



GRIPLINK-PLUGIN FÜR FANUC

Version 4.1.0

August 2026



Inhalt

1	Einführung.....	4
1.1	Notation und Symbole	4
1.2	Bestimmungsgemäße Verwendung.....	4
1.3	Systemvoraussetzungen	4
1.4	Lizenzbestimmungen	5
2	Aufbau des GRIPLINK-Plugins	6
3	Änderungen gegenüber vorigen Versionen.....	7
4	Installation	9
4.1	Vorbereitung des Roboters.....	9
4.2	Installation des GRIPLINK Core-Plugins für Standard-Roboter	10
4.3	Installation des GRIPLINK CRX-Plugins.....	12
4.4	Deinstallieren der Software	13
4.5	Konfiguration der Socket Messaging Schnittstelle	14
4.6	Verhalten im Fehlerfall	14
5	Befehlsreferenz (Core-Plugin).....	16
5.1	Verbindung aufbauen – CONNECT (1)	18
5.2	Verbindung trennen – BYE (2)	19
5.3	Angeschlossene Geräte prüfen – ASRTDEV (3).....	20
5.4	Gerät aktivieren – ENABLE (11)	21
5.5	Gerät deaktivieren – DISABLE (12).....	22
5.6	Referenzieren – HOME (13)	23
5.7	Werkstück über Preset greifen – GRIP (20)	24
5.8	Werkstück über Preset freigeben – RELEASE (22)	26
5.9	Werkstück über Parameter greifen – FLEXGRIP (26).....	27
5.10	Werkstück über Parameter freigeben – FLEXRELEASE (27).....	28
5.11	Parametrierung eines Griff-Presets – SGRIPCFG (60)	29
5.12	Aktuellen Gerätezustand abfragen – STATE (10).....	31
5.13	Warten auf eine Änderung des Gerätezustands – WSTR (80).....	32
5.14	Gerätewerte abfragen – VALUE (41).....	33
5.15	Gerätewerte setzen – SETVAL (42)	34
5.16	Warten auf einen Gerätewert – WAITVAL (43)	35
5.17	Greifkraftherhaltung steuern – CLAMP (24)	36
5.18	Ansteuerung der Status-LED – LED (25).....	37
5.19	Veraltete Befehle	38
5.19.1	Gleichzeitiges Greifen von Werkstücken – MGRIP (21).....	38
5.19.2	Gleichzeitiges Freigeben von Werkstücken – MRELEASE (23).....	40
5.19.3	Warten auf die Änderung mehrerer Gerätezustände – MWAITFOR (81)	41
5.19.4	Aktuelle Fingerposition auslesen – POS (40)	42

6	Grafische Programmierung mit dem FANUC CRX-Plugin.....	43
6.1	Funktionsweise	43
6.2	Hardware-Inbetriebnahme	44
6.3	Vorbereitung des Plugins	45
6.4	Anwendungen mit Instruktionen erstellen.....	47
6.5	Stack-Größe	47
6.6	Empfohlene Programmstruktur.....	48
6.7	Übersicht der verfügbaren Instruktionen	48
6.7.1	Verbindung aufbauen – CONNECT	50
6.7.2	Verbindung trennen – BYE.....	51
6.7.3	Angeschlossene Geräte prüfen – ASSERTDEV	52
6.7.4	Gerät aktivieren – ENABLE.....	53
6.7.5	Gerät deaktivieren – DISABLE	54
6.7.6	Referenzieren – HOME	55
6.7.7	Werkstück über Preset greifen – GRIP.....	56
6.7.8	Werkstück über Preset freigeben – RELEASE	57
6.7.9	Werkstück über Parameter greifen – FLEXGRIP	58
6.7.10	Werkstück über Parameter freigeben – FLEXRELEASE	60
6.7.11	Parametrierung eines Griff-Presets – SGRIPCFG	62
6.7.12	Auslesen des Gerätezustands – DEVSTATE.....	64
6.7.13	Warten auf eine Änderung des Gerätezustands – WAITSTATE	65
6.7.14	Gerätewert abfragen – VALUE.....	66
6.7.15	Gerätewert setzen – SETVALUE	67
6.7.16	Warten auf einen Gerätewert – WAITVALUE	68
6.7.17	Greifkrafterhaltung steuern – CLAMP	69
6.7.18	Ansteuerung der Status-LED – LED	70
7	Fehlersuche.....	71
7.1	TELNET-Ausgabe	71
7.2	Fehlermeldungen.....	72
7.2.1	Installationsfehlermeldung „MEMO-094 Parameters are different“	72
7.2.2	Systemfehlermeldung „EtherNet Adapter Error (1)“	72
7.2.3	Meldung „Could send command [command]: not connected“	73
7.2.4	Meldung „GRIPLINK: Cmd [command] failed: E_NOT_OPEN“	73
7.2.5	Meldung „GRIPLINK Version not supported“	74
7.2.6	Meldung „GL_COMM 190: undefined built-in“	74
7.2.7	Meldung „GRIPLINK could not connect via SM slot Cx:“	75
7.2.8	Meldung „Listen failed:“	75
7.2.9	Meldung „Device x not connected“	75
7.2.10	Meldung „Stack Overflow“	75
7.2.11	Meldung „GRIPLINK: Failed to listen on socket“	76
7.2.12	Verbindung zwischen Roboter und GRIPLINK-Controller nicht trennbar.....	76

Anhang A	Gerätezustand.....	78
Anhang B	Statuscodes.....	79

1 Einführung

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

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 Notation und Symbole

Zur besseren Übersicht werden in dieser Anleitung folgende Symbole verwendet:



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



Zusatzinformation zum besseren Verständnis des beschriebenen Sachverhalts.



Verweis auf weiterführende Informationen.

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 darüberhinausgehende Verwendung gilt als nicht bestimmungsgemäß. Für hieraus resultierende Schäden haftet der Hersteller nicht.

1.3 Systemvoraussetzungen

Dieses Plugin ist kompatibel mit der **GRIPLINK Protokollversion 3** und höher.

Zum Betrieb wird eine der folgenden FANUC Robotersteuerungen benötigt:

- R-30iB mit Softwarestand 8.30 (oder höher)
- R-30iB plus mit Softwarestand 9.10 (oder höher)

Zum Betrieb des Plugins auf einem Roboter der CRX-Serie wird folgende Robotersteuerung benötigt:

- R-30iB plus mit Softwarestand V9.40P/06 (oder höher) mit Tablet Teach Pendant

Folgende Roboter-Optionen werden zum Betrieb der Software benötigt:

- R632 KAREL
- R648 User Socket Msg



Kontaktieren Sie Ihren FANUC Händler zum Bezug der Roboter-Optionen.



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.

1.4 Lizenzbestimmungen

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

2 Aufbau des GRIPLINK-Plugins

Das GRIPLINK-Plugin für FANUC besteht aus zwei aufeinander aufbauenden Softwarekomponenten:

Core-Plugin

Das Core-Plugin stellt die grundlegende Kommunikationsschnittstelle zwischen der Robotersteuerung und dem GRIPLINK-Controller bereit. Es implementiert die GRIPLINK-Befehle als textbasierte Funktionsaufrufe, die aus Roboterprogrammen heraus aufgerufen werden können.

Das Core-Plugin ist für klassische FANUC-Industrierobotersteuerungen ohne Tablet Teach Pendant ausgelegt und ermöglicht die vollständige Steuerung von GRIPLINK-kompatiblen Geräten über die in Kapitel 5 beschriebene Befehlsreferenz.

CRX-Plugin

Für Roboter der FANUC CRX-Serie steht zusätzlich ein erweitertes Plugin zur Verfügung. Dieses baut auf dem Core-Plugin auf und stellt dessen Funktionen in Form von grafischen Instruktionsblöcken im Programmeditor des Tablet Teach Pendants bereit.

Die Instruktionen können per Drag-and-Drop in Roboterprogramme eingefügt werden und erleichtern die Erstellung von Anwendungen ohne direkte textuelle Programmierung. Intern greifen diese Instruktionen auf die Funktionen des Core-Plugins zurück. Daher stehen beide Programmiermethoden, grafisch oder textbasiert, parallel zur Verfügung. Die Verwendung des CRX-Plugins wird in Kapitel 6 beschrieben.

3 Änderungen gegenüber vorigen Versionen

Neu in Version 4.1.0

Version 4.1.0 baut auf Version 4.0.0 auf und bringt Optimierungen für die CRX-Benutzeroberfläche sowie interne Stabilitätsverbesserungen. Alle Befehle, Parameter und Signaturen bleiben unverändert. Bestehende Programme laufen ohne Anpassung weiter.

GRIPLINK-Befehle jetzt in der Kategorie Plugin

Die GRIPLINK-Befehle befinden sich in der Befehlsauswahl nun in der Kategorie Plugin (bisher I/O).



Es wird empfohlen, nach der Aktualisierung von 4.0.0 auf 4.1.0 jeden GRIPLINK-Programmknoten einmalig zu öffnen und das Rückgaberegister sowie die Befehlsparameter zu kontrollieren. Knoten ohne gültiges Rückgaberegister werden zur Laufzeit durch eine eindeutige Fehlermeldung angezeigt.

Neu in Version 4.0.0

Mit Version 4.0.0 wurde das GRIPLINK-Plugin grundlegend überarbeitet. Die vorgenommenen Änderungen betreffen sowohl die Parametrierung als auch die verfügbaren Befehle und deren Verhalten.

Wichtiger Hinweis: Diese Version ist **nicht vollständig abwärtskompatibel**. Bestehende Anwendungen müssen überprüft und angepasst werden.

Umstellung auf Fließkomma-Parameter (REAL)

Die bisher verwendeten skalierten Ganzzahlparameter (INTEGER) wurden durch Fließkommawerte (REAL) ersetzt. In früheren Versionen wurden physikalische Werte (z. B. Positionen oder Geschwindigkeiten) mit einem Skalierungsfaktor (typischerweise 1000) übertragen.



Ab Version 4.0.0 erfolgt die Übergabe direkt in physikalischen Einheiten.

Diese Änderung betrifft die folgenden Befehle:

- FLEXGRIP
- FLEXRELEASE
- SGRIPCFG
- VALUE
- SETVAL
- WAITVAL

Auswirkungen

- Skalierungsfaktoren (z. B. Faktor 1000) dürfen nicht mehr verwendet werden
- Parameterwerte müssen auf reale physikalische Werte umgestellt werden (z. B. mm statt mm × 1000)

Beispiel

Bis Version 3.x.x	Ab Version 4.0.0
POSITION = 19500	POSITION = 19,5

Weiterhin verfügbare (veraltete) Befehle

Die folgenden Befehle sind weiterhin verfügbar, werden jedoch als **veraltet (deprecated)** eingestuft:

- MGRIP
- MRELEASE
- MWAITFOR
- POS

Diese Befehle bleiben aus Gründen der Abwärtskompatibilität erhalten, sollten für neue Projekte aber nicht mehr genutzt werden. Für neue Anwendungen wird empfohlen, die folgenden Befehle zu verwenden:

- GRIP / RELEASE in Kombination mit DOBLOCK = 0 anstelle von MGRIP/MRELEASE
- WSTR zur Zustandsüberwachung anstelle von MWAITFOR
- VALUE zur Abfrage von Gerätewerten anstelle von POS

Die neue Vorgehensweise bietet eine höhere Flexibilität bei der Programmgestaltung, da Abläufe und Logik individuell an die jeweilige Anwendung angepasst werden können. Gleichzeitig ermöglicht sie eine bessere Kontrolle über den Programmablauf und die Synchronisation mehrerer Geräte. Darüber hinaus sorgt die Verwendung einer einheitlichen Schnittstelle für eine konsistente Ansteuerung unterschiedlicher Gerätetypen.

Hinweis zur Migration

Durch die Änderungen in Version 4.0.0 sind bestehende Programme **nicht vollständig abwärtskompatibel**. Vorhandene Anwendungen müssen insbesondere hinsichtlich der folgenden Punkte überprüft und angepasst werden:

- Umstellung von INTEGER- auf REAL-Parameter
- Entfernung von Skalierungsfaktoren



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

4 Installation



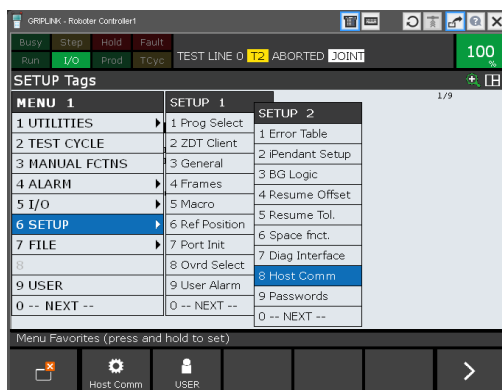
Stellen Sie sicher, dass Sie die aktuelle Version des GRIPLINK-Plugins verwenden. Die aktuelle Version kann unter www.griplink.de/software heruntergeladen werden.

4.1 Vorbereitung des Roboters

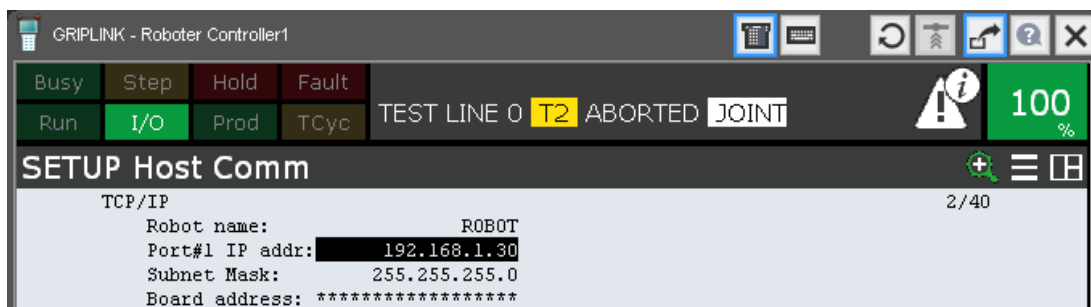
Schalten Sie den Roboter ein und richten Sie die IP-Adresse ein (z.B. 192.168.1.30). Stellen Sie dabei sicher, dass sich Roboter und der GRIPLINK-Controller im gleichen Subnetz befinden.

Führen Sie dazu folgende Schritte aus:

1. Wählen Sie MENU → 6 (SETUP) → 0 (--NEXT--) → 0 (--NEXT--) → 7 (Host Comm).
2. Wählen Sie „TCP/IP“ aus und drücken Sie ENTER.



3. Geben Sie die IP-Adresse und die Subnetz-Maske des Roboters ein (im Beispiel 192.168.1.30, Subnetz 255.255.255.0).



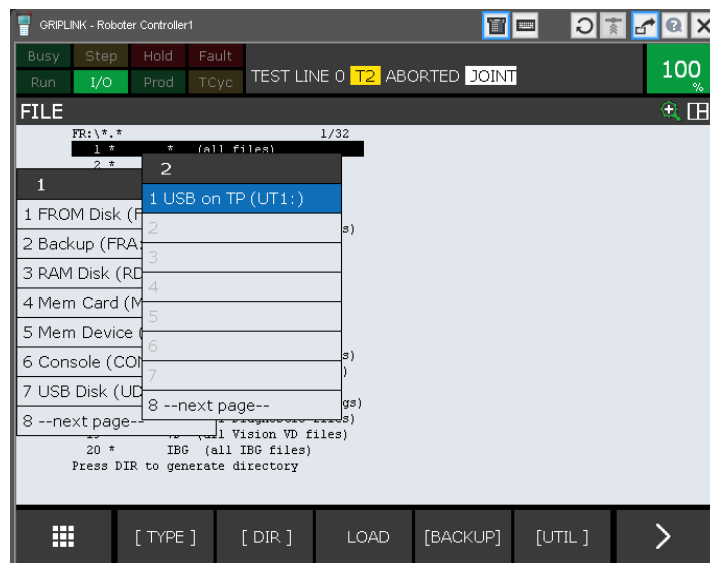
4.2 Installation des GRIPLINK Core-Plugins für Standard-Roboter

Für Standard-Roboter ohne Table Teach Pendant installieren Sie das Core-Plugin direkt über das Teach Pendant Ihrer Robotersteuerung.

1. Laden Sie die Plugin-Datei „griplink_fanuc_<Version>.zip“ herunter.
2. Entpacken Sie das ZIP-Archiv in das Stammverzeichnis eines USB-Sticks und stecken Sie diesen in den USB-Slot des Teach Pendants.
3. Drücken Sie die Taste Menu → 7 (FILE) → 1 (File)



4. Drücken Sie den Softkey [UTIL] → 1 (Set Device)
5. Wählen Sie 8 (-- next page --) → 1 (USB on TP (UT1:))



6. Wählen Sie „All Files“. Sie sollten nun unter anderem die Installationsdatei „SETUP.CM“ sehen.

7. Führen Sie die Installationsdatei aus. Bei Steuerungen der R-30iB Serie müssen Sie zum Anwenden der Parameter danach einen Kaltstart durchführen.

Überprüfen der Installation

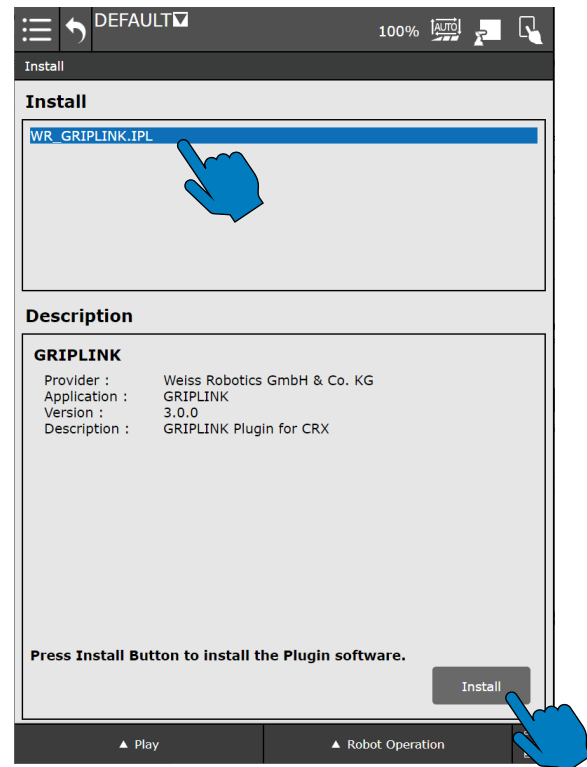
Nachdem Sie den Installationsprozess beendet haben, drücken Sie den SELECT auf dem Teach Pendant. Sie sollten nun das Programm „GRIPLINK.PC“ sehen.

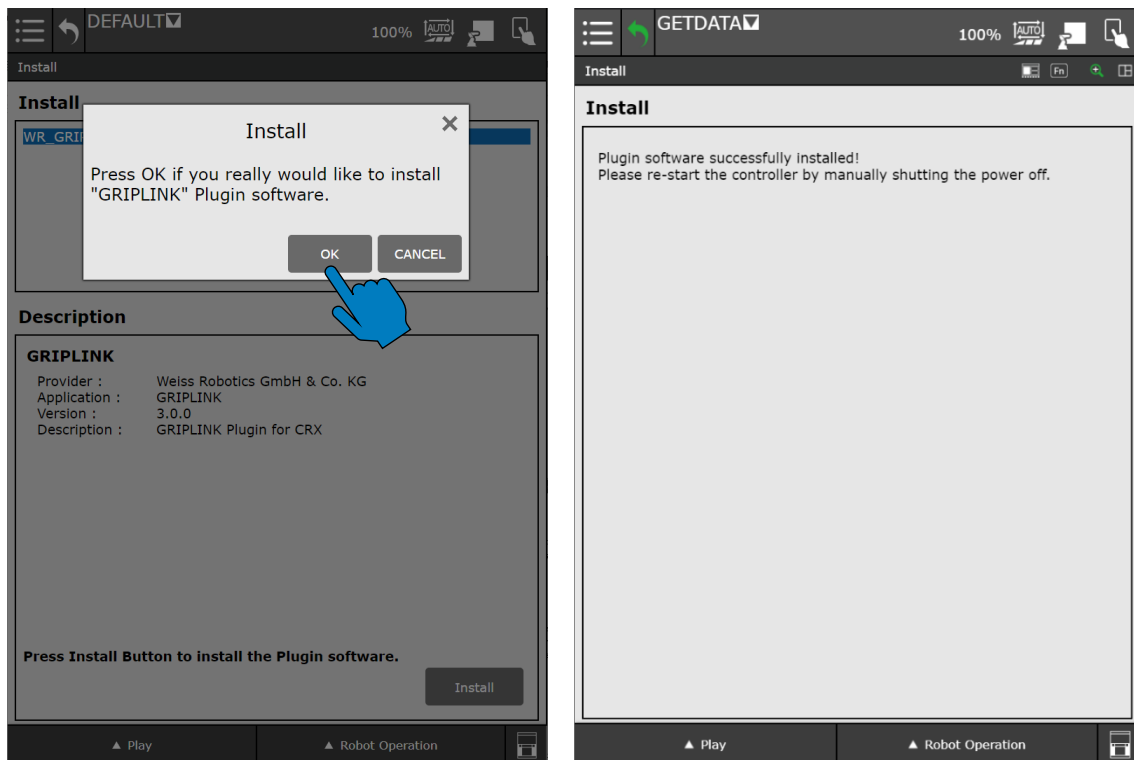
4.3 Installation des GRIPLINK CRX-Plugins

Für Roboter der Serie FANUC CRX empfehlen wir die Installation des CRX-spezifischen GRIPLINK-Plugins, das Instruktionsblöcke zur grafischen Programmierung zur Verfügung stellt und damit eine besonders intuitive Integration der GRIPLINK-Technologie in Ihre Anwendung ermöglicht.

1. Laden Sie die Plugin-Datei „griplink_fanuc_<Version>.zip“ herunter.
2. Entpacken Sie das ZIP-Archiv
3. Kopieren Sie die Datei **GRIPLINK.IPL** in das Stammverzeichnis eines USB-Sticks und stecken Sie diesen in den USB-Slot des Teach Pendants (UD1).

Für die Installation starten Sie zunächst den Roboter und schalten Sie das Tablet Teach Pendant ein. Warten Sie, bis die Tablet-TP-App geöffnet ist und sich das Teach Pendant mit der Steuerung verbunden hat. Führen Sie dann die nachfolgenden Schritte aus:





Nach der Aufforderung zum Neustarten der Robotersteuerung schalten Sie diese am Hauptschalter aus und nach etwa zehn Sekunden wieder ein oder nutzen Sie die Neustart-Funktion über den Teach Pendant (weitere Informationen hierzu entnehmen Sie der Anleitung der Robotersteuerung). Das Teach Pendant muss nicht neu gestartet werden. Warten Sie, bis sich das Teach Pendant erneut mit der Robotersteuerung verbunden hat.



Das CRX-Plugin enthält sämtliche Komponenten des Core-Plugins. Bei Verwendung des CRX-Plugins muss das Core-Plugin daher nicht separat installiert werden.

4.4 Deinstallieren der Software

Um das GRIPLINK-Plugin wieder von Ihrem Roboter zu entfernen, führen Sie die folgenden Schritte aus.

Deinstallation auf Standard-Robotern

das Deinstallations-Skript aus. Dieses ist ebenfalls Teil der Software-Distribution. Folgen Sie hierzu den Schritten in Kapitel 4.2 und wählen Sie anstelle des Installations-Skripts „SETUP.CM“ das Deinstallations-Skript „UNINSTALL.CM“ aus.

Deinstallation CRX-Plugin

Zur Deinstallation des CRX-Plugins nutzen Sie die grafische Bedienoberfläche am Tablet Teach Pendant. Navigieren Sie im Menu zum Eintrag „PLUGINS“ -> „Plugin List“. Wählen Sie das GRIPLINK-Plugin aus und klicken Sie auf „Uninstall“.

4.5 Konfiguration der Socket Messaging Schnittstelle

Zur Kommunikation zwischen Roboter und GRIPLINK-Controller wird die Socket Messaging Schnittstelle benötigt. Diese muss wie folgt konfiguriert werden:

4. Drücken Sie die „MENU“ Taste und navigieren Sie zu 6 (SETUP) → (Type) → 0 (NEXT) → 8 (HOST COMM). Es erscheint das Menü „SETUP Protocols“. Wählen Sie darin 3 (SM Socket Messaging Device) und drücken Sie den Softkey SHOW. Wählen Sie 2 (Clients).
5. Socket Messaging bietet Ihnen acht TAGs zur freien Verwendung an. Selektieren Sie das TAG, welches Sie zur Kommunikation mit dem GRIPLINK-Controller verwenden möchten (im Beispiel TAG 1).
6. Tragen Sie folgende Daten in die Eingabemaske ein:

Comment:	GRIPLINK (frei wählbar)
Protocol:	SM
Startup-State:	START
Server IP/Hostname:	IP-Adresse des GRIPLINK-Controllers (Werkseinstellung 192.168.1.40)
Remote Port:	10001



Die Felder „Remote Path/Share“, „Port“, „Username“ und „Password“ werden nicht benötigt und sollten leer gelassen werden.

7. Wählen Sie über den Softkey ACTION den Wert 1 (DEFINE) aus.
Im Feld Current State muss nun „DEFINED“ angezeigt werden
8. Wählen Sie über den Softkey ACTION den Wert 3 (START) aus.
Im Feld Current State muss nun „STARTED“ angezeigt werden

4.6 Verhalten im Fehlerfall

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

Ein Fehler kann beispielsweise auftreten, wenn die Verbindung zum GRIPLINK-Controller unterbrochen wird, ein ungültiger Parameter übergeben wurde oder ein angeschlossenes Gerät einen Fehlerzustand

meldet. Das gleiche Verhalten tritt ein, wenn sich das angesprochene Gerät bereits im Zustand **FAULT** befindet oder während der Ausführung eines Befehls in diesen Zustand wechselt.

Alle auftretenden Fehler werden zusätzlich im Logsystem der Robotersteuerung protokolliert. Die entsprechenden Meldungen können über das Teach Pendant aufgerufen werden. Navigieren Sie dazu über **MENU → 4 (ALARM) → 4 (Appl Log)**. In diesem Protokoll werden die vom Plugin erzeugten Fehlermeldungen zusammen mit weiteren Diagnoseinformationen gespeichert.

Einige Meldungen werden darüber hinaus ausführlicher im **USER-Log** der Robotersteuerung ausgegeben. Diese erweiterten Informationen können bei der Analyse von Kommunikationsproblemen oder unerwartetem Geräteverhalten hilfreich sein.

Für eine detailliertere Diagnose während der Inbetriebnahme oder Fehlersuche kann zusätzlich die TELNET-Ausgabe der Robotersteuerung genutzt werden. Hinweise zur Aktivierung und Verwendung dieser Ausgabe finden Sie in Abschnitt 7.1.

Eine Übersicht über typische Fehlermeldungen, deren Ursachen sowie mögliche Lösungsansätze finden Sie in Abschnitt 7 - Fehlersuche.

5 Befehlsreferenz (Core-Plugin)

Das GRIPLINK-Plugin stellt eine Reihe von Funktionen zur Ansteuerung und Überwachung von GRIPLINK-kompatiblen Geräten bereit. Über diese Befehle können Greifmodule aktiviert, gesteuert sowie deren Zustände und Prozesswerte abgefragt werden.

Die Ausführung der Befehle erfolgt über das Interfaceprogramm **GRIPLINK.PC**, welches in das Roboterprogramm eingebunden wird. Dabei wird der auszuführende Befehl zusammen mit den benötigten Parametern an das Interfaceprogramm übergeben. Die folgenden Abschnitte beschreiben die verfügbaren Befehle des GRIPLINK-Plugins sowie deren Parameter, Rückgabewerte und typische Anwendungsbeispiele.



Für die Nutzung des Plugins auf Steuerungen der CRX-Roboterserie sollten die im CRX-Plugin definierten Funktionsaufrufe (beginnend mit „IPL_WR_GRIPLINK“) verwendet werden.

Aufruf eines GRIPLINK-Befehls

Die GRIPLINK-Befehle werden über das Interfaceprogramm GRIPLINK.PC aufgerufen. Dabei wird der gewünschte Befehl zusammen mit den zugehörigen Parametern an das Interfaceprogramm übergeben.

```
CALL GRIPLINK(<Befehl>, <Parameter>)
```

Je nach Robotertyp und Softwarestand kann die Auswahl des Befehls sowie die Übergabe der Parameter über den Argument Wizard erfolgen.

Argument Wizard

Auf Steuerungen mit Softwarestand 9.10 oder höher (R-30iB plus Serie) können Befehl und Parameter direkt über den Argument Wizard ausgewählt werden.

Auf Steuerungen mit Softwarestand 8.30 (R-30iB Serie) kann der Befehl über den Argument Wizard ausgewählt werden. Die für den Befehl notwendigen Parameterwerte müssen über die Parameter A bis K eingetragen werden. Hierbei können nicht benötigte Parameter auf 0 gesetzt werden.

Rückgabewerte

Die Rückgabewerte der Befehle werden in ein Register gespeichert. Der Index dieses Registers bzw. der Startindex eines Registerbereichs kann über den Parameter „RETREGIDX“ (Return Register Index) vorgegeben werden.

Prinzipieller Programmablauf

1. Verbindung mit dem GRIPLINK-Controller herstellen mit **CONNECT**.
Dieser Befehl muss einmal zu Beginn des Programms ausgeführt werden.
2. Optional: Geräte initialisieren oder referenzieren.

Bei bestimmten Gerätetypen (z. B. Servogreifern ohne Absolutgeber) kann eine Referenzfahrt mit **HOME** erforderlich sein. Andere Gerätetypen, wie z. B. Sensoren, benötigen keine Referenzierung.

3. Geräte aktivieren mit **ENABLE** und konfigurieren, sofern dies vom jeweiligen Gerät unterstützt wird.
4. Gerätefunktionen ausführen, z. B. Greifen oder Freigeben bei Greifmodulen (GRIP, RELEASE) oder Zustände und Werte von Geräten abfragen.
5. Verbindung zum GRIPLINK-Controller trennen mit **BYE**, wenn die Schnittstelle physikalisch getrennt werden muss (z. B. bei einem Werkzeugwechsel) oder wenn das Programm beendet wird.

Um eine zuverlässige Kommunikation und die bestmögliche Leistung zu erreichen, sollte die Anwendung so umgesetzt werden, dass die Verbindung zum GRIPLINK-Controller während der gesamten Programmlaufzeit bestehen bleibt. In diesem Fall wird **CONNECT** einmal zu Beginn des Programms ausgeführt und **BYE** nur bei Programmende oder im Fehlerfall aufgerufen.

Beispiel: Minimalprogramm

```
! Connect to GRIPLINK controller
CALL GRIPLINK(CONNECT=1,COM_TAG=1)

! Enable gripper
CALL GRIPLINK(ENABLE=11,PORT=0,RETREGIDX=1)

! Grip part
CALL GRIPLINK(GRIP=20,PORT=0,PRESETIDX=1,DOBLOCK=1)

! Release part
CALL GRIPLINK(RELEASE=22,PORT=0,PRESETIDX=1,DOBLOCK=1)

! Disconnect
CALL GRIPLINK(BYE=2)
```

Im Folgenden sind die verfügbaren Befehle des GRIPLINK-Plugins beschrieben.

5.1 Verbindung aufbauen – CONNECT (1)

Dieser Befehl stellt die Verbindung zwischen GRIPLINK-Controller und der Robotersteuerung her. Als Übergabeparameter wird der Index des zuvor nach Kapitel 4.5 konfigurierten COM-Tags übergeben. Der Befehl wartet, bis die Verbindung hergestellt ist. Kann der GRIPLINK-Controller unter der angegebenen Adresse nicht erreicht werden, wird ein Fehler ausgelöst und die Programmausführung angehalten.

Der Befehl prüft weiterhin, ob der angeschlossene GRIPLINK-Controller die Mindestanforderungen des GRIPLINK Protokolls erfüllt (siehe Abschnitt 1.3).

Syntax

R-30iB plus:	GRIPLINK(CONNECT=1, <COM_TAG>)
R-30iB:	GRIPLINK(CONNECT=1, A=<COM_TAG>, B=0, C=0, D=0, E=0, F=0, G=0, H=0, I=0, J=0, K=0)

Parameter

<COM_TAG>	Index des COM-Tags (1 bis 8)
-----------	------------------------------

Rückgabewerte

keine

Häufige Fehlerursachen

COM-Tag nicht korrekt initialisiert

→ Einstellungen prüfen

GRIPLINK-Controller nicht mit Roboter verbunden

→ Netzwerkanschluss und Spannungsversorgung prüfen

GRIPLINK-Controller anders konfiguriert

→ Netzwerkeinstellungen des GRIPLINK-Controllers über die Weboberfläche prüfen

Protokollversion des GRIPLINK-Controllers passt nicht zum Plugin

→ Firmware des GRIPLINK-Controllers aktualisieren

Beispiel

Verbindung zwischen Roboter und dem GRIPLINK mit dem COM Tag 1 herstellen:

```
CALL GRIPLINK (CONNECT=1, COM_TAG=1)
```



Wenn GRIPLINK-Befehle vor einem CONNECT ausgeführt werden, löst dies einen Fehler aus. Daher muss der Befehl CONNECT vor jedem anderen GRIPLINK-Befehl aufgerufen werden.

5.2 Verbindung trennen – BYE (2)

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.



Das Trennen der Verbindung ist im normalen Programmablauf nicht erforderlich und sollte nur mit Bedacht verwendet werden.

Syntax

R-30iB plus:	GRIPLINK(BYE=2)
R-30iB:	GRIPLINK(BYE=2, A=0, B=0, C=0, D=0, E=0, F=0, G=0, H=0, I=0, J=0, K=0)

Parameter

keine

Rückgabewerte

keine

Beispiel

Verbindung zwischen Roboter und dem GRIPLINK-Controller trennen, wenn ein Werkzeugwechsel erforderlich ist:

```
CALL GRIPLINK (BYE=2)
```

! -> Change tool here!

```
CALL GRIPLINK (CONNECT=1, COM_TAG=1)
```

5.3 Angeschlossene Geräte prüfen – ASRTDEV (3)

Dieser Befehl prüft, ob am ausgewählten Port das vorgesehene Gerät angeschlossen ist. Besitzt das am Port angeschlossene Gerät nicht die angegebene Vendor- und Product-ID, so wird das Roboterprogramm angehalten.

Syntax

R-30iB plus:	GRIPLINK(ASRTDEV=3, <PORT>, <VID>, <PID>)
R-30iB:	GRIPLINK(ASRTDEV=3, A=<PORT>, B=<VID>, C=<PID>, D=0, E=0, F=0, G=0, H=0, I=0, J=0, K=0)

Parameter

<PORT>	Index des Geräts (0 bis 31)
<VID>	In der IODD des erwarteten Geräts spezifizierte Vendor-ID
<PID>	In der IODD des erwarteten Geräts spezifizierte Product-ID

Rückgabewerte

keine

Beispiel

Stelle sicher, dass an Port 3 das Gerät IEG 55-020 von Weiss Robotics (VID 815, PID 20) angeschlossen ist:

```
CALL GRIPLINK (ASRTDEV=3, PORT=3, VID=815, PID=20)
```



Die Vendor- und Product-ID entnehmen Sie der vom Hersteller zur Verfügung gestellten Gerätebeschreibung bzw. der IODD.

5.4 Gerät aktivieren – ENABLE (11)

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

Syntax

R-30iB plus:	GRIPLINK(ENABLE=11, <PORT>, <RETREGIDX>)
R-30iB:	GRIPLINK(ENABLE=11, A=<PORT>, B=<RETREGIDX>, C=0, D=0, E=0, F=0, G=0, H=0, I=0, J=0, K=0)

Parameter

<PORT>	Port-Index (0 bis 31)
<RETREGIDX>	Index des Rückgabewert-Registers (1 bis 200)

Rückgabewerte

R[<RETREGIDX>]	aktueller Gerätezustand
----------------	-------------------------

Beispiel

Aktiviere das Gerät an Port 0:

```
CALL GRIPLINK (ENABLE=11, PORT=0, RETREGIDX=1)
```



Das Verhalten kann je nach angeschlossenem Gerät unterschiedlich sein. Beachten Sie die Hinweise in den Anleitungen des jeweiligen Gerätetreibers!

5.5 Gerät deaktivieren – DISABLE (12)

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

Fehlerquittierung mittels DISABLE

Mit DISABLE können auch Gerätefehler quittiert werden, beispielsweise wenn das angeschlossene Gerät sich im FAULT-Zustand befindet.

Syntax

R-30iB plus:	GRIPLINK(DISABLE=12, <PORT>, <RETREGIDX>)
R-30iB:	GRIPLINK(DISABLE=12, A=<PORT>, B=<RETREGIDX>, C=0, D=0, E=0, F=0, G=0, H=0, I=0, J=0, K=0)

Parameter

<PORT>	Port-Index (0 bis 31)
<RETREGIDX>	Index des Rückgabewert-Registers (1 bis 200)

Rückgabewerte

R[<RETREGIDX>]	aktueller Gerätezustand
----------------	-------------------------

Beispiel

An Port 2 ist ein Servogreifer angeschlossen. Beim Systemstart soll vor dem Referenzieren auf einen etwaigen Fehlerzustand geprüft und dieser quittiert werden:

```
CALL GRIPLINK (STATE=10, PORT=2, RETREGIDX=1)

! Test for FAULT state
IF (R[1]=7) THEN

    ! Gripper in FAULT state (7)
    CALL GRIPLINK (DISABLE=12, PORT=2, RETREGIDX=1)
    WAIT 0.1 (sec)
    CALL GRIPLINK (ENABLE=11, PORT=2, RETREGIDX=1)
ENDIF
CALL GRIPLINK (HOME=13, PORT=2, RETREGIDX=1)
```

5.6 Referenzieren – HOME (13)

Referenziert das ausgewählte Gerät. Der Befehl führt die integrierte Referenzierungs-Funktion des Geräts aus und wartet, bis diese abgeschlossen ist.

Syntax

R-30iB plus:	GRIPLINK(HOME=13, <PORT>, <RETREGIDX>)
R-30iB:	GRIPLINK(HOME=13, A=<PORT>, B=<RETREGIDX>, C=0, D=0, E=0, F=0, G=0, H=0, I=0, J=0, K=0)

Parameter

<PORT>	Port-Index (0 bis 31)
<RETREGIDX>	Index des Rückgabewert-Registers (1 bis 200)

Rückgabewerte

R[<RETREGIDX>]	aktueller Gerätezustand
----------------	-------------------------

Beispiel

Referenziere das Greifmodul an Port 2:

```
CALL GRIPLINK (HOME=13, PORT=2, RETREGIDX=1)
```



Das Verhalten kann je nach angeschlossenem Gerät unterschiedlich sein. Beachten Sie die Hinweise in den Anleitungen des jeweiligen Gerätetreibers!

5.7 Werkstück über Preset greifen – GRIP (20)

Der Befehl GRIP führt einen Griff mit dem angegebenen Griff-Preset aus. Über den optionalen Parameter DOBLOCK kann festgelegt werden, ob der Befehl unmittelbar nach dem Start der Bewegung oder erst nach deren Abschluss zurückkehrt:

DOBLOCK = 0 → Der Befehl kehrt sofort nach Start der Bewegung zurück (nicht blockierend).

DOBLOCK = 1 → Der Befehl wartet bis der Greifvorgang abgeschlossen ist (blockierend).

Der aktuelle Zustand des Greifmoduls kann jederzeit mit dem STATE-Befehl abgefragt werden. Bei nicht blockierender Ausführung (DOBLOCK = 0) kann zusätzlich mit dem WSTR-Befehl auf eine Zustandsänderung gewartet werden, um das Ende des Greifvorgangs zu erkennen.



Die Greifparameter können über die Weboberfläche des GRIPLINK-Controllers konfiguriert werden.

Syntax

R-30iB plus:	GRIPLINK(GRIP=20, <PORT>, <PRESETIDX>, <DOBLOCK (opt)>)
R-30iB:	GRIPLINK(GRIP=20, A=<PORT>, B=<PRESETIDX>, C=<DOBLOCK (opt)>, D=0, E=0, F=0, G=0, H=0, I=0, J=0, K=0)

Parameter

<PORT>	Port-Index (0 bis 31)
<PRESETIDX>	Index des Griff-Presets (zulässiger Wertebereich ist geräteabhängig)
<DOBLOCK (opt)>	Optional. Wenn 0, kommt der Befehl direkt nach Start der Bewegung zurück. Wenn 1 wartet GRIPLINK, bis das Greifen abgeschlossen wurde. Default-Wert: 0

Rückgabewerte

keine

Beispiel

Greifmodul an Port 0 soll mit Griff 2 greifen. Wurde kein Werkstück gefunden, soll der Greifer warten, wieder öffnen und es erneut versuchen:

```
LBL[1]

! Grip
CALL GRIPLINK (GRIP=20, PORT=0, PRESETIDX=2)

! Wait until the gripper changed its state
CALL GRIPLINK (WSTR=80, PORT=0, RETREGIDX=1)
IF (R[1]=5) THEN
    ! Holding a part
    JMP LBL[2]
ENDIF
```

```

! Release and try again...
CALL GRIPLINK(RELEASE=22,PORT=0,PRESETIDX=2)
! Wait until the gripper changed its state
CALL GRIPLINK(WSTR=80,PORT=0,RETREGIDX=1)

WAIT 0.5(sec)
JMP LBL[1]
LBL[2]
! Workpiece gripped!

```

Das gleiche Beispiel mit blockierenden Befehlen:

```

LBL[1]

! Grip
CALL GRIPLINK(GRIP=20,PORT=0,PRESETIDX=2,DOBLOCK=1)

! Read device state state
CALL GRIPLINK(STATE=10,PORT=0,RETREGIDX=1)
IF (R[1]=5) THEN
    ! Holding a part
    JMP LBL[2]
ENDIF

! Release and try again...
CALL GRIPLINK(RELEASE=22,PORT=0,PRESETIDX=2,DOBLOCK=1)
WAIT 0.5(sec)
JMP LBL[1]

LBL[2]
! Workpiece gripped!

```

5.8 Werkstück über Preset freigeben – RELEASE (22)

Der Befehl RELEASE gibt ein zuvor gegriffenes Werkstück mit den Parametern des angegebenen Griff-Presets frei. Über den optionalen Parameter DOBLOCK kann gesteuert werden, ob der Befehl unmittelbar nach dem Start der Bewegung oder erst nach deren Abschluss zurückkehrt:

DOBLOCK = 0 → Der Befehl kehrt sofort nach Start der Bewegung zurück (nicht blockierend).

DOBLOCK = 1 → Der Befehl wartet bis die Freigabe abgeschlossen ist (blockierend).

Der aktuelle Zustand des Greifmoduls kann jederzeit mit dem STATE-Befehl abgefragt werden. Bei nicht blockierender Ausführung (DOBLOCK = 0) kann zusätzlich mit dem WSTR-Befehl auf eine Zustandsänderung gewartet werden, um das Ende der Freigabe zu erkennen.



Die Greifparameter können über die Weboberfläche des GRIPLINK-Controllers konfiguriert werden.

Syntax

R-30iB plus:	GRIPLINK(RELEASE=22, <PORT>, <PRESETIDX>, <DOBLOCK (opt)>)
R-30iB:	GRIPLINK(RELEASE=22, A=<PORT>, B=<PRESETIDX>, C=<DOBLOCK (opt)>, D=0, E=0, F=0, G=0, H=0, I=0, J=0, K=0)

Parameter

<PORT>	Port-Index (0 bis 31)
<PRESETIDX>	Index des Griff-Presets (zulässiger Wertebereich ist geräteabhängig)
<DOBLOCK (opt)>	Optional. Wenn 0, kommt der Befehl direkt nach Start der Bewegung zurück. Wenn 1 wartet GRIPLINK, bis die Freigabebewegung abgeschlossen wurde. Default-Wert: 0

Rückgabewerte

keine

Beispiel

Siehe Beispiel in Kapitel 5.7

5.9 Werkstück über Parameter greifen – FLEXGRIP (26)

Der Befehl FLEXGRIP führt einen Greifvorgang mit frei wählbaren Bewegungsparametern aus.

Im Gegensatz zum GRIP-Befehl wird hier keine vordefinierte Griffkonfiguration (Preset) verwendet, sondern der Bewegungsablauf vollständig über die Befehlsparameter bestimmt.

Damit kann der Anwender individuelle Zielpositionen, Kräfte, Geschwindigkeiten und Beschleunigungen vorgeben, ohne auf ein gespeichertes Griff-Preset zurückgreifen zu müssen.

Der Befehl kann wahlweise blockierend oder nicht blockierend ausgeführt werden.

Bei blockierender Ausführung kehrt der Befehl erst nach Abschluss des Greifvorgangs zurück.

Nach Aufruf des Befehls wird der Greifvorgang mit den angegebenen Bewegungsparametern gestartet. Wird der Parameter DOBLOCK = 0 gesetzt, kehrt der Befehl sofort zurück, während die Bewegung im Hintergrund ausgeführt wird. Bei DOBLOCK = 1 wartet der Befehl, bis der Vorgang abgeschlossen ist (Werkstück gegriffen oder kein Werkstück erkannt).

Der aktuelle Zustand kann jederzeit mit dem Befehl STATE abgefragt werden. Alternativ kann der Abschluss des Befehls mit WSTR erkannt werden, wenn der Befehl nicht blockierend ausgeführt wurde.

Syntax

R-30iB plus:	GRIPLINK(FLEXGRIP=26, <PORT>, <POSITION>, <FORCE>, <SPEED>, <ACCELERATION>, <DOBLOCK>)
R-30iB:	GRIPLINK(FLEXGRIP=26, A=<PORT>, B=<POSITION>, C=<FORCE>, D=<SPEED>, E=<ACCELERATION>, F=<DOBLOCK>, G=0, H=0, I=0, J=0, K=0)

Parameter

<PORT>	Port-Index (0 – 31)
<POSITION>	Zielposition für „No Part Limit“ in mm
<FORCE>	Greifkraft in N
<SPEED>	Greifgeschwindigkeit in mm/s (0 = nutze automatische Berechnung)
<ACCELERATION>	Fingerbeschleunigung in mm/s ² (0 = nutze Standardwert)
<DOBLOCK>	0 = nicht blockierend, 1 = blockierend

Beispiel

Greifen eines Werkstücks mit dem Greifer an Port 0 und 60 N Kraft, No-Part-Limit = 15 mm, Geschwindigkeit = 45,5 mm/s, Standardbeschleunigung, nicht blockierend:

```
CALL GRIPLINK(FLEXGRIP=26, PORT=0, POSITION=15.0, FORCE=60.0,
SPEED=45.5, ACCELERATION=0, DOBLOCK=0)
```

Blockierende Variante:

```
CALL GRIPLINK(FLEXGRIP=26, PORT=0, POSITION=15.0, FORCE=60.0,
SPEED=45.5, ACCELERATION=0, DOBLOCK=1)
```

5.10 Werkstück über Parameter freigeben – FLEXRELEASE (27)

Der Befehl FLEXRELEASE gibt ein zuvor gegriffenes Werkstück mit frei wählbaren Bewegungsparametern aus. Er kann sowohl für eine einfache Freigabe als auch für eine gezielte Vorpositionierung eines Servogreifers verwendet werden. Die Bewegungsparameter werden vollständig über die Befehlsargumente definiert.

Nach Aufruf des Befehls öffnet das Greifmodul die Finger bis zur angegebenen Zielposition. Bei nicht blockierender Ausführung kehrt der Befehl sofort zurück, während die Bewegung im Hintergrund läuft. Bei blockierender Ausführung wartet der Befehl bis zum Abschluss der Bewegung.

Der Fortschritt kann über den Befehl STATE oder über den Abfragebefehl WSTR überwacht werden.

Der Zustand ändert sich nach Abschluss typischerweise zu DS_RELEASED.

Syntax

R-30iB plus:	GRIPLINK(FLEXRELEASE=27, <PORT>, <POSITION>, <SPEED>, <ACCELERATION>, <DOBLOCK>)
R-30iB:	GRIPLINK(FLEXRELEASE=27, A=<PORT>, B=<POSITION>, C=<SPEED>, D=<ACCELERATION>, E=<DOBLOCK>, F=0, G=0, H=0, I=0, J=0, K=0)

Parameter

<PORT>	Port-Index (0 – 31)
<POSITION>	Zielposition (Release-Limit) in mm
<SPEED>	Öffnungsgeschwindigkeit in mm/s (0 = nutze Standardwert)
<ACCELERATION>	Fingerbeschleunigung in mm/s ² (0 = nutze Standardwert)
<DOBLOCK>	0 = nicht blockierend, 1 = blockierend

Beispiel

Vorpositionierung des Greifers an Port 0 auf 19,5 mm mit Standardparametern, blockierend:

```
CALL GRIPLINK(FLEXRELEASE=27, PORT=0, POSITION=19.5, SPEED=0,  
ACCELERATION=0, DOBLOCK=1)
```

Nicht blockierend:

```
CALL GRIPLINK(FLEXRELEASE=27, PORT=0, POSITION=19.5, SPEED=0,  
ACCELERATION=0, DOBLOCK=0)
```

5.11 Parametrierung eines Griff-Presets – SGRIPCFG (60)

Der Befehl SGRIPCFG 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 konfiguriert werden. Auch wenn ein Gerät weniger als acht konfigurierbare Parameter unterstützt, müssen immer alle acht Werte übertragen werden.



Die Bedeutung und zulässigen Wertebereiche dieser Parameter sind gerätespezifisch und müssen der jeweiligen Geräte- oder Treiberdokumentation entnommen werden.

Syntax

R-30iB plus: GRIPLINK(SGRIPCFG=60, <PORT>, <PRESETIDX>, <PRM0>, <PRM1>, <PRM2>, <PRM3>, <PRM4>, <PRM5>, <PRM6>, <PRM7>)

R-30iB: GRIPLINK(SGRIPCFG=60, A=<PORT>, B=<PRESETIDX>, C=<PRM0>, D=<PRM1>, E=<PRM2>, F=<PRM3>, G=<PRM4>, H=<PRM5>, I=<PRM6>, J=<PRM7>, K=0)

Parameter allgemein

<PORT>	Port-Index (0 bis 31)
<PRESETIDX>	Index des Griff-Presets (zulässiger Wertebereich ist geräteabhängig)
<PRM0..7>	Gerätespezifische Preset-Parameter 0 bis 7. Die Parameter sind optional: werden nicht alle Parameter angegeben, werden diese intern auf 0 gesetzt.

Beispiel: Parameter für Greifmodule von Weiss Robotics über GRIPLINK-ET4

<PORT>	Port-Index (0 bis 31)
<PRESETIDX>	Index des Griff-Presets (zulässiger Wertebereich ist geräteabhängig)
<PRM0>	No Part Limit (Wertebereich hängt von Greifermodell ab, in mm)
<PRM1>	Release Limit (Wertebereich hängt von Greifermodell ab, in mm)
<PRM2>	Force Factor (Wertebereich hängt von Greifermodell ab, in % [0..100])
<PRM3..7>	Nicht benutzt

Rückgabewerte

keine

Beispiel

Setze beim IEG PLUS Greifmodul an Port 3 des GRIPLINK-ET4 Controllers die Parameter des Griffs mit dem Index 1 auf:

No Part Limit = 31 mm
Release Limit = 41 mm
Force Factor = 59 %

```
CALL GRIPLINK (SGRIPCFG=60,PORT=3,PRESETIDX=1,PRM0=31.0,PRM1=41.0,  
PRM2=59.0)
```

5.12 Aktuellen Gerätezustand abfragen – STATE (10)

Dieser Befehl gibt den Gerätezustand des ausgewählten Geräts zurück.

Syntax

R-30iB plus:	GRIPLINK(STATE=10, <PORT>, <RETREGIDX>)
R-30iB:	GRIPLINK(STATE=10, A=<PORT>, B=<RETREGIDX>, C=0, D=0, E=0, F=0, G=0, H=0, I=0, J=0, K=0)

Parameter

<PORT>	Port-Index (0 bis 31)
<RETREGIDX>	Index des Rückgabewert-Registers (1 bis 200)

Rückgabewerte

R[<RETREGIDX>]	aktueller Gerätezustand
----------------	-------------------------

Beispiel

Warte, bis der Gerätezustand von Greifmodul an Port 2 „HOLDING“ (4) ist:

```
LBL[1]
```

```
! Get device state
```

```
CALL GRIPLINK(STATE=10, PORT=2, RETREGIDX=1)
```

```
IF R[1]=5, JMP LBL[2]
```

```
WAIT 0.01(sec)
```

```
! Do some timeout processing, e.g. using a counter variable
```

```
JMP LBL[1]
```

```
...
```

```
LBL[2]
```

```
! Gripper at port 2 is in HOLDING state
```

5.13 Warten auf eine Änderung des Gerätezustands – WSTR (80)

Der Befehl WSTR wartet auf eine Änderung des Gerätezustands am angegebenen Port. Er kann beispielsweise verwendet werden, um zu prüfen, ob eine asynchron ausgeführte Freigabebewegung (RELEASE mit DOBLOCK=0) abgeschlossen ist.



Tritt die Zustandsänderung nicht in einem vorgegebenen Zeitintervall ein, wird ein Timeout-Fehler ausgelöst und die Programmausführung unterbrochen.

Syntax

R-30iB plus: GRIPLINK(WSTR=80, <PORT>, <RETREGIDX>)
R-30iB: GRIPLINK(WSTR=80, A=<PORT>, B=<RETREGIDX>, C=0, D=0, E=0, F=0,
 G=0, H=0, I=0, J=0, K=0)

Parameter

<PORT> Port-Index (0 bis 31)
<RETREGIDX> Index des Rückgabewert-Registers (1 bis 200)

Rückgabewerte

R[<RETREGIDX>] aktueller Gerätezustand

Beispiel

Positioniere die Greiffinger mit Preset 2 vor, während der Roboter über die Aufnahmeposition P[1] fährt. An dieser angekommen, wird gewartet, bis die Finger geöffnet sind, um dann an die Aufnahme-position P[2] zu fahren, um das Werkstück zu greifen.

```
! Start pre-positioning the fingers
CALL GRIPLINK(RELEASE=22, PORT=0, PRESETIDX=2, DOBLOCK=0)

! Move to Approach point
J P[1]...

! Wait until the gripper changed its state
CALL GRIPLINK(WSTR=80, PORT=0, RETREGIDX=1)
IF (R[1]<>3) THEN
  ! Not in RELEASED (3) state. Handle ERROR here!
ENDIF

! Move to Stack point
J P[2]...

! Grip workpiece
CALL GRIPLINK(GRIP=20, PORT=0, PRESETIDX=2, DOBLOCK=1)
```

5.14 Gerätewerte abfragen – VALUE (41)

Dieser Befehl liest einen indizierten Gerätewert aus.



Der Rückgabewert wird in das Register mit dem angegebenen Index geschrieben.



Der zulässige Wertebereich für den Value-Index ist gerätespezifisch und kann der Treiber- oder Gerätedokumentation des jeweiligen Geräts entnommen werden.

Syntax

R-30iB plus:	GRIPLINK(VALUE=41, <PORT>, <VALUEIDX>, <RETREGIDX>)
R-30iB:	GRIPLINK(VALUE=41, A=<PORT>, B=<VALUEIDX>, C=<RETREGIDX>, D=0, E=0, F=0, G=0, H=0, I=0, J=0, K=0)

Parameter

<PORT>	Port-Index (0 bis 31)
<VALUEIDX>	Index des auszulesenden Wertes (zulässiger Wertebereich ist geräteabhängig)
<RETREGIDX>	Index des Rückgabewert-Registers (1 bis 200)

Rückgabewerte

R[<RETREGIDX>]	Aktueller Gerätewert
----------------	----------------------

Beispiel

Lese vom Gerät an Port 3, einem Servogreifer, den Primärwert (Value-Index 0, hier: Fingerposition) aus und speichere den Wert in das Register 31. Nutze diesen für eine IO/NIO-Prüfung.

```
! Grip workpiece
CALL GRIPLINK (GRIP=20, PORT=3, PRESETIDX=0, DOBLOCK=1)

! Get part width
CALL GRIPLINK (VALUE=41, PORT=3, VALUEIDX=0, RETREGIDX=31)

! Is part width 30+/-0.1?
IF (R[31]<29.9 OR R[31]>30.1) THEN

    ! Part is NIO -> Move to Drop position
    J P[99]
    CALL GRIPLINK (RELEASE=22, PORT=3, PRESETIDX=0, DOBLOCK=1)
ELSE

    ! Part is IO -> Move to Use position
    J P[20]
    CALL GRIPLINK (RELEASE=22, PORT=3, PRESETIDX=0, DOBLOCK=1)
ENDIF
```

5.15 Gerätewerte setzen – SETVAL (42)

Mit diesem Befehl kann ein gerätespezifischer Wert gesetzt werden. Der Wert ist über einen Index auswählbar, welcher einen vom Gerät abhängigen gültigen Wertebereich hat. Der zu schreibende Wert kann je nach angeschlossenem Gerät verschiedene Bedeutungen haben.



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

Syntax

R-30iB plus: GRIPLINK(SETVAL=42, <PORT>, <VALUEIDX>, <VALUE>)
R-30iB: GRIPLINK(SETVAL=42, A=<PORT>, B=<VALUEIDX>, C=<VALUE>, D=0,
 E=0, F=0, G=0, H=0, I=0, J=0, K=0)

Parameter

<PORT>	Port-Index (0 bis 31)
<VALUEIDX>	Index des zu schreibenden Werts (zulässiger Wertebereich ist geräteabhängig)
<VALUE>	Zu schreibender Wert (zulässiger Wertebereich ist geräteabhängig)

Rückgabewerte

keine

Beispiel

Setze im Gerät an Port 0 den Wert mit dem Index 2 auf 42.5 und warte, bis der Wert im Bereich ± 0.1 liegt, jedoch warte darauf maximal für 10 Sekunden:

```
CALL GRIPLINK (SETVAL=42, PORT=0, VALUEIDX=2, VALUE=42.5)
```

```
! Wait for value reached the set value
```

```
CALL GRIPLINK (WAITVAL=43, PORT=0, VALUEIDX=2, THRESHOLD=42.5,  
WINDOWSIZE=0.1, TIMEOUT=10000)
```

5.16 Warten auf einen Gerätewert – WAITVAL (43)

Der Befehl WAITVAL wartet, bis ein bestimmter Gerätewert den zuvor gesetzten Sollwert erreicht oder innerhalb eines definierten Toleranzbereichs liegt. Er kann beispielsweise verwendet werden, um zu prüfen, ob ein Sensor- oder Positionswert einen Zielbereich erreicht hat, bevor das Programm fortgesetzt wird.

Der Befehl blockiert, bis eine der folgenden Bedingungen erfüllt ist:

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



Der Befehl WAITVAL ist mit der GRIPLINK-Protokollversion 3 kompatibel. Der überwachte Wert entspricht dem über VALUE auslesbaren Indexwert.



Erreicht der zu überwachende Wert das Detektionsfenster nicht im vorgegebenen TIMEOUT-Zeitintervall, wird ein Timeout-Fehler ausgelöst und die Programmausführung unterbrochen.

Syntax

R-30iB plus:	GRIPLINK(WAITVAL=43, <PORT>, <VALUEIDX>, <THRESHOLD>, <WINDOWSIZE>, <TIMEOUT>)
R-30iB:	GRIPLINK(WAITVAL=43, A=<PORT>, B=<VALUEIDX>, C=<THRESHOLD>, D=<WINDOWSIZE>, E=<TIMEOUT>, F=0, G=0, H=0, I=0, J=0, K=0)

Parameter

<PORT>	Port-Index (0 bis 31)
<VALUEIDX>	Index des zu überwachenden Werts (zulässiger Wertebereich ist geräteabhängig)
<THRESHOLD>	Zielwert in der Einheit des Messwerts
<WINDOWSIZE>	Breite des Toleranzfensters symmetrisch zum Zielwert
<TIMEOUT>	Wert für das Timeout in Millisekunden

Rückgabewerte

keine

Beispiel

Siehe Beispiel in Kapitel 5.15

5.17 Greifkraftherhaltung steuern – CLAMP (24)

Mit dem CLAMP Befehl können Greifmodule mit integrierter Greifkraftsicherung im Zustand HOLDING den Antrieb stromlos schalten, um den Strombedarf des Greifmoduls zu reduzieren. Das Werkstück wird weiterhin sicher gehalten.

Wird CLAMP aktiviert, schaltet sich der Antrieb im Zustand HOLDING ab. Wird CLAMP deaktiviert, bleibt im Zustand HOLDING die Greifkraftregelung aktiv und die Steuerung des Greifmoduls regelt die Greifkraft permanent nach.



Der CLAMP Befehl ist nicht bei allen Greifmodulen verfügbar.

Syntax

R-30iB plus:	GRIPLINK(CLAMP=24, <PORT>, <ENABLE>)
R-30iB:	GRIPLINK(CLAMP=24, A=<PORT>, B=<ENABLE>, C=0, D=0, E=0, F=0, G=0, H=0, I=0, J=0, K=0)

Parameter

<PORT>	Port-Index (0 bis 3)
<ENABLE>	Greifkraftherhaltung: 1 = aktiviert, 0 = deaktiviert

Rückgabewerte

keine

Beispiel

Aktiviert die Greifkraftherhaltung im Greifmodul an Port 2:

```
CALL GRIPLINK (CLAMP=24, PORT=2, ENABLE=1)
```

5.18 Ansteuerung der Status-LED – LED (25)

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.



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



Der LED-Befehl wird nicht von allen Geräten unterstützt.

Syntax

R-30iB plus:	GRIPLINK(LED=25, <PORT>, <PRESETIDX>)
R-30iB:	GRIPLINK(LED=25, A=<PORT>, B=<PRESETIDX>, C=0, D=0, E=0, F=0, G=0, H=0, I=0, J=0, K=0)

Parameter

<PORT>	Port-Index (0 bis 31)
<PRESETIDX>	Index des LED-Presets (0 bis 7)

Rückgabewerte

keine

Beispiel

Führe mit dem Greifmodul an Port 3 einen Greifbefehl aus und lese die Greifposition aus. Ändere die Farbe des Leuchtrings auf LED-Preset 1, wenn die Fingerposition danach größer gleich 8,1 mm und auf LED-Preset 2, wenn kleiner:

```
! Grip with gripper at port 3 and preset 0, wait for state transition
CALL GRIPLINK (GRIP=20, PORT=3, PRESETIDX=0, DOBLOCK=0)
CALL GRIPLINK (WSTR=80, PORT=3, RETREGIDX=1)

! Read finger position and update LED
CALL GRIPLINK (VALUE=41, PORT=3, VALUEIDX=0, RETREGIDX=2)
IF (R[2]>=8.1) THEN
  CALL GRIPLINK (LED=25, PORT=3, PRESETIDX=1)
ELSE
  CALL GRIPLINK (LED=25, PORT=3, PRESETIDX=2)
ENDIF
```

5.19 Veraltete Befehle

In diesem Abschnitt sind Befehle aufgeführt, die aus Gründen der Abwärtskompatibilität weiterhin im GRIPLINK-Plugin enthalten sind, deren Verwendung jedoch nicht mehr empfohlen wird.

Diese Befehle wurden durch funktional gleichwertige, jedoch flexiblere Mechanismen ersetzt. Insbesondere ermöglichen die neueren Befehle eine feinere Steuerung des Programmablaufs sowie eine bessere Anpassung an unterschiedliche Anwendungen und Gerätetypen.

Für neue Programme sollte daher auf die Verwendung dieser veralteten Befehle verzichtet werden. Stattdessen wird empfohlen, die entsprechenden aktuellen Befehle und Programmiermuster zu verwenden.

5.19.1 Gleichzeitiges Greifen von Werkstücken – MGRIP (21)

Dieser Befehl führt mit den ausgewählten Greifmodulen einen simultanen Greifbefehl aus. Mit dem MWAITFOR -Befehl kann nachfolgend auf eine Zustandsänderung der Greifmodule gewartet werden.



Dieser Befehl ist veraltet und wird nur aus Kompatibilitätsgründen bereitgestellt. Für neue Anwendungen wird empfohlen, stattdessen GRIP mit DOBLOCK = 0 in Kombination mit WSTR zu verwenden.

Syntax

R-30iB plus:	CALL GRIPLINK(MGRIP=21, <PORTS>, <PRESETIDX>)
R-30iB:	CALL GRIPLINK(MGRIP=21, A=<PORTS>, B=<PRESETIDX>, C=0, D=0, E=0, F=0, G=0, H=0, I=0, J=0, K=0)

Parameter

<PORTS>	Ausgewählte Greifmodule als 32-Bit-Vektor: Bit x = 1: Gerät an Port (1 <<x>) ist selektiert Bit x = 0: Gerät an Port (1 <<x>) ist nicht selektiert Beispiel: Greifer an Ports 0 und 3 selektieren → Bit 0 und 3 setzen → Ports = (1 <<0>) + (1 <<3>) = 1 + 8 = 9
<PRESETIDX>	Index des Griff-Presets (zulässiger Wertebereich ist geräteabhängig)

Rückgabewerte

keine

Beispiel

Greifmodule an Port 0 und 3 greifen Werkstück mit Griff-Preset 2 und anschließend auf Zustandsänderungen warten:

! Grip with grippers at ports 0 and 3

```
CALL GRIPLINK(MGRIP=21,PORTS=9,PRESET=2)
CALL GRIPLINK(MWAITFOR=81,PORTS=9)
! Get states of the grippers
CALL GRIPLINK(STATE=10,PORT=0,RETREGIDX=1)
CALL GRIPLINK(STATE=10,PORT=3,RETREGIDX=4)
...
! Release with grippers at ports 0 and 3
CALL GRIPLINK(MRELEASE=23,PORTS=9,PRESET=2)
CALL GRIPLINK(MWAITFOR=81,PORTS=9)
! Get actual states of the grippers
CALL GRIPLINK(STATE=10,PORT=0,RETREGIDX=1)
CALL GRIPLINK(STATE=10,PORT=3,RETREGIDX=4)
```

5.19.2 Gleichzeitiges Freigeben von Werkstücken – MRELEASE (23)

Dieser Befehl führt mit den ausgewählten Greifmodulen einen simultanen Freigabebefehl aus. Mit dem MWAITFOR-Befehl kann nachfolgend auf eine Zustandsänderung der Greifmodule gewartet werden.



Dieser Befehl ist veraltet und wird nur aus Kompatibilitätsgründen bereitgestellt. Für neue Anwendungen wird empfohlen, stattdessen RELEASE mit DOBLOCK = 0 in Kombination mit WSTR zu verwenden.

Syntax

R-30iB plus: CALL GRIPLINK(MRELEASE=23, <PORTS>, <PRESETIDX>)
R-30iB: CALL GRIPLINK(MRELEASE=23, A=<PORTS>, B=<PRESETIDX>, C=0,
 D=0, E=0, F=0, G=0, H=0, I=0, J=0, K=0)

Parameter

<PORTS> Ausgewählte Greifmodule als 32-Bit-Vektor:
 Bit x = 1: Gerät an Port (1 <<x>) ist selektiert
 Bit x = 0: Gerät an Port (1 <<x>) ist nicht selektiert

Beispiel: Greifer an Ports 0 und 3 selektieren

→ Bit 0 und 3 setzen

→ Ports = (1 <<0>) + (1 <<3>) = 1 + 8 = 9

<PRESETIDX> Index des Griff-Presets (zulässiger Wertebereich ist geräteabhängig)

Rückgabewerte

keine

Beispiel

Siehe Beispiel in 5.19.1

5.19.3 Warten auf die Änderung mehrerer Gerätezustände – MWAITFOR (81)

Der Befehl MWAITFOR wartet darauf, dass mehrere Geräte ihren Zustand ändern.



Tritt die Zustandsänderung bei mindestens einem ausgewählten Port nicht in einem vorgegebenen Zeitintervall ein, wird ein Timeout-Fehler ausgelöst und die Programmausführung unterbrochen.



Dieser Befehl ist veraltet und wird nur aus Kompatibilitätsgründen bereitgestellt. Für neue Anwendungen wird empfohlen, stattdessen WSTR zu verwenden.

Syntax

R-30iB plus:	GRIPLINK(MWAITFOR=81, <PORTS>)
R-30iB:	GRIPLINK(MWAITFOR=81, A=<PORTS>, B=0, C=0, D=0, E=0, F=0, G=0, H=0, I=0, J=0, K=0)

Parameter

<PORTS> Ausgewählte Greifmodule als 32-Bit-Vektor:
Bit x = 1: Gerät an Port (1 <<x>) ist selektiert
Bit x = 0: Gerät an Port (1 <<x>) ist nicht selektiert

Beispiel: Greifer an Ports 0 und 3 selektieren
→ Bit 0 und 3 setzen
→ Ports = (1 <<0>) + (1 <<3>) = 1 + 8 = 9

Rückgabewerte

keine

Beispiel

Siehe Beispiel in 5.19.1

5.19.4 Aktuelle Fingerposition auslesen – POS (40)

Dieser Befehl liest die aktuelle Fingerposition des ausgewählten Greifmoduls zurück und speichert diese in das Register mit dem gegebenen Index.



Dieser Befehl ist veraltet. Verwenden Sie stattdessen VALUE mit VALUEIDX = 0.

Syntax

R-30iB plus:	CALL GRIPLINK(POS=40, <PORT>, <RETREGIDX>)
R-30iB:	CALL GRIPLINK(POS=40,A=<PORT>, B=<RETREGIDX>, C=0, D=0, E=0, F=0, G=0, H=0, I=0, J=0, K=0)

Parameter

<PORT>	Index des Greifmoduls (0 bis 31)
<RETREGIDX>	Index des Rückgabewert-Registers (1 bis 100)

Rückgabewerte

R[<RETREGIDX>]	Fingerposition in mm
----------------	----------------------

Beispiel

Führe das Unterprogramm „PROG“ aus, wenn die Fingerposition des Greifmoduls an Port 2 größer als 10,5 mm ist:

```
CALL GRIPLINK(POS=40, PORT=2, RETREGIDX=42)
IF R[42]>10.5, CALL PROG
```

6 Grafische Programmierung mit dem FANUC CRX-Plugin

Für Roboter der FANUC CRX-Serie steht zusätzlich zum Core-Plugin ein erweitertes Plugin zur Verfügung, das eine grafische (instruktionsbasierte) Programmierung von GRIPLINK-Funktionen ermöglicht. Das Plugin stellt die GRIPLINK-Befehle als Instruktionsblöcke im Programmeditor des Tablet Teach Pendants bereit. Diese können per Drag & Drop in ein Roboterprogramm eingefügt und anschließend über eine Detailansicht parametrisiert werden.

Dadurch lassen sich GRIPLINK-kompatible Geräte schnell und übersichtlich in Roboterprogramme integrieren, ohne dass die zugrunde liegenden Funktionsaufrufe des Core-Plugins manuell programmiert werden müssen.

Die folgenden Abschnitte beschreiben die Einrichtung des Plugins, die Hardware-Inbetriebnahme sowie die Verwendung der verfügbaren Instruktionen im Roboterprogramm.

6.1 Funktionsweise

Das CRX-Plugin besteht aus einer Konfigurationsseite sowie einer Reihe von Programminstruktionen, über die die Robotersteuerung mit einem angeschlossenen GRIPLINK-Gerät interagiert.

Über die Konfigurationsseite werden grundlegende Einstellungen des Plugins vorgenommen, beispielsweise die Auswahl des verwendeten COM-Tags für die Kommunikation mit dem GRIPLINK-Controller.

Instruktionen stellen die Befehle des Core-Plugins in Form grafischer Funktionsbausteine bereit und dienen als Konfigurations- und Ausführungsoberfläche im Programmeditor.

Die Rückgabewerte der Instruktionen werden grundsätzlich in Numeric Registers (im Folgenden „Register“ genannt) der Robotersteuerung gespeichert. Diese Register können anschließend im Roboterprogramm ausgewertet werden, beispielsweise zur Zustandsüberwachung oder zur Prozesssteuerung. Der Registerindex kann für jeden Befehl individuell festgelegt werden. Alternativ kann über die Plugin-Konfiguration ein Standardwert für den Registerindex definiert werden.

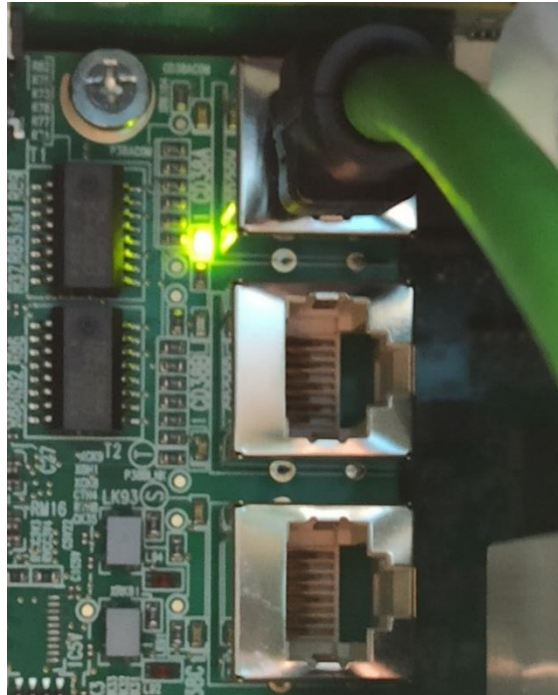
Das CRX-Plugin nutzt intern die Funktionen des Core-Plugins, welches die eigentliche Kommunikation mit dem GRIPLINK-Controller übernimmt. Die grafischen Instruktionen dienen somit als komfortable Benutzeroberfläche für die im Core-Plugin implementierten Befehle. Anwendungen können daher sowohl grafisch über Instruktionen als auch textuell über Funktionsaufrufe erstellt werden. Beide Varianten greifen auf dieselbe Kommunikationsschnittstelle zu.

6.2 Hardware-Inbetriebnahme



TRENNEN SIE DIE ROBOTERSTEUERUNG VOR DEM ÖFFNEN UNBEDINGT VOM NETZ!
Verletzungs- und Beschädigungsgefahr!

Verbinden Sie den GRIPLINK-Controller mit dem oberen Netzwerkanschluss in Ihrer Robotersteuerung wie im folgenden Bild dargestellt.

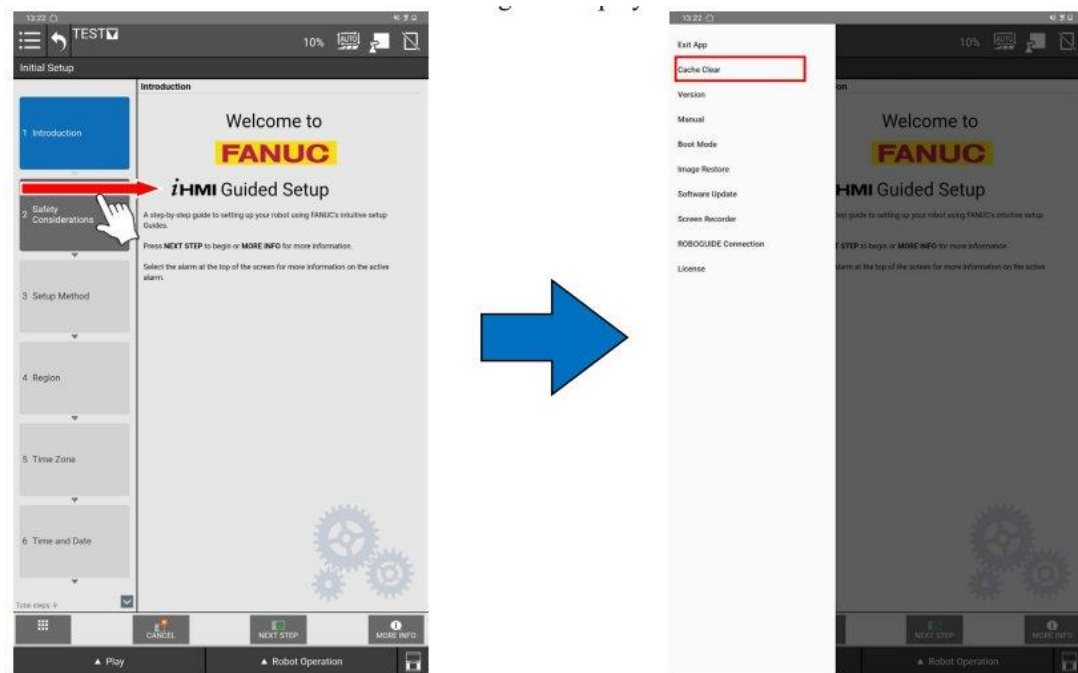


6.3 Vorbereitung des Plugins



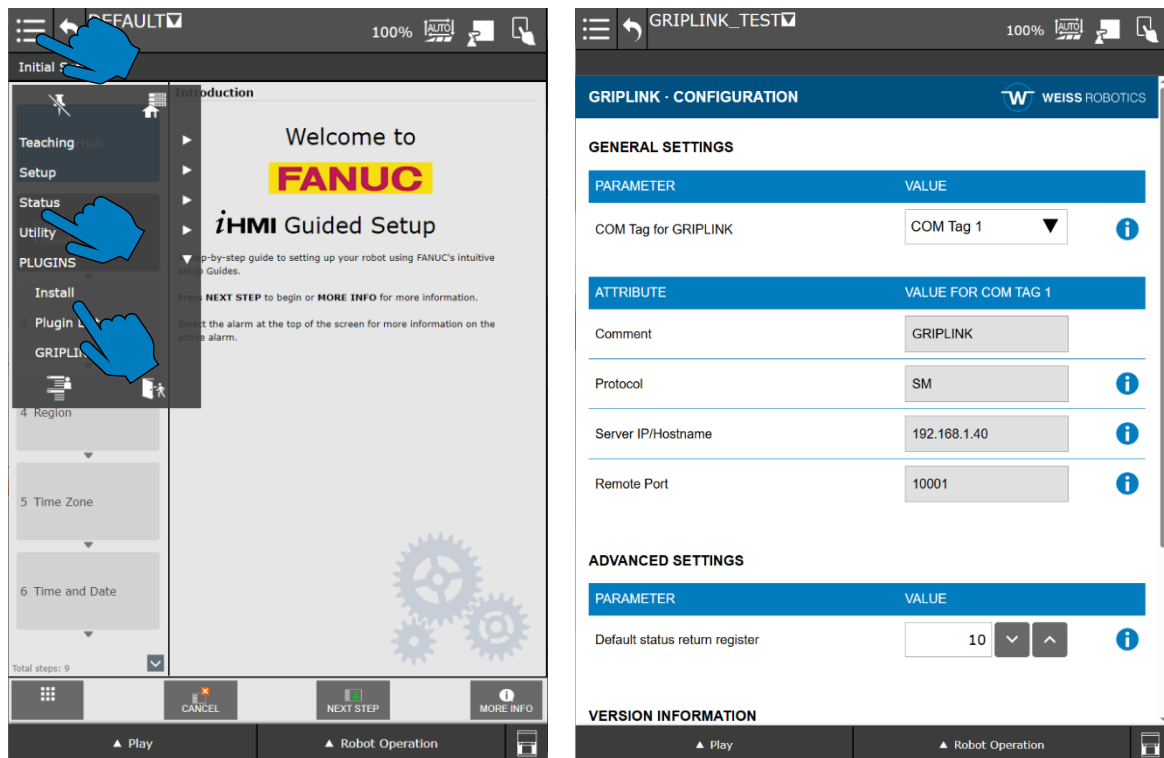
Vor der Einrichtung des Plugins wird empfohlen, den Cache des Tablet Teach Pendants zu leeren. Auf diese Weise lassen sich Anzeigefehler auf der Konfigurationsseite sowie bei den grafischen Instruktionen vermeiden, die durch zwischengespeicherte Daten verursacht werden können.

Um den Cache zu löschen, wischen Sie vom linken Bildschirmrand nach rechts, um das Menü zu öffnen, und wählen Sie „Cache Clear“. Bestätigen Sie den Dialog „Clear the cache and exit the Tablet TP“ mit „OK“. Das Tablet Teach Pendant wird daraufhin geschlossen.



Eine ausführliche Beschreibung des Cache-Löschvorgangs finden Sie im FANUC-Handbuch „Tablet UI Operator’s Manual“ (Dokument-Nr. B-84274EN/05), Abschnitt 2.7 „When Tablet TP connect to another controller“.

Öffnen Sie die Konfigurationsseite, indem Sie im Menü den Reiter „Plugins“ auswählen und auf den Eintrag „GRIPLINK“ tippen.

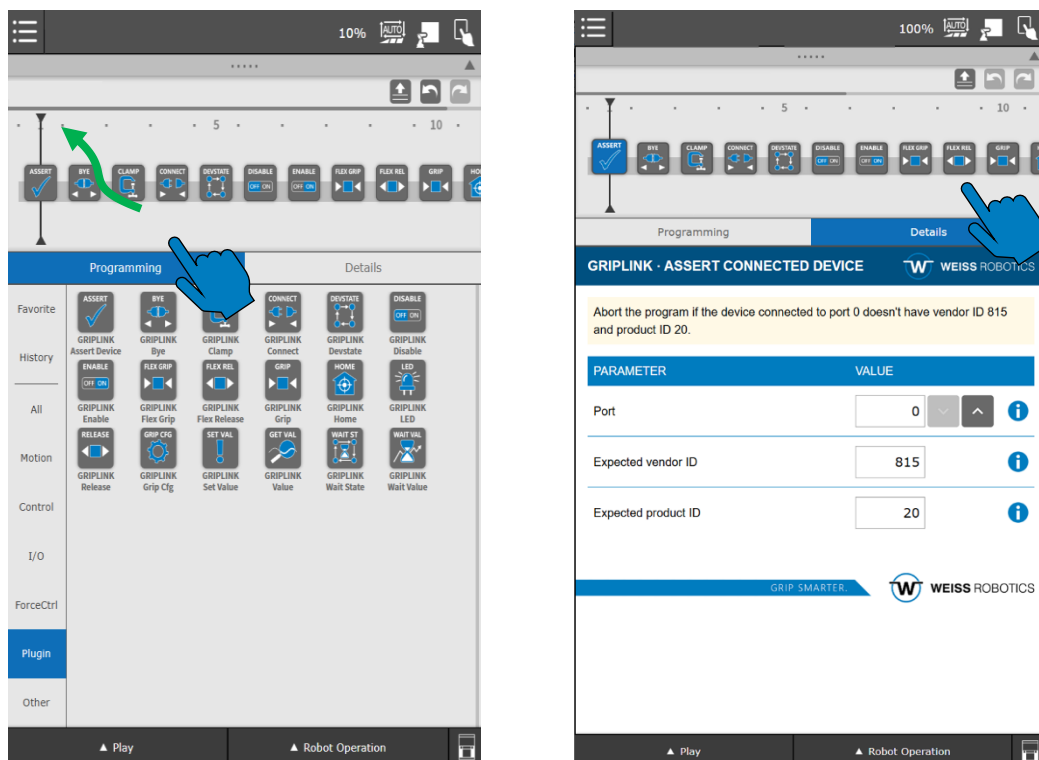


Nutzen Sie das Dropdown-Feld „COM Tag“, um den zuvor parametrierten COM-Tag einzustellen. Sollte ein COM-Tag ausgewählt werden, der fehlerhaft konfiguriert ist, wird eine Fehlermeldung angezeigt.

6.4 Anwendungen mit Instruktionen erstellen

Instruktionen sind vordefinierte Funktionsbausteine, die einzelne GRIPLINK-Befehle in grafischer Form kapseln. Sie ermöglichen die Konfiguration und Ausführung von Gerätefunktionen ohne direkte textuelle Programmierung. Jede Instruktion entspricht dabei einem Befehl des Core-Plugins und stellt dessen Parameter in einer strukturierten Benutzeroberfläche zur Verfügung. Dadurch können auch komplexe Abläufe schnell und fehlerarm erstellt werden.

Die Instruktionen des GRIPLINK-Plugins finden Sie unter der Rubrik „I/O“ im Programm-Editor. Sie können einfach per Drag & Drop in das Programm gezogen werden.



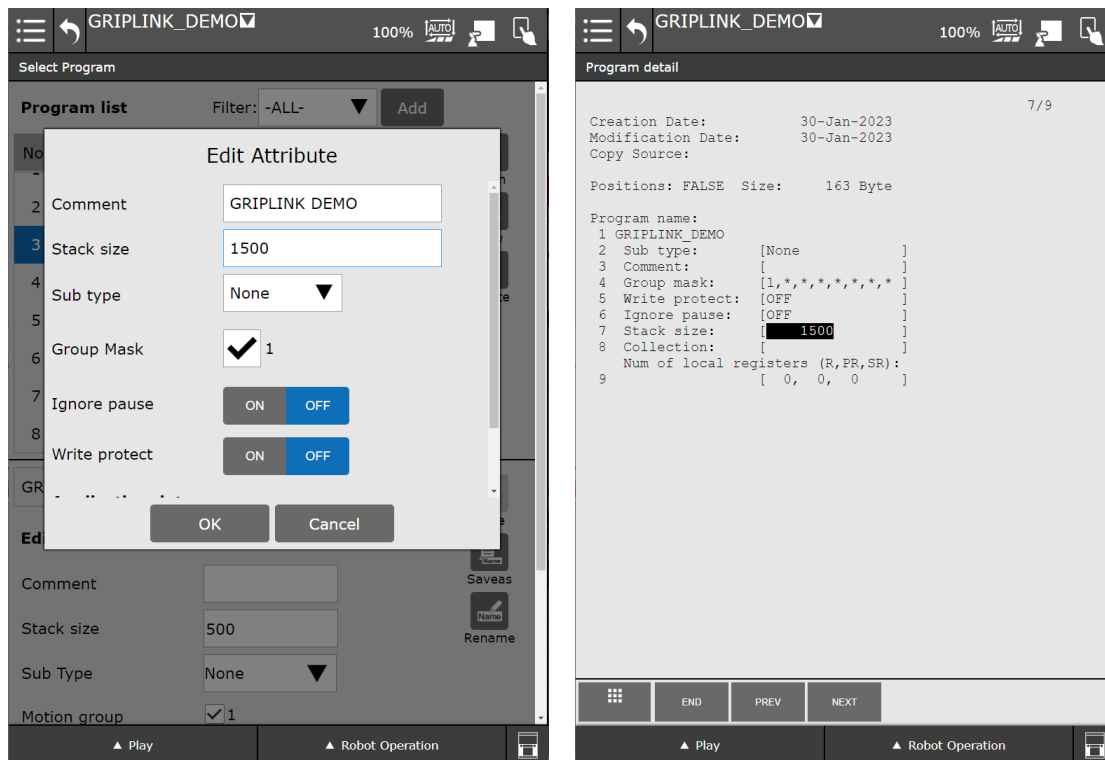
Wenn Sie eine Instruktion auswählen, gelangen Sie über den Reiter „Details“ auf eine Detailansicht des Befehls, wo Sie je nach Befehl spezifische Einstellungen vornehmen können. In der Detailansicht einer Instruktion werden alle verfügbaren Parameter übersichtlich dargestellt. Die Parametrierung erfolgt über Eingabefelder, Auswahloptionen sowie Registerzuweisungen.

Über die Schaltfläche „i“ können kontextsensitive Beschreibungen zu den einzelnen Parametern angezeigt werden, die zusätzliche Informationen zur Verwendung und Bedeutung liefern.

6.5 Stack-Größe

Da GRIPLINK-Instruktionen intern zusätzliche Funktionsaufrufe und Datenverarbeitung durchführen, sollte für entsprechende Roboterprogramme eine ausreichend große Stack-Größe konfiguriert werden. Insbesondere bei umfangreichen Programmen oder der Kombination mehrerer Instruktionen kann eine zu geringe Stack-Größe zu Laufzeitfehlern führen.

Tritt während der Programmausführung der Fehler „Stack Overflow“ auf, muss die Stack-Größe des Programms in den Programmeigenschaften entsprechend erhöht werden.



6.6 Empfohlene Programmstruktur

Programme, die GRIPLINK-Instruktionen verwenden, sollten grundsätzlich folgende Struktur aufweisen:

1. Verbindung herstellen mit der Instruktion GRIPLINK Connect
2. Geräte prüfen oder initialisieren (z. B. Assert Device, Enable, Home)
3. Gerätefunktionen ausführen (z. B. Grip, Release, Value, Wait Value)
4. Optional Verbindung trennen mit GRIPLINK Bye, wenn die physikalische Verbindung unterbrochen werden muss

In den meisten Anwendungen bleibt die Verbindung während der gesamten Programmlaufzeit bestehen.

6.7 Übersicht der verfügbaren Instruktionen

Die Instruktionen stellen eine grafische Konfigurations- und Ausführungsoberfläche für die Befehle des Core-Plugins zur Verfügung. Sie kapseln die zugrunde liegenden Funktionsaufrufe und ermöglichen es, deren Parameter strukturiert und übersichtlich über das Tablet Teach Pendant zu konfigurieren.

Jede Instruktion entspricht dabei einem spezifischen Befehl des Core-Plugins und bildet dessen Funktionalität vollständig ab. Anstelle der manuellen Eingabe von Funktionsparametern werden diese über

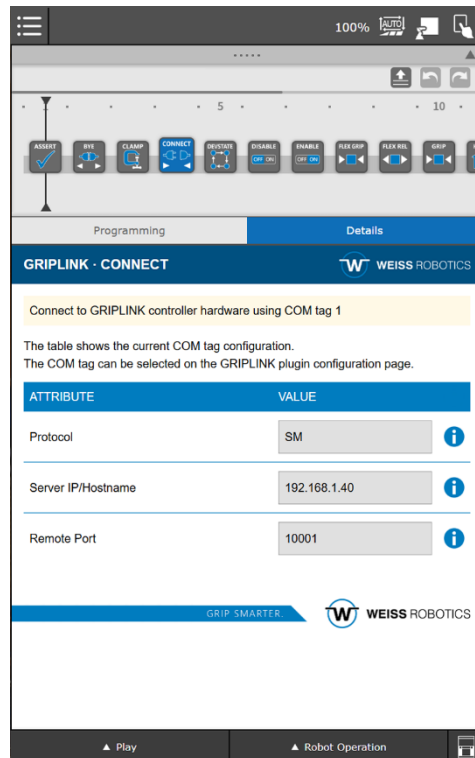
Eingabefelder, Auswahloptionen und vordefinierte Parameterstrukturen bereitgestellt. Dadurch wird die Erstellung von Roboterprogrammen vereinfacht und die Fehleranfälligkeit reduziert.

Intern werden die in der Instruktion gesetzten Parameter an das Interfaceprogramm übergeben und dort als entsprechender Core-Befehl ausgeführt. Somit bleibt die zugrunde liegende Kommunikationslogik unverändert, während dem Anwender eine komfortable und intuitive Bedienoberfläche zur Verfügung gestellt wird. Es gelten folgende allgemeine Hinweise:

- Port ist bei allen portbezogenen Befehlen ein ganzzahliger Portindex, dessen Wertebereich vom angeschlossenen GRIPLINK-Controller abhängen.
- Registerindizes werden im UI als R[n] angezeigt und liegen durchgängig im Bereich R[1] bis R[200].
- Mehrere Befehle verwenden ein Rückgaberegister für den Gerätezustand. Wenn in einer Instruktion kein gültiger Registerindex hinterlegt ist, wird der auf der Plugin-Konfigurationsseite definierte Standardwert verwendet.
- Parameter mit Umschaltung const/reg können entweder als Konstante im Befehl hinterlegt oder zur Laufzeit aus einem Numeric Register gelesen werden.
- Das interne Format solcher Parameter ist entweder ein numerischer Wert, zum Beispiel 12.5, oder ein Registerbezug im Format R[n], zum Beispiel R[42].
- Rückgabewerte vom Typ Gerätezustand sind Statuscodes des angeschlossenen Geräts, siehe Anhang A.

6.7.1 Verbindung aufbauen – CONNECT

Stellt die Verbindung zwischen Robotersteuerung und GRIPLINK-Controller anhand der globalen Plugin-Konfiguration her.



Parameter

Keine editierbaren Instruktionsparameter. CONNECT verwendet implizit den auf der GRIPLINK-Konfigurationsseite ausgewählten COM-Tag. Bei ungültiger Kommunikationskonfiguration wird ein Fehler angezeigt.

Beschreibung

Keine.

Syntax im Skriptmodus

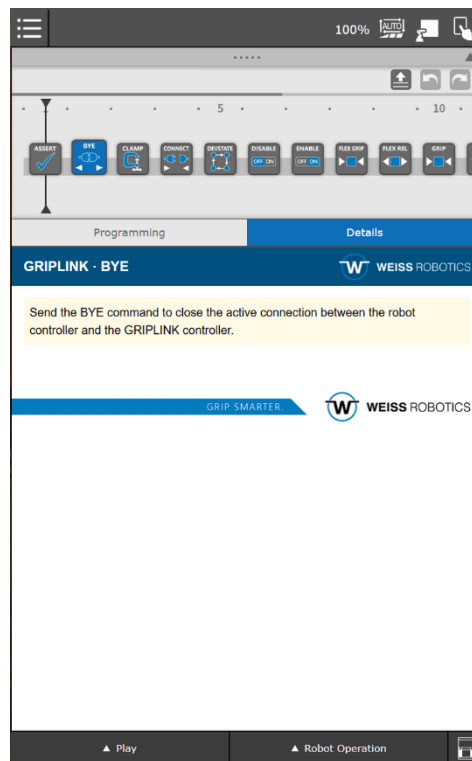
```
CALL IPL_WR_GRIPLINK_CONNECT
```

6.7.2 Verbindung trennen – BYE

Sendet den Befehl BYE, um die aktive Verbindung zwischen FANUC-Steuerung und GRIPLINK-Controller sauber zu schließen.



Nutzen Sie diese Instruktion innerhalb eines Roboterprogramms nur, wenn die physikalische Datenverbindung zum GRIPLINK-Controller unterbrochen werden soll, beispielsweise bei einem Werkzeugwechsel.



Parameter

keine

Rückgabewerte

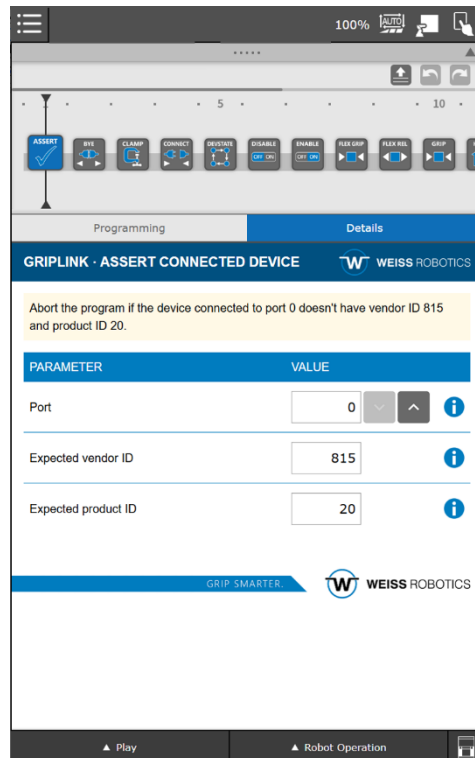
keine

Syntax im Skriptmodus

```
CALL IPL_WR_GRIPLINK_BYE
```

6.7.3 Angeschlossene Geräte prüfen – ASSERTDEV

Prüft, ob am angegebenen Port das erwartete Gerät angeschlossen ist. Das TP-Programm wird abgebrochen, wenn Vendor-ID oder Product-ID nicht passen.



Parameter

Port	Port-Index (0 bis 31)
Expected vendor ID	Erwartete Vendor-ID aus der IODD-Datei.
Expected product ID	Erwartete Product-ID aus der IODD-Datei.

Rückgabewerte

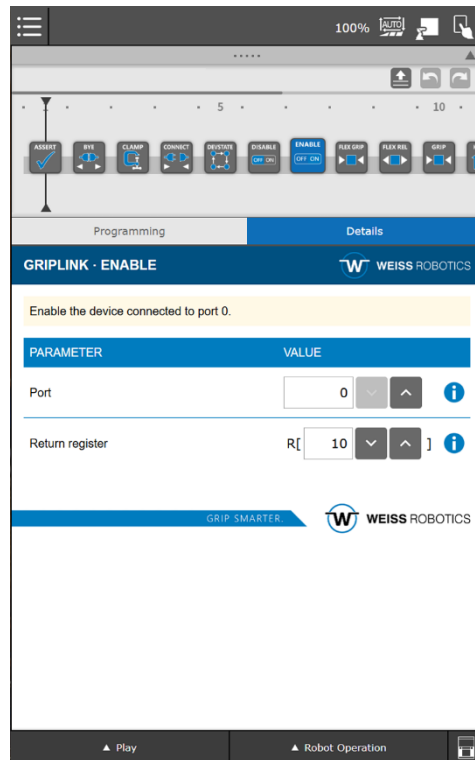
Bei Abweichung erfolgt ein Programmabbruch statt einer normalen Rückgabe.

Syntax im Skriptmodus

```
CALL IPL_WR_GRIPLINK_ASSERTDEV(<Port>,<VID>,<PID>)
```

6.7.4 Gerät aktivieren – ENABLE

Aktiviert das am Port angeschlossene Gerät.



Parameter

Port	Port-Index (0 bis 31)
Return register	Zielregister für den resultierenden Gerätezustand

Rückgabewerte

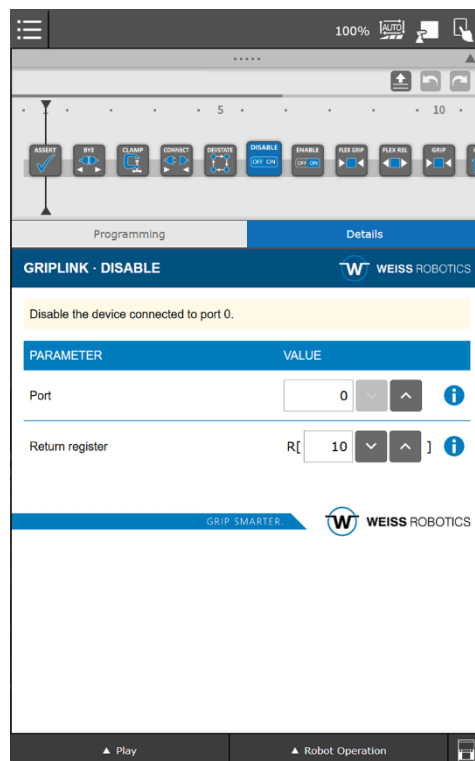
Schreibt den Gerätezustand nach R[n].

Syntax im Skriptmodus

```
CALL IPL_WR_GRIPLINK_ENABLE(<Port>,<Register Index>)
```

6.7.5 Gerät deaktivieren – DISABLE

Deaktiviert das am Port angeschlossene Gerät.



Parameter

Port	Port-Index (0 bis 31)
Return register	Zielregister für den resultierenden Gerätezustand

Rückgabewerte

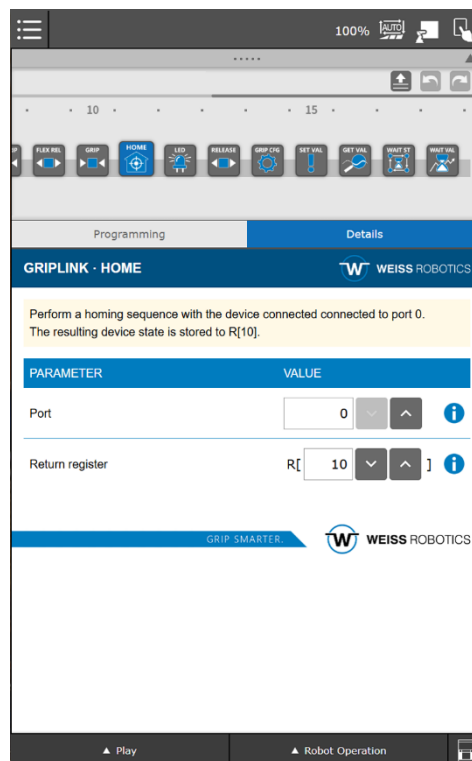
Schreibt den Gerätezustand nach R[n].

Syntax im Skriptmodus

```
CALL IPL_WR_GRIPLINK_DISABLE(<Port>,<Register Index>)
```

6.7.6 Referenzieren – HOME

Führt eine Homing-Sequenz des Geräts aus.



Parameter

Port	Port-Index (0 bis 31)
Return register	Zielregister für den Gerätezustand nach dem Homing.

Rückgabewerte

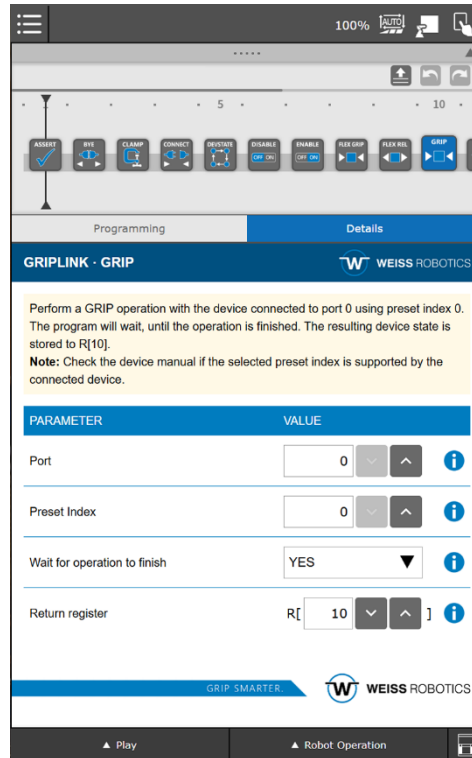
Schreibt den resultierenden Gerätezustand nach R[n].

Syntax im Skriptmodus

```
CALL IPL_WR_GRIPLINK_HOME(<Port>,<Register Index>)
```

6.7.7 Werkstück über Preset greifen – GRIP

Führt eine GRIP-Operation mit einem vorgegebenen Preset aus. Optional kann blockierend auf das Ende der Operation gewartet werden.



Parameter

Port:	Port-Index (0 bis 31)
Preset Index:	Ganzzahl 0 bis 9999. Preset der Greifkonfiguration.
Wait for operation...:	Auswahl NO oder YES. NO: Startet die Operation und kehrt sofort zurück. YES: Wartet bis zum Ende der Operation.
Return register:	Nur relevant für den blockierenden Modus; Zielregister für den resultierenden Gerätezustand.

Rückgabewerte

Bei Wait = YES wird der resultierende Gerätezustand nach R[n] geschrieben.

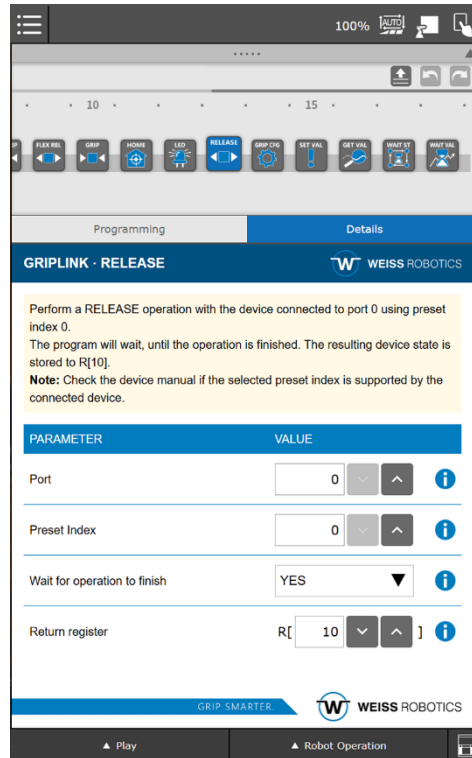
Bei Wait = NO gibt es keine synchron ausgewertete Statusrückgabe durch die Instruktion selbst.

Syntax im Skriptmodus

```
CALL IPL_WR_GRIPLINK_GRIP(<Port>,<Preset Index>,  
    <do block: 0=no,1=yes>,<Register Index>)
```

6.7.8 Werkstück über Preset freigeben – RELEASE

Führt eine RELEASE-Operation mit einem vorgegebenen Preset aus. Optional kann blockierend auf das Ende der Operation gewartet werden.



Parameter

Port:	Port-Index (0 bis 31)
Preset Index:	Ganzzahl 0 bis 9999. Preset der Freigabekonfiguration.
Wait for operation...:	Legt fest, ob auf die Beendigung der Operation gewartet werden soll. NO: Startet die Operation und kehrt sofort zurück. YES: Wartet bis zum Ende der Operation.
Return register:	Nur relevant für den blockierenden Modus; Zielregister für den resultierenden Gerätezustand.

Rückgabewerte

Bei Wait = YES wird der resultierende Gerätezustand nach R[n] geschrieben.

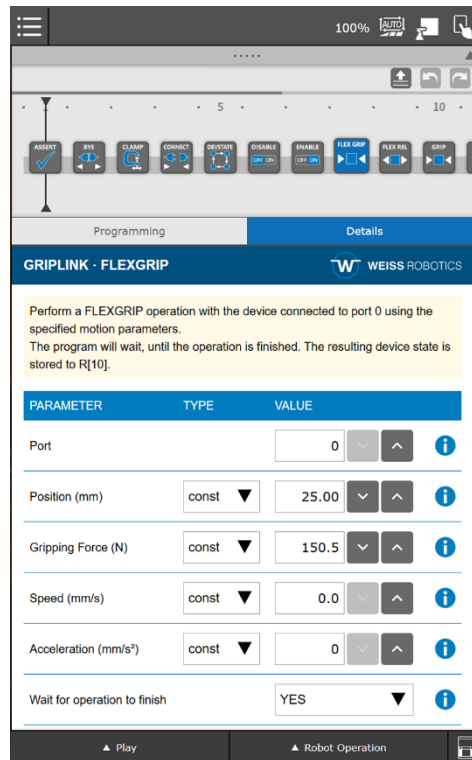
Bei Wait = NO gibt es keine synchron ausgewertete Statusrückgabe durch die Instruktion selbst.

Syntax im Skriptmodus

```
CALL IPL_WR_GRIPLINK_RELEASE(<Port>,<Preset Index>,  
    <do block: 0=no,1=yes>,<Register Index>)
```

6.7.9 Werkstück über Parameter greifen – FLEXGRIP

Startet eine Greifbewegung mit frei parametrierbaren Bewegungswerten. Die Operation kann nicht blockierend oder blockierend ausgeführt werden.



Parameter

Port:	Port-Index (0 bis 31)
Position (mm):	Positionswert des No-Part-Limits in mm, bei dem der Greifer während des Greifens in den Zustand NO PART wechselt. Kann entweder als konstanter Wert (Auswahl „const“) oder zur Laufzeit über ein Register gesetzt werden (Auswahl „reg“).
Gripping Force (N):	Greifkraft in Newton. Kann entweder als konstanter Wert (Auswahl „const“) oder zur Laufzeit über ein Register gesetzt werden (Auswahl „reg“).
Speed (mm/s):	Fingergeschwindigkeit in mm/s. Kann entweder als konstanter Wert (Auswahl „const“) oder zur Laufzeit über ein Register gesetzt werden (Auswahl „reg“).
Acceleration (mm/s²):	Speed = 0 verwendet die Standardvorgabe des Geräts. Fingerbeschleunigung beim Greifen in mm/s². Kann entweder als konstanter Wert (Auswahl „const“) oder zur Laufzeit über ein Register gesetzt werden (Auswahl „reg“).
Wait for operation...:	Acceleration = 0 verwendet die Standardvorgabe des Geräts. Legt fest, ob auf die Beendigung der Operation gewartet werden soll. NO: Startet die Operation und kehrt sofort zurück.

Return register: YES: Wartet bis zum Ende der Operation.
Nur relevant für den blockierenden Modus: Legt das Rückgaberegister für den Gerätezustand fest.

Rückgabewerte

Bei Wait = YES wird der resultierende Gerätezustand nach R[n] geschrieben.

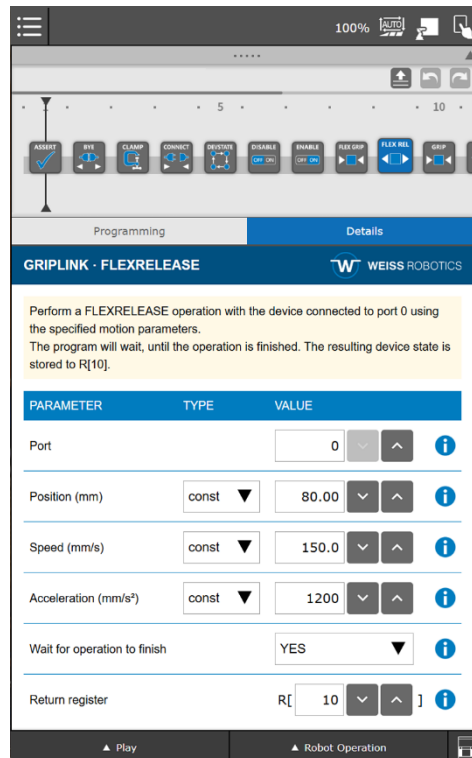
Bei Wait = NO startet die Bewegung und die Instruktion kehrt sofort zurück.

Syntax im Skriptmodus

```
CALL IPL_WR_GRIPLINK_FLEXGRIP(<Port>,<Position>,<Gripping Force>,  
    <Speed>,<Acceleration>,<do block: 0=no,1=yes>,<Register Index>)
```

6.7.10 Werkstück über Parameter freigeben – FLEXRELEASE

Gibt ein zuvor gegriffenes Werkstück mit frei parametrierbaren Bewegungswerten frei. Die Operation kann auch zum Vorpositionieren der Greiffinger genutzt werden. Sie kann sowohl nicht blockierend als auch blockierend ausgeführt werden.



Parameter

Port:	Port-Index (0 bis 31)
Position (mm):	Freigabe-Position (Release-Limit) in mm. Erreichen die Finger die Position, wechselt der Greifer in den Zustand RELEASED. Kann entweder als konstanter Wert (Auswahl „const“) oder zur Laufzeit über ein Register gesetzt werden (Auswahl „reg“).
Speed (mm/s):	Fingergeschwindigkeit in mm/s. Kann entweder als konstanter Wert (Auswahl „const“) oder zur Laufzeit über ein Register gesetzt werden (Auswahl „reg“). Speed = 0 verwendet die Standardvorgabe des Geräts.
Acceleration (mm/s²):	Fingerbeschleunigung beim Freigeben in mm/s². Kann entweder als konstanter Wert (Auswahl „const“) oder zur Laufzeit über ein Register gesetzt werden (Auswahl „reg“). Acceleration = 0 verwendet die Standardvorgabe des Geräts.
Wait for operation...:	Legt fest, ob auf die Beendigung der Operation gewartet werden soll. NO: Startet die Operation und kehrt sofort zurück. YES: Wartet bis zum Ende der Operation.

Return register: Nur relevant für den blockierenden Modus: Legt das Rückgaberegister für den Gerätezustand fest.

Rückgabewerte

Bei Wait = YES wird der resultierende Gerätezustand nach R[n] geschrieben.

Bei Wait = NO startet die Bewegung und die Instruktion kehrt sofort zurück.

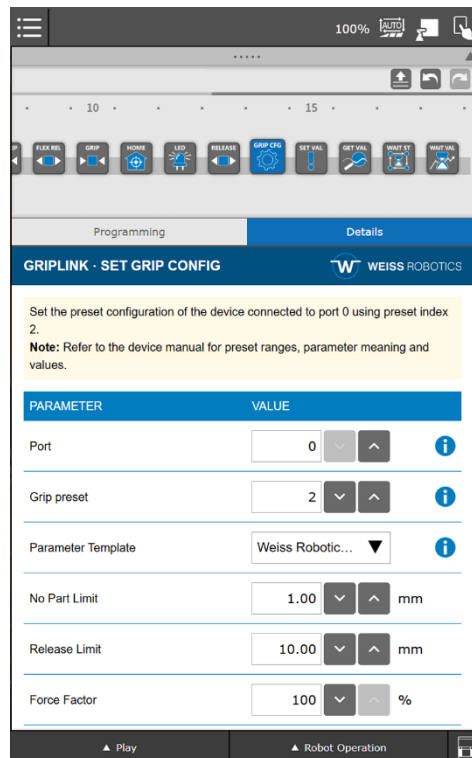
Syntax im Skriptmodus

```
CALL IPL_WR_GRIPLINK_FLEXRELEASE(<Port>,<Position>,<Speed>,  
                                  <Acceleration>,<do block: 0=no,1=yes>,<Register Index>)
```

6.7.11 Parametrierung eines Griff-Presets – SGRIPCFG

In der Detailseite der Instruktion „SGRIPCFG“ können Port des gewünschten Greifers und der Index des Presets ausgewählt werden.

Mit dem Parameter Template kann dargestellt werden, welche Bedeutung die einzelnen Parameter haben. Für bestimmte Geräte werden nicht-relevante Parameter ausgeblendet. Abhängig vom Template werden auch zulässige Wertebereiche geprüft.



Parameter

Port:	Port-Index (0 bis 31)
Grip preset:	Preset-Index, der überschrieben werden soll.
Parameter Template:	Auswahl des Interpretationsschemas

Rückgabewerte

Keine

Syntax im Skriptmodus

```
CALL IPL_WR_GRIPLINK_SGRIPCFG(<Port>,<Preset Index>,<Template>,  
<Param 1>,<Param 2>,<Param 3>,<Param 4>,<Param 5>,<Param 6>,<Param 7>,<Param 8>)
```



Der "Template"-Parameter ist für die Dekodierung der Parameter in der grafischen Ansicht notwendig. Damit Sie zwischen textueller und grafischer Darstellung wechseln können, muss er einen der folgenden Werte besitzen:
0: zeige generisches UI-Template

- 1: zeige Weiss Robotics UI-Template (Standard für alle Weiss Robotics IO-Link Greifer)
- 2: zeige Weiss Robotics UI-Template (Standard für Weiss Robotics WPG Greifer)
- 3: zeige generisches Vakuum-Sauger UI-Template

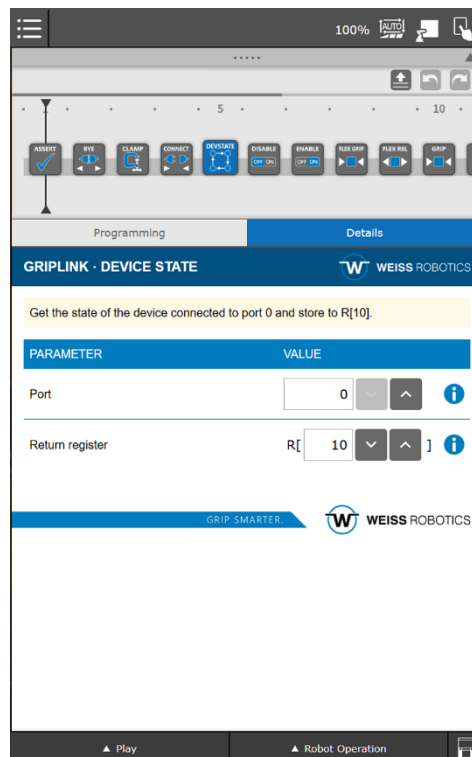


Beachten Sie, dass die Anzahl der konfigurierbaren Griffe vom Gerätetyp abhängt.

6.7.12 Auslesen des Gerätezustands – DEVSTATE

Um den Gerätezustand eines angeschlossenen Geräts auszulesen, kann die Instruktion „DEVSTATE“ verwendet werden. Diese fragt den aktuellen Zustand des Geräts am ausgewählten Port ab und speichert ihn im angegebenen Register.

Jedes Gerät besitzt stets einen eindeutigen Gerätezustand, der sich durch Aktionen wie Greifen, Freigeben, Aktivieren oder Deaktivieren ändern kann. Mit diesem Befehl kann im Roboterprogramm direkt auf Zustandsänderungen eines Geräts reagiert werden. So kann beispielsweise erkannt werden, ob ein Greifer erfolgreich ein Bauteil gegriffen hat oder ob kein Bauteil detektiert wurde. Eine vollständige Liste der möglichen Gerätezustände befindet sich in Anhang A.



Parameter

Port:	Port-Index (0 bis 31)
Return register:	Legt das Rückgaberegister für den Gerätezustand fest.

Rückgabewerte

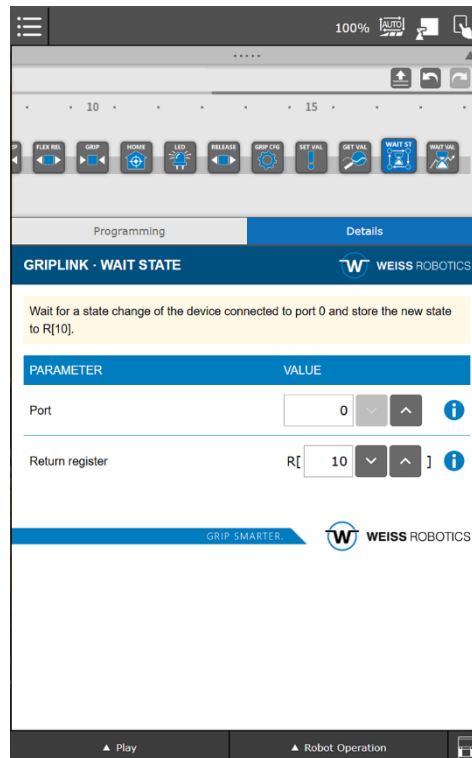
Schreibt den aktuell gelesenen Gerätezustand nach R[n].

Syntax im Skriptmodus

```
CALL IPL_WR_GRIPLINK_DEVSTATE(<Port>,<Register Index>)
```

6.7.13 Warten auf eine Änderung des Gerätezustands – WAITSTATE

Wartet blockierend auf eine Zustandsänderung des Geräts am angegebenen Port. Dies kann beispielsweise verwendet werden, um zu prüfen, ob eine asynchron ausgeführte Freigabebewegung (RELEASE mit Option „Wait for operation to finish“=NO) abgeschlossen ist.



Parameter

Port:	Port-Index (0 bis 31)
Return register:	In dieses Register wird der neue Gerätezustand geschrieben, sobald sich der Zustand ändert.

Rückgabewerte

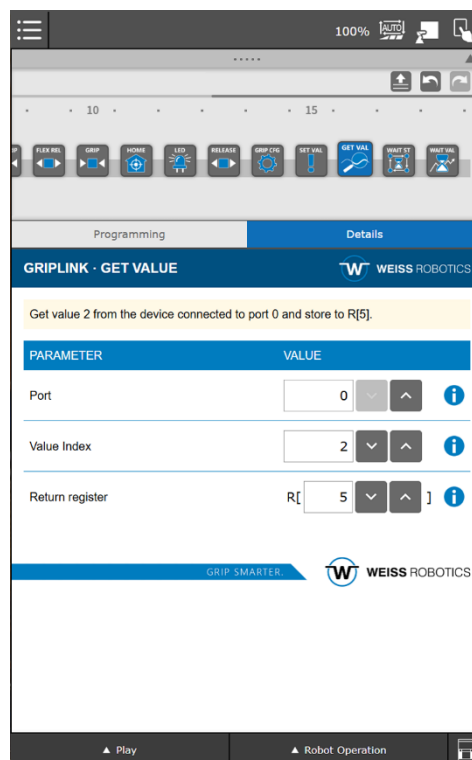
Schreibt den neuen Gerätezustand nach R[n].

6.7.14 Gerätewert abfragen – VALUE

Um einen Messwert eines angeschlossenen Geräts im Roboterprogramm zu verwenden, kann die Instruktion „VALUE“ genutzt werden. Mit dieser Instruktion lassen sich Mess- oder Prozesswerte eines Geräts auslesen und in einem Register speichern und im weiteren Programmverlauf nutzen.



Der verwendete Index hat eine gerätespezifische Bedeutung. Welche Werte über die einzelnen Indizes verfügbar sind, ist der Dokumentation des jeweiligen Geräts zu entnehmen.



Parameter

Port:	Port-Index (0 bis 31)
Value Index:	Index des abzufragenden Wertes. Value Index = 0 liefert immer den Primärwert des Geräts. Weitere Indizes sind geräteabhängig.
Return register:	Zielregister für den gelesenen Wert.

Rückgabewerte

Schreibt den gelesenen Wert nach R[n].

Syntax im Skriptmodus

```
CALL IPL_WR_GRIPLINK_VALUE(<Port>,<Value Index>,<Register Index>)
```

6.7.15 Gerätewert setzen – SETVALUE

Mit der Instruktion „SETVALUE“ kann ein Wert an ein angeschlossenes Gerät übertragen werden. Der zu schreibende Wert kann entweder direkt als konstante Fließkommazahl eingegeben oder indirekt über den Inhalt eines Registers definiert werden.

Der verwendete Index bestimmt, welcher Gerätewert geschrieben wird. Die Bedeutung der einzelnen Indizes ist gerätespezifisch und muss der Dokumentation des jeweiligen Geräts entnommen werden.



Parameter

Port:	Port-Index (0 bis 31)
Value Index:	Geräteabhängiger Werteindex.
Value:	Zu setzender Wert. Kann entweder als konstanter Wert (Auswahl „const“) oder zur Laufzeit über ein Register gesetzt werden (Auswahl „reg“).

Rückgabewerte

Keine.

Syntax im Skriptmodus

```
CALL IPL_WR_GRIPLINK_SETVALUE(<Port>,<Value Index>,<Value>)
```

6.7.16 Warten auf einen Gerätewert – WAITVALUE

Mit der Instruktion „WAITVALUE“ kann das Roboterprogramm warten, bis ein bestimmter Mess- oder Prozesswert eines angeschlossenen Geräts erreicht wurde.

Die Instruktion überwacht kontinuierlich einen ausgewählten Gerätewert und setzt die Programmausführung erst fort, wenn der gemessene Wert innerhalb eines definierten Toleranzfensters um den Zielwert liegt. Dadurch kann das Roboterprogramm zuverlässig auf Zustände reagieren, die durch externe Geräte oder Sensoren gemeldet werden.

Typische Anwendungsfälle sind:

- Warten, bis ein Sensor einen bestimmten Messwert erreicht
- Prüfen, ob ein Prozesswert innerhalb eines zulässigen Bereichs liegt
- Synchronisieren des Roboterprogramms mit einem externen Gerät

Wird der gewünschte Wert innerhalb der angegebenen Timeout-Zeit nicht erreicht, beendet die Instruktion die Ausführung mit einem Timeout-Fehler, wodurch das Roboterprogramm abgebrochen wird.

Parameter

Port:	Port-Index (0 bis 31)
Value Index:	Index des zu überwachenden Werts (zulässiger Wertebereich ist geräteabhängig).
Threshold Value:	Zielwert in der Einheit des Messwerts
Window Size:	Breite des Toleranzfensters symmetrisch zum Zielwert
Timeout (ms):	Wert für das Timeout in Millisekunden

Rückgabewerte

Keine. Bei Zeitüberschreitung wird ein Timeout-Fehler ausgelöst und das TP-Programm abgebrochen.

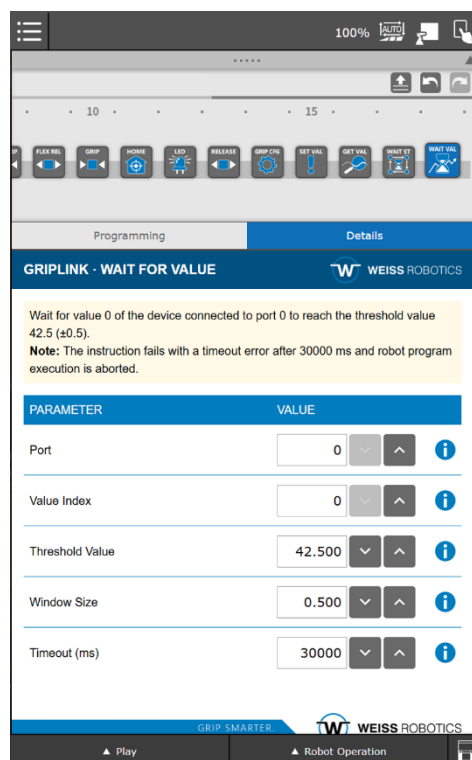
6.7.17 Greifkrafterhaltung steuern – CLAMP

Mit dem CLAMP Befehl können Greifmodule mit integrierter Greifkraftsicherung im Zustand HOLDING den Antrieb stromlos schalten, um den Strombedarf des Greifmoduls zu reduzieren. Das Werkstück wird weiterhin sicher gehalten.

Wird CLAMP aktiviert, schaltet sich der Antrieb im Zustand HOLDING ab. Wird CLAMP deaktiviert, bleibt im Zustand HOLDING die Greifkraftregelung aktiv und die Steuerung des Greifmoduls regelt die Greifkraft permanent nach.



Der CLAMP Befehl ist nicht bei allen Greifmodulen verfügbar.



Parameter

Port:	Port-Index (0 bis 31)
State:	Steuert die mechanische Greifkrafterhaltung ON: Greifkrafterhaltung wird im Zustand HOLDING aktiviert und der Motor abgeschaltet. OFF: Der Motor bleibt im Zustand HOLDING geregelt.

Rückgabewerte

Keine.

Syntax im Skriptmodus

```
CALL IPL_WR_GRIPLINK_CLAMP (<Port>, <State>)
```

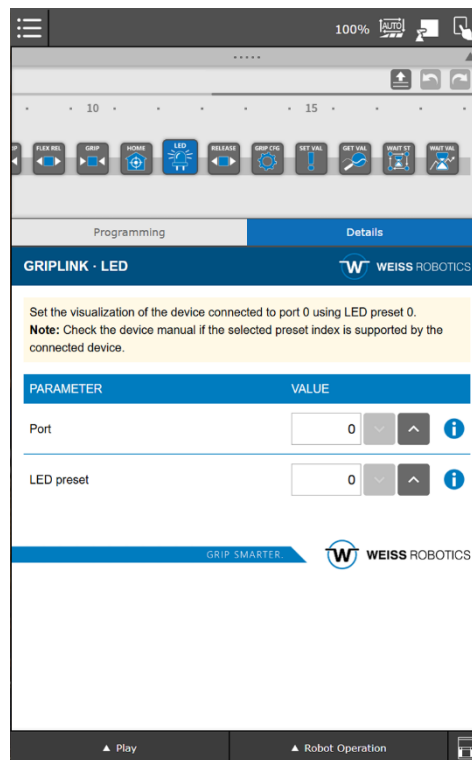
6.7.18 Ansteuerung der Status-LED – LED

Mit der Instruktion „LED“ kann die LED-Anzeige eines angeschlossenen Geräts gesteuert werden. Über diese Funktion lassen sich vordefinierte LED-Presets aktivieren, um beispielsweise den aktuellen Betriebszustand eines Geräts visuell darzustellen.

Die Instruktion sendet das ausgewählte LED-Preset an das Gerät, das am angegebenen Port angeschlossen ist. Das Gerät stellt daraufhin seine LED-Anzeige entsprechend der gewählten Voreinstellung dar.



Die Verfügbarkeit dieser Funktion hängt vom jeweils angeschlossenen Gerät ab. Nicht alle Geräte unterstützen eine externe Steuerung der LED-Anzeige. Informationen darüber, ob und in welchem Umfang die LED-Anzeige gesteuert werden kann, sind der Betriebsanleitung des jeweiligen Geräts zu entnehmen.



Parameter

Port:	Port-Index (0 bis 31)
LED preset:	Index zur Auswahl des LED-Presets. Die Bedeutung des Presets ist geräteabhängig.

Rückgabewerte

Keine.

Syntax im Skriptmodus

```
CALL IPL_WR_GRIPLINK_LED(<Port>,<Preset Index>)
```

7 Fehlersuche

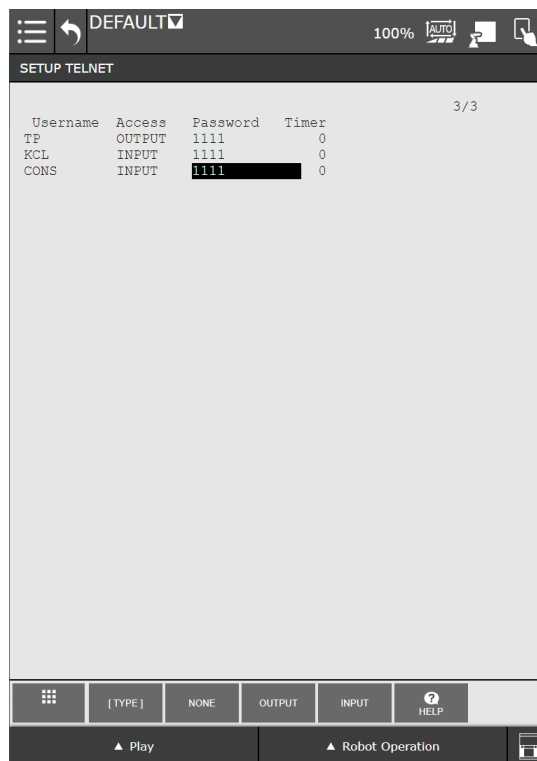
Das GRIPLINK-Plugin gibt im Betrieb Fehlermeldungen aus. Im Folgenden werden wichtige Meldungen und Lösungswege erläutert.

7.1 TELNET-Ausgabe

Um Fehler im laufenden Betrieb zu analysieren, können Sie die TELNET-Ausgabe aktivieren. Navigieren Sie über SETUP – Host Comm – 2 TELNET in das entsprechende Setup-Menü. Klicken Sie in das Feld „Password“ in der Zeile „CONS“ und geben Sie ein Passwort ein (z.B. 1111).



Hinweis für RoboGuide: Bei Steuerungen der Version R-30iB muss die Eingabe des Passworts über das Tastenfeld des emulierten alten Teach Pendants erfolgen. Nutzen Sie den Ziffernblock und die Tasten „ENTER“, um das Passwort einzugeben und zu bestätigen.



Nun können Sie sich mit einem Terminal-Programm (z.B. PuTTY) über TELNET mit der Robotersteuerung verbinden. Dies funktioniert auch mit RoboGuide.

7.2 Fehlermeldungen

7.2.1 Installationsfehlermeldung „MEMO-094 Parameters are different“

```
2 MEMO-094 Parameters are different
3 R E S E T
4 SYST-045 TP enabled in AUTO mode
5 R E S E T
```

Mögliche Ursache

Aktualisierung von älterer auf neueste Plugin-Version

Behebung

- Löschen Sie sämtliche Programm- und Variablendateien mit den Präfixen „GL_“ und „IPL_WR_GL_“ sowie die Dateien mit dem Namen „GRIPLINK“ von der Robotersteuerung (siehe Abbildungen unten)
- Wiederholen Sie die Installation des Plugins

```
MD:\*. * 128/438
127 GIGELOG DG
128 GL_CMD PC
129 GL_COMM PC
130 GL_COMMON PC
131 GL_DEBUG PC
132 GRIPLINK PC
133 GRIPLINK VA
134 GRIPLINK VR
```

```
MD:\*. * 157/438
156 IPL_FANUC_SYS_ERPOST VR
157 IPL_WR_GRIPLINK_ASSERTDEV LS
158 IPL_WR_GRIPLINK_ASSERTDEV TP
159 IPL_WR_GRIPLINK_BYE LS
160 IPL_WR_GRIPLINK_BYE TP
161 IPL_WR_GRIPLINK_CFG PC
162 IPL_WR_GRIPLINK_CFG VA
163 IPL_WR_GRIPLINK_CFG VR
164 IPL_WR_GRIPLINK_CLAMP LS
165 IPL_WR_GRIPLINK_CLAMP TP
166 IPL_WR_GRIPLINK_CONNECT LS
167 IPL_WR_GRIPLINK_CONNECT TP
168 IPL_WR_GRIPLINK_CRXINIT LS
169 IPL_WR_GRIPLINK_CRXINIT TP
170 IPL_WR_GRIPLINK_DEVSTATE LS
171 IPL_WR_GRIPLINK_DEVSTATE TP
172 IPL_WR_GRIPLINK_DISABLE LS
173 IPL_WR_GRIPLINK_DISABLE TP
```

Abbildung zeigt nicht alle Programme!

7.2.2 Systemfehlermeldung „EtherNet Adapter Error (1)“

Mögliche Ursache

Standardmäßig ist ein Ethernet-Adapter eingerichtet, der deaktiviert werden muss.

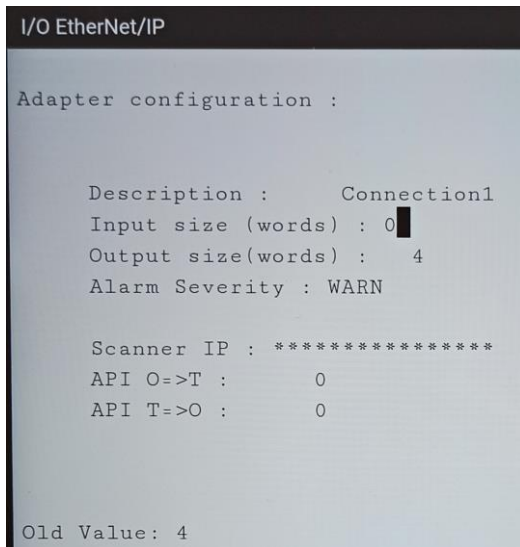
Behebung

- Navigieren Sie in das Menü „I/O“ → „EtherNet/IP“
- Setzen Sie in der ersten Zeile (Connection 1) das Feld ENABLED von „True“ auf „False“
- Wechseln Sie über den Softkey „Config“ in die Konfigurationsansicht

- Setzen Sie die Datengröße für „Input size“ und „Output size“ auf 0
Ggf. müssen Sie das Legacy Pendant über die Schaltfläche unten rechts aktivieren und die Zahl über dessen Numpad eingeben
- Starten Sie die Robotersteuerung neu

EtherNet/IP List(Rack 89)

Description	TYP	Enable	Status	Slot
Connection1	ADP	TRUE	ONLINE	1



7.2.3 Meldung „Could send command [command]: not connected“

Der Befehl [command] konnte nicht gesendet werden.

Mögliche Ursache

Ein GRIPLINK-Befehl wurde ausgeführt, ohne dass zuvor GRIPLINK Connect aufgerufen wurde oder der Programmzeiger steht nicht auf der ersten Instruktion.

Behebung

- Programmknoten GRIPLINK Connect einbauen (siehe Abschnitte 5.1 und 6.7.1)
- Programmzeiger auf die erste Instruktion verschieben

7.2.4 Meldung “GRIPLINK: Cmd [command] failed: E_NOT_OPEN”

Der Befehl [command] konnte nicht gesendet werden, da die Schnittstelle nicht geöffnet ist.

Mögliche Ursache

Ein GRIPLINK-Befehl wurde ausgeführt, ohne dass zuvor GRIPLINK Connect aufgerufen wurde.

Behebung

Programmknoten GRIPLINK Connect einbauen (siehe Abschnitte 5.1 und 6.7.1)

7.2.5 Meldung „GRIPLINK Version not supported“

Mögliche Ursache

Veraltete Firmware-Version auf dem GRIPLINK-Controller

Behebung

Aktualisieren Sie alle verwendeten GRIPLINK-Controller auf die in den Systemvoraussetzungen aufgeführte Firmware-Version (siehe Abschnitt 1.3).

7.2.6 Meldung „GL_COMM 190: undefined built-in“

Mögliche Ursache

Die KAREL-Option ist nicht aktiviert

Behebung

Prüfen Sie, ob die KAREL-Option aktiviert wurde, indem Sie im Setup „Host Comm“ zu den HTTP-Einstellungen navigieren. Hier muss in der linken Spalte der Zeile „KAREL:*“ ein „U“ eingetragen sein. Falls nicht, drücken Sie den Shortkey „UNLOCK“.

HTTP Setup		
PROTECTED RESOURCES		
Name	Pwrd	Resource
U	*****	iPendant
U	*****	KAREL:*
U	*****	KCL:*
U	*****	VISION SETUP
A	*****	*****
A	*****	*****
A	*****	*****
A	*****	*****

Die Installation der KAREL- oder der User Socket Messaging-Option wurde nicht korrekt verarbeitet. Machen Sie ein vollständiges Backup Ihres Roboters. Kontaktieren Sie Ihren FANUC-Support, um die Software des Roboters vollständig zurückzusetzen und installieren Sie die neueste Software in Verbindung mit den erforderlichen Software-Optionen für GRIPLINK und Ihre Applikation.



Setzen Sie sich unbedingt mit dem FANUC-Support in Verbindung, bevor Sie den Roboter zurücksetzen und Software neu installieren!

7.2.7 Meldung „GRIPLINK could not connect via SM slot Cx:“

Mögliche Ursache

Socket-Verbindung zwischen Robotersteuerung und GRIPLINK-Controller konnte nicht aufgebaut werden.

Behebung

- Prüfen Sie die Spannungsversorgung des GRIPLINK-Controllers
- Gehen Sie im Setup „Host Comm“ zu den Einstellungen des gewählten Client-Tags (Cx) (siehe Kapitel 4.5)
- Wählen Sie über den Button [ACTION] die Aktion „UNDEFINE“
- Wählen Sie über den Button [ACTION] die Aktion „DEFINE“
- Wählen Sie über den Button [ACTION] die Aktion „START“
- Starten Sie das aktuelle Programm neu
- Gegebenenfalls unterbrechen Sie die Spannungsversorgung des GRIPLINK, um diesen zurückzusetzen

7.2.8 Meldung „Listen failed:“

Mögliche Ursache

GRIPLINK-Controller nicht aktiv

Behebung

Prüfen Sie die Spannungsversorgung des GRIPLINK-Controllers.

7.2.9 Meldung „Device x not connected“

Mögliche Ursache

Es wurde ein GRIPLINK-Port ausgewählt, an dem aktuell kein Gerät angeschlossen ist oder es liegt ein Hardwaredefekt vor.

Behebung

Prüfen Sie die Ports und die angeschlossenen Geräte

7.2.10 Meldung „Stack Overflow“

Mögliche Ursache

Der Stack des aktuellen Programms ist nicht ausreichend groß gewählt.

Behebung

Erhöhen Sie die Stack-Größe über die Einstellungen des Programms.

7.2.11 Meldung „GRIPLINK: Failed to listen on socket“

Mögliche Ursache

Die Socket-Verbindung zwischen Roboter-Steuerung und GRIPLINK wurde unterbrochen.

Behebung

- Prüfen Sie die Netzwerkverbindung (Netzkabel) zwischen Roboter-Steuerung und GRIPLINK-Controller. Vergewissern Sie sich außerdem, dass sie den Anforderungen entspricht (Ist die Strombegrenzung am Netzteil korrekt eingestellt?)
- Starten Sie das gewünschte Roboter-Programm neu. Es muss zu Beginn einen „Connect“-Node ausführen!
- Die Meldung erscheint auch, wenn Sie die Verbindung bewusst unterbrechen wollen.

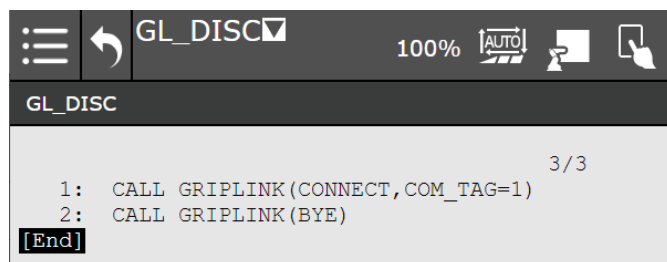
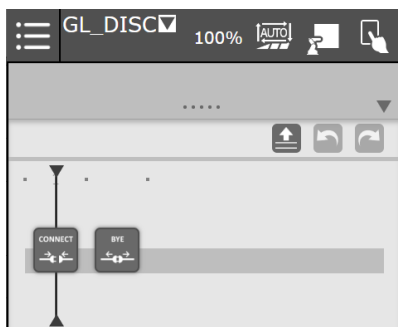
7.2.12 Verbindung zwischen Roboter und GRIPLINK-Controller nicht trennbar

Mögliche Ursache

Die Socket-Verbindung zwischen Roboter-Steuerung und GRIPLINK ist noch aktiviert, weil das Roboterprogramm in der Mitte gestoppt wurde.

Behebung

- Unterbrechen Sie kurzzeitig die Spannungsversorgung vom GRIPLINK-Controller
- Erstellen Sie ein separates Roboterprogramm und fügen Sie lediglich die CONNECT und die BYE Instruktionen ein. Wählen Sie das Programm an und führen Sie es aus. Dadurch wird eine Verbindung zum GRIPLINK-Controller auf- und wieder korrekt abgebaut.



Anhang A Gerätezustand

Gerätezustand	Code	Beschreibung
NOT CONNECTED	0	Greifmodul nicht verbunden
NOT INITIALIZED	1	Greifmodul nicht initialisiert
IDLE	2	Betriebsbereit, nicht aktiv
RELEASED	3	Werkstück freigegeben
NO PART	4	Kein Werkstück gefunden
HOLDING	5	Werkstück wird gehalten
OPERATING	6	Sensor betriebsbereit
FAULT	7	Fehlerzustand

Anhang B Statuscodes

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

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

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

© 2026 WEISS ROBOTICS GmbH & Co. KG. Alle Rechte vorbehalten.

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

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