und lautet standardmäßig "sock_griplink".

Syntax

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


Parameter

string <IP> IP-Adresse des GRIPLINK Controllers (Ethernet) oder 127.0.0.1 (Tool I/O).

string <SOCKET_NAME> Name des Sockets, z. B. "sock_griplink". Von allen weiteren Aufrufen verwendet.

Häufige Ursachen einer fehlgeschlagenen Verbindung

- Falsche IP-Adresse oder abweichendes Subnetz.
- Spannungsloser GRIPLINK Controller oder nicht angeschlossenes Kabel.

Beispiel

Öffnen Sie die Verbindung einmalig zu Beginn des Programms, vor jedem anderen GRIPLINK-Aufruf:

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



Alle weiteren GL_-Funktionen akzeptieren <SOCKET_NAME> als optionalen letzten Parameter mit dem Standardwert "sock_griplink". Die Angabe ist nur erforderlich, wenn GL_CONNECT mit einem anderen Socket-Namen aufgerufen wurde.

7.2 Verbindung trennen - GL_DISCONNECT

Dieser Befehl trennt die Verbindung zwischen Robotersteuerung und GRIPLINK Controller. Dies kann notwendig werden, wenn über einen Werkzeugwechsler die Netzwerkverbindung zum GRIPLINK Controller unterbrochen wird, oder wenn das Programm den Controller vor dem Ende oder vor einer erneuten Verbindung freigeben muss.

Syntax

```
GL_DISCONNECT(<SOCKET_NAME>)
```

Parameter

string <SOCKET_NAME> Name des zu schließenden Sockets.

Beispiel

```
GL_DISCONNECT("sock_griplink")  
# ... Werkzeugwechsel hier ...  
GL_CONNECT("192.168.1.40", "sock_griplink")
```

7.3 Ein angeschlossenes Gerät prüfen - GL_DEVASSERT

Dieser Befehl prüft, ob am ausgewählten Port das vorgesehene Gerät angeschlossen ist. Besitzt das angeschlossene Gerät nicht die angegebene **Vendor-** und **Product-ID** (aus der IODD-Datei des Geräts), wird das Roboterprogramm angehalten. Verwenden Sie den Befehl am Programmanfang, um das korrekte Gerät zu bestätigen.

Syntax

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

Parameter

int <PORT> Port-Index (0...3 Standalone, 0...31 Proxy).
int <VENDOR_ID> Erwartete Vendor ID (IODD-Datei des Geräts).
int <PRODUCT_ID> Erwartete Product ID (IODD-Datei des Geräts).



Vendor- und Product-ID finden Sie in der Gerätebeschreibung bzw. IODD-Datei.

Beispiel

Prüfen Sie, ob an Port 0 ein Gerät von WEISS ROBOTICS (Vendor ID 815, Product ID 52) angeschlossen ist:

```
GL_DEVASSERT(0, 815, 52, "sock_griplink")
```

7.4 Gerät referenzieren - GL_HOME

Der Befehl HOME referenziert das ausgewählte Gerät. Er führt die integrierte Referenzierungsfunktion des Geräts aus und wartet, bis diese abgeschlossen ist.

Syntax

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

Parameter

int <PORT> Port-Index.



Das Verhalten variiert je nach Gerät - siehe die Betriebsanleitung des angeschlossenen Geräts.

Beispiel

Referenzieren Sie das Gerät an Port 0:

```
GL_HOME(0, "sock_griplink")
```

7.5 Gerät aktivieren – GL_ENABLE

Dieser Befehl aktiviert das am gewählten Port angeschlossene Gerät. IO-Link-Greifmodule wechseln dabei in den Zustand RELEASED und führen die Freigabeaktion des zuletzt gewählten Griff-Presets aus.

Mit dem Parameter <WSTR> kann das Warten auf einen Zustandsübergang (WSTR) aktiviert werden. Ist WSTR aktiviert, wartet die Befehlsausführung, bis der Gerätezustand am gewählten Port auf ENABLED oder FAULT wechselt. Erfolgt kein Zustandswechsel, endet die Befehlsausführung mit TIMEOUT.

Syntax

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

Parameter

int <PORT>	Port-Index.
bool <WSTR>	True wartet auf ENABLED, bevor zurückgekehrt wird.



Verwenden Sie <WSTR> = True nur, wenn das Gerät aktuell DISABLED ist - ENABLE auf einem bereits ENABLED-Gerät mit <WSTR> = True kann zu einem Timeout führen, da kein Übergang stattfindet.

Beispiel

Aktivieren Sie das Gerät an Port 0 und warten Sie, bis es einsatzbereit ist:

```
GL_ENABLE(0, True, "sock_griplink")
```

7.6 Gerät deaktivieren – GL_DISABLE

Dieser Befehl deaktiviert das am gewählten Port angeschlossene Gerät. Er kann beispielsweise genutzt werden, um die Finger eines Greifmoduls kraftfrei zu schalten oder den Laserstrahl eines Distanzsensors temporär zu deaktivieren.

Mit DISABLE können auch Gerätefehler quittiert werden, beispielsweise wenn sich das angeschlossene Gerät im FAULT-Zustand befindet. Mit <WSTR> = True wartet der Aufruf, bis der Gerätezustand auf DISABLED oder FAULT wechselt; erfolgt kein Zustandswechsel, endet die Befehlsausführung mit TIMEOUT.

Syntax

GL_DISABLE(<PORT>, <WSTR>, <SOCKET_NAME>)

Parameter

int <PORT>	Port-Index.
bool <WSTR>	True wartet auf DISABLED, bevor zurückgekehrt wird.



Verwenden Sie <WSTR> = True nur, wenn das Gerät aktuell ENABLED (oder FAULT) ist - DISABLE auf einem bereits DISABLED-Gerät mit <WSTR> = True kann zu einem Timeout führen.

Beispiel

Prüfen Sie beim Start, ob sich das Gerät an Port 0 im Zustand FAULT befindet, und quittieren Sie diesen vor dem Referenzieren:

```
state = GL_DEVSTATE(0, "sock_griplink")
if state == 7:
    GL_DISABLE(0, True, "sock_griplink")
    GL_ENABLE(0, True, "sock_griplink")
end
GL_HOME(0, "sock_griplink")
```

7.7 Werkstück greifen – GL_GRIP

Der Befehl GRIP führt einen Griff mit dem angegebenen Griff-Preset (0-7) aus. Über den Parameter <BLOCKING> wird festgelegt, ob der Befehl unmittelbar nach dem Start der Bewegung oder erst nach deren Abschluss zurückkehrt: Mit <BLOCKING> = 1 wartet der Aufruf, bis der Greifvorgang abgeschlossen ist; mit 0 kehrt er sofort nach Start der Bewegung zurück, während die Bewegung im Hintergrund weiterläuft.

Bei nicht blockierender Ausführung kann zusätzlich mit GL_WSTR auf eine Zustandsänderung gewartet und das Ergebnis anschließend mit GL_DEVSTATE geprüft werden.



Befindet sich ein Gerät bereits im Zustand HOLDING oder NO PART, weist das GRIPLINK-Protokoll einen erneuten GRIP-Befehl mit der Meldung „Service unavailable within current state“ ab.

Syntax

GL_GRIP(<PORT>, <PRESET>, <BLOCKING>, <SOCKET_NAME>)

Parameter

int <PORT>	Port-Index.
int <PRESET>	Preset-Index (0...7).
int <BLOCKING>	1 wartet, bis der Griff abgeschlossen ist, bevor zurückgekehrt wird; 0 kehrt sofort zurück.



Die Greifparameter können über die Weboberfläche des GRIPLINK Controllers oder über den Befehl GL_SETGRIPCFG konfiguriert werden.

Beispiel

Greifen Sie mit Preset 2 an Port 0; versuchen Sie es bei fehlendem Teil einmal erneut mit einem Freigeben:

```
GL_GRIP(0, 2, 1, "sock_griplink")
state = GL_DEVSTATE(0, "sock_griplink")
if state == 4:
# NO_PART - freigeben und erneut versuchen
GL_RELEASE(0, 2, 1, "sock_griplink")
GL_GRIP(0, 2, 1, "sock_griplink")
end
```

7.8 Werkstück freigeben – GL_RELEASE

Der Befehl RELEASE gibt ein zuvor gegriffenes Werkstück mit den Parametern des angegebenen Griff-Presets (0–7) frei. Mit <BLOCKING> = 1 wartet der Aufruf, bis die Freigabe abgeschlossen ist; mit 0 kehrt er sofort nach Start der Bewegung zurück.

Syntax

```
GL_RELEASE(<PORT>, <PRESET>, <BLOCKING>, <SOCKET_NAME>)
```

Parameter

int <PORT>	Port-Index.
int <PRESET>	Preset-Index (0...7).
int <BLOCKING>	1 wartet, bis die Freigabe abgeschlossen ist, bevor zurückgekehrt wird; 0 kehrt sofort zurück.

Beispiel

Geben Sie das an Port 0 gehaltene Teil mit Preset 2 frei:

```
GL_RELEASE(0, 2, 1, "sock_griplink")
```

7.9 Werkstück über Parameter greifen – GL_FLEXGRIP

Der Befehl FLEXGRIP führt einen Greifvorgang mit frei wählbaren Bewegungsparametern aus. Im Gegensatz zum GRIP-Befehl wird hier kein gespeichertes Grip-Preset verwendet, sondern der Bewegungsablauf vollständig über die Befehlsparameter bestimmt. Damit können individuelle Zielpositionen, Kräfte, Geschwindigkeiten und Beschleunigungen direkt im Befehl vorgegeben werden, nützlich, wenn sich das Ziel zur Laufzeit ändert.

Mit <BLOCKING> = 1 wartet der Aufruf, bis der Vorgang abgeschlossen ist (Werkstück gegriffen oder kein Werkstück erkannt); mit 0 kehrt er sofort zurück, während die Bewegung im Hintergrund ausgeführt wird.

Syntax

```
GL_FLEXGRIP(<PORT>, <POSITION>, <FORCE>, <SPEED>, <ACCELERATION>, <BLOCKING>,  
            <SOCKET_NAME>)
```

Parameter

int <PORT>	Port-Index.
float <POSITION>	Zielposition (mm).
float <FORCE>	Ziel-Greifkraft (N).
float <SPEED>	Zielgeschwindigkeit – physikalische mm/s im Ethernet-Betrieb; ein 0–200 %-Override im Tool-I/O-Betrieb.
float <ACCELERATION>	Zielbeschleunigung – physikalische mm/s ² im Ethernet-Betrieb; ein 0–100 %-Override im Tool-I/O-Betrieb. 0 verwendet den Gerätestandard.
int <BLOCKING>	1 wartet, bis die Bewegung abgeschlossen ist, bevor zurückgekehrt wird; 0 kehrt sofort zurück.

Beispiel

Greifen Sie an Port 0 mit 60 N Kraft, einem No-Part-Limit von 15 mm und 45,5 mm/s, blockierend:

```
GL_FLEXGRIP(0, 15.0, 60, 45.5, 0, 1, "sock_griplink")
```

7.10 Werkstück über Parameter freigeben – GL_FLEXRELEASE

Der Befehl FLEXRELEASE gibt ein zuvor gegriffenes Werkstück mit frei wählbaren Bewegungsparametern frei. Er kann sowohl für eine einfache Freigabe als auch für eine gezielte Vorpositionierung eines Servogreifers verwendet werden. Nach Aufruf des Befehls öffnet das Greifmodul die Finger bis zur angegebenen Zielposition; der Zustand ändert sich nach Abschluss typischerweise zu RELEASED.

Bei nicht blockierender Ausführung (<BLOCKING> = 0) kehrt der Befehl sofort zurück, während die Bewegung im Hintergrund läuft; mit <BLOCKING> = 1 wartet er bis zum Abschluss der Bewegung.

Syntax

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

Parameter

int <PORT>	Port-Index.
float <POSITION>	Zielposition (mm).
float <SPEED>	Zielgeschwindigkeit – physikalische mm/s im Ethernet-Betrieb; ein 10–100 %-Override im Tool-I/O-Betrieb.
float <ACCELERATION>	Zielbeschleunigung – physikalische mm/s ² im Ethernet-Betrieb; ein 0–100 %-Override im Tool-I/O-Betrieb. 0 verwendet den Gerätestandard.
int <BLOCKING>	1 wartet, bis die Bewegung abgeschlossen ist, bevor zurückgekehrt wird; 0 kehrt sofort zurück.

Beispiel

Positionieren Sie die Finger an Port 0 auf 19,5 mm vor, während der Roboter noch zum Greifpunkt fährt (nicht blockierend):

```
GL_FLEXRELEASE(0, 19.5, 0, 0, 0, "sock_griplink")  
movej(pick_approach)  
GL_WSTR(0, "sock_griplink")
```



Der Speed-Override liegt im Tool-I/O-Betrieb bei 10–100 %, der Acceleration-Override bei 0–100 % (100 % = Nennwert).

7.11 Greifkraftherhaltung steuern – GL_CLAMP

Mit dem CLAMP-Befehl können Greifmodule mit integrierter Greifkraftsicherung (PERMAGRIP) im Zustand HOLDING den Antrieb stromlos schalten, um den Strombedarf des Greifmoduls zu reduzieren. Das Werkstück wird durch die PERMAGRIP-Greifkraftherhaltung weiter gehalten.

Wird CLAMP aktiviert (<STATE> = True), schaltet sich der Antrieb im Zustand HOLDING ab. Wird CLAMP deaktiviert (<STATE> = False), bleibt die Greifkraftregelung aktiv und die Steuerung des Greifmoduls regelt die Greifkraft permanent nach. Verfügbar auf kompatiblen Geräten (z. B. CRG-Serie).

Syntax

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

Parameter

int <PORT>	Port-Index.
------------	-------------

bool <STATE> True aktiviert den Clamp; False löst ihn (ein boolescher Zustand, kein Kraftwert).



Erfordert ein Gerät mit integrierter Greifkrafterhaltung (z. B. CRG-Serie).

Beispiel

Aktivieren Sie den Clamp an Port 1 nach dem Greifen:

```
GL_GRIP(1, 0, 1, "sock_griplink")
GL_CLAMP(1, True, "sock_griplink")
```

7.12 Ansteuerung der Status-LED - GL_LED

Dieser Befehl aktiviert ein vordefiniertes LED-Preset der Status-LED eines angeschlossenen Geräts. Dadurch kann der Betriebszustand oder der Fortschritt eines Prozesses direkt am Endeffektor visuell signalisiert werden. Verfügbar auf kompatiblen Geräten (z. B. CRG-Serie).



LED-Presets können über die Weboberfläche des GRIPLINK Controllers konfiguriert werden.

Syntax

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

Parameter

int <PORT> Port-Index.
int <PRESET_INDEX> LED-Preset-Index (0...7).



Nur auf Geräten mit Status-LED verfügbar.

Beispiel

Greifen Sie an Port 2, setzen Sie dann die LED auf Preset 1, wenn das Teil breit genug ist, sonst Preset 2:

```
GL_GRIP(2, 0, 1, "sock_griplink")
width = GL_VALUE(2, 0, "sock_griplink")
if width >= 8.1:
```



```

GL_LED(2, 1, "sock_griplink")
else:
GL_LED(2, 2, "sock_griplink")
end

```

7.13 Aktuellen Gerätezustand abfragen - GL_DEVSTATE

Dieser Befehl liest den aktuellen Gerätezustand am angegebenen Port und gibt ihn direkt zurück; den Aufruf einer Variable zuweisen. Siehe den Anhang Gerätezustand für mögliche Werte und die zugehörigen URScript-Konstanten (z. B. S_HOLDING = 5, S_NO_PART = 4).

Syntax

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

Parameter

int <PORT> Port-Index.

Rückgabewert

int Aktueller Gerätezustand, siehe den Anhang Gerätezustand (z. B. 5 = HOLDING, 4 = NO_PART).

Beispiel

Warten Sie per Polling, bis das Gerät an Port 2 HOLDING (5) meldet:

```

GL_GRIP(2, 0, 0, "sock_griplink")
state = 0
while state != 5:
state = GL_DEVSTATE(2, "sock_griplink")
sleep(0.1)
end

```

7.14 Warten auf eine Änderung des Gerätezustands - GL_WSTR

Der Befehl WSTR wartet auf eine Änderung des Gerätezustands am angegebenen Port. Er kann beispielsweise verwendet werden, um zu prüfen, ob eine asynchron ausgeführte Bewegung (GL_GRIP/GL_RELEASE/GL_FLEXGRIP/GL_FLEXRELEASE mit <BLOCKING> = 0) abgeschlossen ist. Der Befehl blockiert, bis am angegebenen Gerät eine Zustandsänderung auftritt. Lesen Sie das Ergebnis anschließend mit GL_DEVSTATE.

Syntax

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

Parameter

int <PORT> Port-Index.



Tritt kein Zustandswechsel auf, wird Statuscode 5 (TIMEOUT) ausgelöst.

Beispiel

Positionieren Sie den Greifer vor, während der Roboter fährt, und warten Sie dann auf den Übergang, bevor Sie greifen:

```
GL_RELEASE(0, 2, 0, "sock_griplink")
movej(approach)
GL_WSTR(0, "sock_griplink")
state = GL_DEVSTATE(0, "sock_griplink")
if state != 3:
  popup("Gripper not RELEASED")
halt
end
movej(pick_point)
GL_GRIP(0, 2, 1, "sock_griplink")
```

7.15 Gerätewert abfragen - GL_VALUE

Dieser Befehl liest einen indizierten Gerätewert (0-7) aus und gibt ihn direkt zurück. Die Bedeutung jedes Index ist gerätespezifisch; der zulässige Wertebereich für den Wertindex kann in der Anleitung des jeweiligen Geräts nachgelesen werden.

Syntax

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

Parameter

int <PORT> Port-Index.

int <VALUE_INDEX> Index des Gerätewerts (0...7); welche Indizes verfügbar sind, hängt vom Gerätetyp ab.

Rückgabewert

float Der gelesene Wert. Bei unbrauchbarem Messwert ein Sentinel-Wert, siehe den Hinweis unten.



GL_VALUE gibt bei einem unbrauchbaren Messwert einen Sentinel-Wert zurück: 2147483647.0 für positiv unendlich, -2147483648.0 für negativ unendlich, -1.0 für NaN. Diese als ungültige Daten behandeln.

Beispiel

Lesen Sie nach dem Greifen an Port 3 die Teilbreite (Index 0) und verzweigen Sie auf eine OK/NOK-Prüfung:

```
GL_GRIP(3, 0, 1, "sock_griplink")
part_width = GL_VALUE(3, 0, "sock_griplink")
if part_width < 29.9 or part_width > 30.1:
# NOK - zur Ausschleusposition fahren
movej(reject_pose)
GL_RELEASE(3, 0, 1, "sock_griplink")
else:
# OK - zur Verwendungsposition fahren
movej(use_pose)
GL_RELEASE(3, 0, 1, "sock_griplink")
end
```

7.16 Warten auf einen Gerätewert – GL_WAITVAL

Der Befehl WAITVAL wartet, bis ein bestimmter Gerätewert den zuvor gesetzten Sollwert erreicht oder innerhalb eines definierten Toleranzbereichs liegt; das vermeidet Polling von GL_VALUE in einer Schleife. Der überwachte Wert entspricht dem über GL_VALUE auslesbaren Indexwert. Der Befehl blockiert, bis eine der folgenden Bedingungen erfüllt ist:

- Der überwachte Wert liegt innerhalb des vorgegebenen Toleranzfensters, oder
- das Timeout wurde überschritten.

Syntax

```
GL_WAITVAL(<PORT>, <VALUE_INDEX>, <THRESHOLD>, <WINDOW_SIZE>, <TIMEOUT>,
           <SOCKET_NAME>)
```

Parameter

```
int <PORT>          Port-Index.
int <VALUE_INDEX>   Index des Gerätewerts (0..7).
float <THRESHOLD>   Der zu erreichende Zielwert.
float <WINDOW_SIZE> Toleranzfenster um den Schwellwert.
int <TIMEOUT>       Maximale Wartezeit in Millisekunden (ms).
```



Erreicht der Wert das Fenster nicht innerhalb von <TIMEOUT>, wird ein Timeout-Fehler (Statuscode 5) ausgelöst und das Programm unterbrochen.

Beispiel

Warten Sie bis zu 5 Sekunden (5000 ms), bis ein Abstandssensor an Port 2 $100,0 \pm 0,5$ erreicht:
`GL_WAITVAL(2, 0, 100.0, 0.5, 5000, "sock_griplink")`

7.17 Gerätewert setzen – GL_SETVAL

Mit diesem Befehl kann ein gerätespezifischer Wert gesetzt werden. Der Wert ist über einen Index adressierbar, dessen gültiger Wertebereich vom angeschlossenen Gerät abhängt; auch die Bedeutung des geschriebenen Werts ist gerätespezifisch. Die Unterstützung dieses Befehls hängt vom Gerätetyp ab.



Wertebereiche von Wertindex und zu schreibendem Wert entnehmen Sie der Anleitung des jeweiligen Geräts.

Syntax

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

Parameter

`int <PORT>` Port-Index.
`int <VALUE_INDEX>` Index des Gerätewerts (0...7).
`float <VALUE>` Zu schreibender Wert.



Das Setzen von Gerätewerten kann Aktoren bewegen.

Beispiel

Setzen Sie Wertkanal 2 an Port 3 auf 10,5:
`GL_SETVAL(3, 2, 10.5, "sock_griplink")`

7.18 Parametrierung eines Griff-Presets – GL_SETGRIPCFG

Der Befehl SETGRIPCFG dient zur Anpassung der Parameter eines Griff-Presets. Ein Griff-Preset definiert die Bewegungs- und Kraftparameter, die ein Greifmodul beim Ausführen eines GRIP- oder RELEASE-Befehls verwendet. Für jedes Griff-Preset können bis zu acht Parameter (<P0>...<P7>) konfiguriert werden; ihre Bedeutung und zulässigen Wertebereiche sind gerätespezifisch.

Syntax

```
GL_SETGRIPCFG(<PORT>, <PRESET>, <TAG>, [<P0>, <P1>, <P2>, <P3>, <P4>, <P5>, <P6>, <P7>], <SOCKET_NAME>)
```

Parameter

int <PORT> Port-Index.

int <PRESET> Zu konfigurierender Preset-Index (0...7).

string <TAG> Freitext-Tag zur Identifikation der Konfiguration (Buchstaben, Ziffern sowie
 - _ * + # % / < > :, max. 32 Zeichen).

float <P0> ... <P7> Bis zu acht gerätespezifische Konfigurationsparameter, als URScript-Liste.

Beispiel

Konfigurieren Sie für einen IO-Link-Greifer von WEISS ROBOTICS an Port 3 Preset 1 mit 31 mm No-Part-Limit, 41 mm Release-Limit und 59 % Force Factor:

```
GL_SETGRIPCFG(3, 1, "bearing", [31, 41, 59, 0, 0, 0, 0, 0],  
              "sock_griplink")
```

8 Fehlersuche und Diagnose

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.

Fehlerbehandlungsmodell

Ordnen Sie den Fehler zuerst einer Ebene zu: ein Kommunikationsproblem, ein Gerätezustandsproblem und ein Tool-I/O-Zugriffsproblem erfordern unterschiedliche Prüfungen.

Ebene	Typisches Symptom	Zuerst prüfen	Empfohlener nächster Schritt
Ethernet-Kommunikation	Check GRIPLINK Controller schlägt fehl, oder Befehle können nicht verbinden.	IP-Adresse, Subnetz, Controller-Spannung, Kabel, Weboberfläche.	Netzwerkeinstellungen korrigieren und die Prüfung erneut ausführen.
Tool-I/O-Zugriff	Waiting for Tool I/O access, oder Statuscode 100/101.	Tool-I/O-Zugriffszustand, konkurrierende URCapXs/Funktionen.	GRIPLINK die Tool-I/O-Steuerung überlassen; Rescan drücken, sobald das Gerät mit Spannung versorgt ist.
Gerätezustand	Befehl stoppt: DISABLED, FAULT oder nicht initialisiert.	Anhang Gerätezustand und Sidebar-Statuszeile.	Aktivieren, referenzieren, Fehler quittieren oder Programmablauf anpassen.
Prozessergebnis	NO PART oder Timeout trotz funktionierender Kommunikation.	Greifparameter, Preset, Werkstück, Hub, Kraft, Zielzustand.	Mit DEVICESTATE, WSTR, VALUE oder WAITVAL sicher verzweigen.

8.1 Fehlersuche-Matrix

Prüfen Sie zuerst diese Matrix; die einzelnen Themen unten ergänzen Details zu den häufigsten Situationen bei Inbetriebnahme und Betrieb.

Symptom / Meldung	Wahrscheinliche Ursache	Prüfen	Maßnahme	Verwandter Abschnitt
Tool I/O Interface zeigt	PolyScope X hat die Tool-I/O-	Nur ein Plugin darf Tool I/O	Sicherstellen, dass	Tool-I/O-spezifische

Waiting for Tool I/O access	Steuerung nicht erteilt, oder eine andere Funktion steuert sie.	steuern.	ausschließlich GRIPLINK Tool I/O steuert; Programm oder Schnittstelle neu starten, falls der Zugriff entzogen wurde.	Statuscodes
Tool I/O Interface zeigt Waiting for device	Kein kompatibles Gerät an der Tool-I/O-Schnittstelle erkannt.	Kabel, Spannung, unterstützter Gerätetyp.	Nach Wiederherstellen von Spannung/Verka belung Rescan drücken.	Gerät anschließen und trennen
Scanning devices... bleibt zu lange sichtbar	Das Plugin sucht nach Geräten, oder die Schnittstelle hängt fest.	2-5 Sekunden warten; prüfen, ob sich die Gerätetabelle aktualisiert.	Restart Interface verwenden; schlägt dies fehl, den Roboter aus- und wieder einschalten.	Rescan und Restart
Rescan ist nicht verfügbar	Ein Roboterprogramm hält die Schnittstelle.	Laufendes/ pausiertes Programm.	Das Programm stoppen und aktive GRIPLINK-Verbindungen schließen, bevor erneut gescannt wird.	Hinweise zur Verwendung der Sidebar
Ethernet Check GRIPLINK Controller schlägt fehl	Falsche IP, anderes Subnetz, spannungsloser Controller oder Kabelproblem.	IP-Adresse, Subnetz, Kabel, Spannung, Standard 192 . 168 . 1 . 4 0.	Netzwerkeinstellungen korrigieren und Check GRIPLINK Controller erneut drücken.	Ethernet-Betrieb
Befehls-Popup meldet NO_COMM / Statuscode 18	Kommunikation zum Controller oder Tool-I/O-Gerät unterbrochen.	Ethernet-Verbindung, Spannung, Tool-I/O-Schnittstelle, Controller-Zustand.	Spannung/ Kommunikation wiederherstellen und erneut versuchen; im Tool-I/O-Betrieb Rescan drücken, falls das Gerät spannungslos war.	Statuscodes
Befehl meldet TIMEOUT / Statuscode 5	Gerät hat Zielzustand/-wert nicht rechtzeitig erreicht.	Zielzustand, Greifparameter, Blockierend-Option.	Von einem gültigen Zustand ausgehen, Greifparameter	Timeout bei aktivierter Option „Auf Zustandsübergang

			prüfen oder Blockierend deaktivieren.	g warten“
Nicht blockierender Befehl verhält sich nicht asynchron	Gerätefirmware unterstützt das erforderliche asynchrone Verhalten nicht.	Gerätetyp und Firmware-Version.	Bei GRIPKIT-EASY-Firmware unter 3.0.0 die Firmware aktualisieren oder blockierend ausführen.	Nicht blockierende Ausführung bei GRIPKIT-EASY-Firmware-Versionen unter 3.0.0
Sidebar- oder Smart-Skill-Befehle sind blockiert	Das Roboterprogramm ist aktiv und die Verbindung ist offen.	Roboterprogramm.	Die laufende Skriptverbindung schließen oder das Programm beenden, bevor die manuelle Bedienung verwendet wird.	Hinweise zur Verwendung der Sidebar
Gerät wechselt in FAULT	Gerät meldet einen Fehler, oder ein Befehl war für den Zustand ungültig.	Gerätezustand, Gerätehandbuch, Fehler-Popup.	DISABLE verwenden, Gerätehandbuch und Statuscodes prüfen.	Gerätezustand / Statuscodes
Pick-/Place-Zweig liefert NO PART	Der Greifer hat nach der Bewegung kein Teil erkannt.	Preset, Position, Kraft, Werkstückgeometrie, Sensorzustand.	Preset anpassen oder FLEXGRIP/FLEXRELEASE mit passenden Parametern verwenden.	Pick / Place
INVALID_PARAMETER / Statuscode 10	Mindestens ein Parameter liegt außerhalb des zulässigen Bereichs.	Port, Preset, Position, Kraft, Geschwindigkeit, Index.	Parameter mit der Befehlsreferenz und den Gerätefähigkeiten abgleichen.	Programmknoten - Befehlsreferenz / URScripting

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

Mögliche Ursache

Der Tool-I/O-Zugriff wurde von PolyScope X nicht an GRIPLINK erteilt; das Feld **Tool I/O Interface** zeigt Waiting for Tool I/O access (Statuscode 100/101). Eine andere Funktion oder ein anderes URCapX steuert die Tool-I/O-Schnittstelle.

Empfohlene Maßnahme

Prüfen Sie in den Robotereinstellungen unter **Communication → Robot → Tool I/O**, dass **Controlled By** auf **GRIPLINK** steht, und stellen Sie sicher, dass GRIPLINK das einzige URCapX bzw. die einzige Funktion ist, die die Tool-I/O-Schnittstelle steuert. Starten Sie das Programm oder die Tool-I/O-Schnittstelle neu, falls ein zuvor erteilter Zugriff entzogen wurde. Kontrollieren Sie anschließend im Applikationsknoten des GRIPLINK-Plugins, ob der Tool-I/O-Zugriff wieder erteilt ist.

8.3 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. Die Tool-I/O-Schnittstelle verfügt über keine Hot-Plug-Erkennung; das Feld **Tool I/O Interface** zeigt Waiting for device.

Empfohlene Maßnahme

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

8.4 Timeout bei aktivierter Option „Auf Zustandsübergang warten“

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. Nach dem internen Timeout wird ein Timeout-Fehler (Statuscode 5) ausgelöst.

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 DEVICESTATE / GL_DEVSTATE**.

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

Mögliche Ursache

GRIPKIT-EASY-Geräte mit einer Firmware-Version unter **3.0.0** unterstützen für Greif- und Freigabebefehle nur die blockierende Ausführung.

Empfohlene Maßnahme

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

9 Anhänge

9.1 Gerätezustand

Die folgende Tabelle listet die möglichen Zustandswerte angeschlossener Geräte auf. Der Gerätezustand wird von GRIPLINK DEVICESTATE / GL_DEVSTATE zurückgegeben. Die in der rechten Spalte angegebenen URScript-Konstanten werden vom Plugin definiert und können im Roboterprogramm verwendet werden.

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 URCapX beeinträchtigt wird und es zu Fehlverhalten kommen kann!

9.2 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. Tool-I/O-spezifische Codes (100+) sind im Kapitel Tool-I/O-Integration aufgeführt.

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

9.3 Tool-I/O-spezifische Statuscode-Aktionen

Diese Codes beschreiben den Zustand der Tool-I/O-Schnittstelle; sie sind von den GRIPLINK-Protokoll-Statuscodes oben zu unterscheiden.

Code	Name	Bedeutung	Erste Prüfung / empfohlene Maßnahme
100	TOOL_IO_NOT_GRANTED	PolyScope X hat den Tool-I/O-Zugriff nicht erteilt.	Sicherstellen, dass das GRIPLINK-Plugin Tool I/O steuern darf und keine andere Funktion es verwendet.
101	TOOL_IO_REVOKED	Der Tool-I/O-Zugriff wurde von PolyScope X entzogen.	Programm stoppen, Zugriff wiederherstellen, Schnittstelle bei Bedarf neu starten.
102	TOOL_IO_CONNECTING	Ein Gerätescan an der Tool-I/O-Schnittstelle läuft.	Warten, bis der Scan abgeschlossen ist; während des Scans keine Befehle senden.
103	DAEMON_BOOTING	Das Tool-I/O-Backend startet.	Nach Installation/Neustart einige Sekunden warten.
104	DAEMON_SHUTDOWN	Das Tool-I/O-Backend wird beendet.	Vor weiteren Befehlen das Beenden/den Neustart abwarten.

9.4 Versionsverlauf

Datum	Version	Typ	Beschreibung	Autor
07/2026	1.0.0	Initiale Version	GRIPLINK-Plugin für Universal Robots PolyScope	H. Efkere

			X. Unterstützt das GRIPLINK- Protokoll über TCP/IP (Ethernet) sowie das FLEXGRIP- Protokoll über Modbus RTU (Tool I/O).	
--	--	--	---	--

© 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. Unsere Produkte sind nicht für den Einsatz in lebenserhaltenden Systemen oder für Systeme, bei denen ein Fehlverhalten zu Personenschäden führen könnte, vorgesehen.

GRIP SMARTER.



WEISS ROBOTICS