



**GRIPLINK-PLUGIN
FÜR STÄUBLI ROBOTER**
Version 2.0.0



Inhalt

1	Einführung.....	2
1.1	Notation und Symbole	2
1.2	Bestimmungsgemäße Verwendung.....	2
1.3	Systemvoraussetzungen	3
1.4	Lizenzbestimmungen	3
2	Installation des GRIPLINK-Plugins	4
2.1	SIO Socket Anschluss erstellen für den GRIPLINK.....	5
2.2	Verhalten im Fehlerfall	6
3	Befehlsreferenz.....	7
3.1	Verbindung aufbauen – connect()	8
3.2	Gerät aktivieren – enable()	9
3.3	Gerätezustand abfragen – devstate()	10
3.4	Gerät deaktivieren – disable()	12
3.5	Greifmodul referenzieren – home()	13
3.6	Werkstück greifen – grip()	14
3.7	Gleichzeitiges Greifen von Werkstücken – mgrip()	15
3.8	Werkstück freigeben – release().....	16
3.9	Gleichzeitiges Freigeben von Werkstücken – mrelease()	17
3.10	Greifkraftherhaltung steuern – hold()	18
3.11	Ansteuerung der LED-Anzeige – LED()	19
3.12	Abfrage von Positions- und Sensorwerten – value()	20
Anhang A.	Gerätezustand.....	21

1 Einführung

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



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/manuals

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. 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

Zum Betrieb wird eine der folgenden STÄUBLI Robotersteuerungen benötigt:

- CS9 (s8.10.6)

Folgende Roboter-Optionen werden zum Betrieb der Software benötigt:

- Stäubli SRS



Kontaktieren Sie Ihren STÄUBLI 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 Installation des GRIPLINK-Plugins

Zum Betrieb des GRIPLINK Controllers wird das GRIPLINK-Plugin auf der Robotersteuerung benötigt. Zur Installation des GRIPLINK-Plugins folgen Sie den Punkten 1 bis 4.



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

1. Entpacken Sie das zuvor heruntergeladene ZIP-Archiv mit dem GRIPLINK-Plugin in das SRS Verzeichnis der verwendeten Zelle: Staubli > SRS > CellName > ControllerName > usr > usrapp.
2. Öffnen Sie die STAUBLI Robotics Suite:
 - Öffnen Sie die Zelle
 - Gehen Sie zum Zellen-Explorer
 - Wählen Sie die Val3-Applikation aus, die das GRIPLINK-Plugin verwenden soll, dann „Rechtsklick“ > Hinzufügen > Neue Bibliothek > ... > GRIPLINK > GRIPLINK.pjx > Öffnen > Alias angeben > Automatisch Laden > Ok



In diesem Manuel wird als Alias für alle Beispiele GRIPLINK verwendet. Befehle werden über „call Alias:Befehl()“ aufgerufen.

2.1 SIO Socket Anschluss erstellen für den GRIPLINK

In der STAUBLI Robotics Suite unter dem Reiter Startseite > E/A-Verwaltung in dem neuen Fenster auf Sockets > „Rechtsklick“ > Bearbeite E/A-Karte > + > Tcp client > Port. Geben Sie die IP-Adresse des GRIPLINK Controllers als „Server IP“ und das Timeout an (siehe Abbildung 1 für die Default-Werte).

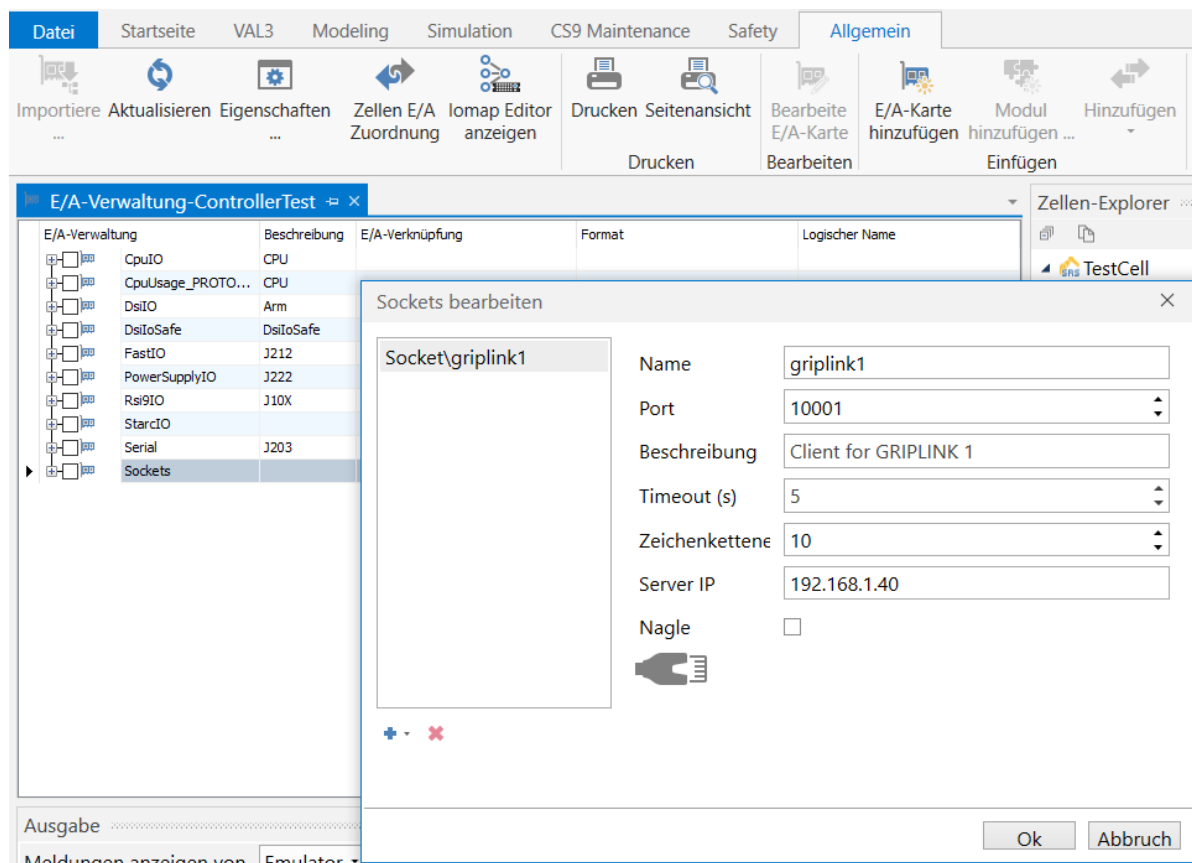


Abbildung 1: Socket-Einstellungen



Die IP-Adresse des GRIPLINK Controllers kann über die GRIPLINK-ET4 Website geändert werden. Standardmäßig sind der Port 10001 und die IP-Adresse 192.168.1.40 eingestellt.



Wenn die Val3-Applikationen aus der STAUBLI Robotics Suite auf die Steuerung geladen werden, kann es sein, dass die SIO-Variable neu über den Applikation Manager verlinkt werden muss. Erstellen Sie dafür einen SIO Socket Anschluss über das Teach Pendant: E/A > Sockets > TCP Client > + > Port und geben Sie die Server IP und das Timeout an. Jetzt im Programm die Globale SIO-Variable verlinken über: ≡ > Applikations Manager > Val3 Applikation > Festplatte > Programm > Globale Variablen > sio > SIO-Variable > Link (F2) > Edit (F6) > Socket > TCP Clients > Socket Name > Ok (F8) > Ok (F8) > Save (F8)

2.2 Verhalten im Fehlerfall

Tritt innerhalb des GRIPLINK-Plugins oder bei der Kommunikation mit dem GRIPLINK Controller ein Fehler auf, so wird grundsätzlich eine Fehlermeldung in den Logger geschrieben. In der Regel werden auch die laufenden Bewegungen des Roboters abgebrochen. Gleiches gilt, wenn das angesprochene Gerät sich im FAULT Zustand befindet oder aufgrund eines Befehls dahin wechselt.

3 Befehlsreferenz

Das GRIPLINK-Plugin stellt dem Anwender eine Sammlung an generischen Bewegungs- und Greiffunktionen sowie Funktionen zum Auslesen angeschlossener Sensoren bereit. Die Befehle sind als Val3-Programm realisiert, welche ihre Parameter als Übergabewerte erhalten. Die Rückgabewerte der Befehle werden in den übergebenen Referenz Variablen gespeichert. Zur Ausführung eines Befehls muss das entsprechende Val3-Programm aufgerufen werden. Einige Befehle stehen sowohl als Einzel- wie auch als Mehrfachbefehle zur Verfügung.

Der prinzipielle Programmablauf mit dem GRIPLINK-Plugin ist stets wie folgt:

1. Verbindung herstellen mit GRIPLINK:connect()
2. Greifmodul oder Sensor aktivieren mit GRIPLINK:enable()
3. Bei Servogreifmodulen ohne Absolutgeber: Greifmodul referenzieren mit GRIPLINK:home()
4. Greifen/Freigeben mit GRIPLINK:grip() bzw. GRIPLINK:release()
5. Position oder Messwert abfragen mit GRIPLINK:value()

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

3.1 Verbindung aufbauen – connect()

Dieser Befehl stellt die Verbindung zwischen GRIPLINK-Plugin und einem Ethernet Socket Anschluss her. Als Übergabeparameter wird die Referenz des GRIPLINK Ethernet Socket Anschlusses als sio-Typ übergeben. Der Befehl hinterlegt den Anschluss in der Bibliothek, damit folgende Befehle darauf zugreifen können. Kann der GRIPLINK über den angegebenen Anschluss nicht erreicht werden, wird eine Log Nachricht geschrieben und die Programmausführung angehalten.

Wenn GRIPLINK-ET4-Befehle vor einem GRIPLINK:connect() ausgeführt werden, wird eine Log Nachricht geschrieben und die Programmausführung angehalten.

Syntax

```
call GRIPLINK:connect(< sio& x_sioSocket >)
```

Parameter

< sio& x_sioSocket > Referenz des GRIPLINK Ethernet Socket Anschlusses

Beispiel

Verbindung zwischen GRIPLINK-plugin und dem GRIPLINK Ethernet Socket Anschluss über die Variable „sioGRIPLINK“ herstellen:

```
call GRIPLINK:connect(sioGRIPLINK)
```

3.2 Gerät aktivieren – enable()

Dieser Befehl aktiviert das am angegebenen Geräte-Port angeschlossene Gerät. GRIPLINK:enable() muss nach GRIPLINK:connect() für alle Geräte ausgeführt werden. Wenn GRIPLINK-Befehle vor einem GRIPLINK:enable() ausgeführt werden, löst dies eine Log Nachricht aus und die Programmausführung wird angehalten.

Syntax

```
call GRIPLINK:enable(< num x_nPort >, < num& x_nDevstate >)
```

Parameter

< x_nPort > Index des Geräte-Ports (0 bis 3)

< num& x_nDevstate > Referenz für den Rückgabewert

Rückgabewert

x_nDevstate aktueller Gerätezustand (siehe Anhang A)

Beispiel

Aktiviere Antrieb und des Greifmoduls an Port 0:

```
call GRIPLINK:enable(0,x_nDevstate)
```

3.3 Gerätezustand abfragen – devstate()

Dieser Befehl gibt den Zustand des ausgewählten Geräts zurück.

Syntax

```
call GRIPLINK:devstate(< num x_nPort >, < num& x_nDevstate >)
```

Parameter

< num x_nPort > Index des Geräte-Ports (0 bis 3)

< num& x_nDevstate > Referenz für den Rückgabewert

Rückgabewert

x_nDevstate aktueller Gerätezustand (siehe Anhang A)

Beispiel

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

```
while x_nDevstate!=4
  call GRIPLINK:devstate(2,x_nDevstate)
  delay(0.01)
endWhile
```

3.4 Gleichzeitiges Abfragen von Gerätezuständen – mdevstate()

Dieser Befehl gibt die Zustände der ausgewählten Greifmodule zurück.

Syntax

```
call GRIPLINK:mdevstate( < num x_nDevstate[1] >)
```

Parameter

< num& x_nDevstate[1] > Referenz Array für die Rückgabewerte

Rückgabewerte

x_nDevstate[0] aktueller Gerätezustand von Greifmodul an Port 0
(siehe Anhang A)

x_nDevstate[1] aktueller Gerätezustand von Greifmodul an Port 1
(siehe Anhang A)

x_nDevstate[2] aktueller Gerätezustand von Greifmodul an Port 2
(siehe Anhang A)

x_nDevstate[3] aktueller Gerätezustand von Greifmodul an Port 3
(siehe Anhang A)

Beispiele

Warte, bis der Gerätezustand von den Greifmodulen an Port 1 und 2 „HOLDING“ (4) ist:

```
while x_nDevstate[1]!=4 or x_nDevstate[2]!=4
  call GRIPLINK:mrelease(x_nDevstates)
  delay(0.01)
endWhile
```

3.5 Gerät deaktivieren – disable()

Deaktiviert das am gewählten Geräte-Port angeschlossene Gerät. Dieser Befehl kann beispielsweise zum Werkzeugwechsel genutzt werden. Das Gerät kann über GRIPLINK:enable() wieder aktiviert werden.

Syntax

```
call GRIPLINK:disable(< num x_nPort >, < num& x_nDevstate >)
```

Parameter

< num x_nPort > Index des Geräte-Ports (0 bis 3)

< num& x_nDevstate > Referenz für den Rückgabewert

Rückgabewert

x_nDevstate aktueller Gerätezustand (siehe Anhang A)

Beispiel

Greifmodul an Port 1 wechseln:

```
//Connect to GRIPLINK  
call GRIPLINK:connect(sioGRIPLINK)  
//Activate gripper 1  
call GRIPLINK:enable(1,x_nDevstate)  
//DO something  
//...  
//Prepare tool change:  
//Disable gripper 1:  
call GRIPLINK:disable(1,x_nDevstate)  
//Operate the tool changer here  
//Activate new gripper:  
call GRIPLINK:enable(1,x_nDevstate)
```

3.6 Greifmodul referenzieren – home()

Referenziert den ausgewählten Servogreifer. Der Befehl führt eine Referenzfahrt des Greifmoduls aus und wartet, bis diese abgeschlossen ist. Nachdem der GRIPLINK:home() Befehl ausgeführt wurde, sind die Finger des Greifmoduls kraftlos und müssen mit einem GRIPLINK:grip() bzw. GRIPLINK:mrip() oder GRIPLINK:release() bzw. GRIPLINK:mrelease() in eine definierte Position verfahren werden.



Die Referenzfahrt kann über die Weboberfläche des GRIPLINK Controllers konfiguriert werden.

Syntax

```
call GRIPLINK:home(< num x_nPort >)
```

Parameter

< num x_nPort > Index des Geräte-Ports (0 bis 3)

Beispiel

Referenziere das Greifmodul an Port 2:

```
call GRIPLINK:home(2)
```

3.7 Werkstück greifen – grip()

Greift mit dem ausgewählten Greifmodul und dem ausgewählten Griff ein Werkstück. Der Befehl wartet, bis der Gerätezustand entweder auf „HOLDING“ oder auf „NO PART“ wechselt.



Die Greifparameter können über die Weboberfläche des GRIPLINK Schnittstellenwandlers konfiguriert werden.

Syntax

```
call GRIPLINK:grip(< num x_nPort >, < num x_nIndex >, < num& x_nDevstate >)
```

Parameter

< num x_nPort >	Index des Geräte-Ports (0 bis 3)
< num x_nIndex >	Ausgewählter Griff (Bereich hängt von Greifermodell ab)
< num& x_nDevstate >	Referenz für den Rückgabewert

Rückgabewert

x_nDevstate	aktueller Gerätezustand (siehe Anhang A)
-------------	--

Beispiele

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

```
while bEnde==false
  call GRIPLINK:grip(0,2,x_nDevstate)
  if x_nDevstate!=5
    //No part, open and try again:
    call GRIPLINK:release(0,2,x_nDevstate)
  else
    //Part gripped!
    bEnde=true
  endif
endwhile
```

3.8 Gleichzeitiges Greifen von Werkstücken – mgrip()

Dieser Befehl führt mit den ausgewählten Greifmodulen einen Griff aus. Der Befehl wartet, bis alle Greifmodule jeweils einen der Zustände „HOLDING“ oder „NO PART“ erreicht haben.



Die Greifparameter können über die Weboberfläche des GRIPLINK Schnittstellenwandlers konfiguriert werden.

Syntax

```
call GRIPLINK:mgrip(< num& x_nPorts[1] >, < num x_nIndex >, < num& x_nDevstate[1] >)
```

Parameter

< num& x_nPorts[1] >	Ausgewählte Greifmodule als Numvektor: x_nPorts[0]: 1 = Greifmodul an Port 0 selektiert, 0 = nicht selektiert x_nPorts[1]: 1 = Greifmodul an Port 1 selektiert, 0 = nicht selektiert x_nPorts[2]: 1 = Greifmodul an Port 2 selektiert, 0 = nicht selektiert x_nPorts[3]: 1 = Greifmodul an Port 3 selektiert, 0 = nicht selektiert
< num x_nIndex >	Ausgewählter Griff (Bereich hängt von Greifermodell ab)
< num& x_nDevstate[1] >	Referenz Array für die Rückgabewerte

Rückgabewerte

x_nDevstate[0]	aktueller Gerätezustand von Greifmodul an Port 0 (siehe Anhang A)
x_nDevstate[1]	aktueller Gerätezustand von Greifmodul an Port 1 (siehe Anhang A)
x_nDevstate[2]	aktueller Gerätezustand von Greifmodul an Port 2 (siehe Anhang A)
x_nDevstate[3]	aktueller Gerätezustand von Greifmodul an Port 3 (siehe Anhang A)

Beispiele

Greifmodule an Port 1 und 2 greifen Werkstück mit Griff 2:

```
L_nPorts[1]=1  
L_nPorts[2]=1  
call GRIPLINK:mgrip(L_nPorts,2,x_nDevstates)
```


3.9 Werkstück freigeben – release()

Gibt das mit dem ausgewählten Greifmodul gegriffene Werkstück wieder frei. Der Befehl wartet, bis das Werkstück freigegeben wurde.



Die Greifparameter können über die Weboberfläche des GRIPLINK Schnittstellenwandlers konfiguriert werden.

Syntax

```
call GRIPLINK:release(< num x_nPort >, < num x_nIndex >, < num& x_nDevstate >)
```

Parameter

< num x_nPort >	Index des Geräte-Ports (0 bis 3)
< num x_nIndex >	Ausgewählter Griff (Bereich hängt von Greifermodell ab)
< num& x_nDevstate >	Referenz für den Rückgabewert

Rückgabewert

x_nDevstate	aktueller Gerätezustand (siehe Anhang A)
-------------	--

Beispiele

Gebe das mit Greifmodul an Port 0 und Griff 2 gegriffene Werkstück wieder frei:

```
call GRIPLINK:release(0,2,x_nDevstate)
```

3.10 Gleichzeitiges Freigeben von Werkstücken – mrelease()

Gibt das mit den ausgewählten Greifmodulen gegriffene Werkstück gleichzeitig wieder frei. Der Befehl wartet, bis alle Greifmodule jeweils den Zustand „RELEASED“ erreicht haben.

Syntax

```
call GRIPLINK:mrelease(< num x_nPorts[1] >, < num x_nIndex >, < num x_nDevstate[1] >)
```

Parameter

< num& x_nPorts[1] >	Ausgewählte Greifmodule als Numvektor: x_nPorts[0]: 1 = Greifmodul an Port 0 selektiert, 0 = nicht selektiert x_nPorts[1]: 1 = Greifmodul an Port 1 selektiert, 0 = nicht selektiert x_nPorts[2]: 1 = Greifmodul an Port 2 selektiert, 0 = nicht selektiert x_nPorts[3]: 1 = Greifmodul an Port 3 selektiert, 0 = nicht selektiert
< num x_nIndex >	Ausgewählter Griff (Bereich hängt von Greifermodell ab)
< num& x_nDevstate[1] >	Referenz Array für die Rückgabewerte

Rückgabewerte

x_nDevstate[0]	aktueller Gerätezustand von Greifmodul an Port 0 (siehe Anhang A)
x_nDevstate[1]	aktueller Gerätezustand von Greifmodul an Port 1 (siehe Anhang A)
x_nDevstate[2]	aktueller Gerätezustand von Greifmodul an Port 2 (siehe Anhang A)
x_nDevstate[3]	aktueller Gerätezustand von Greifmodul an Port 3 (siehe Anhang A)

Beispiele

Greifmodul an Port 0,1 und 2 geben Werkstück mit Griff 3 frei:

```
L_nPorts[0]=1  
L_nPorts[1]=1  
L_nPorts[2]=1  
call GRIPLINK:mrelease(L_nPorts,3,x_nDevstates)
```

3.11 Greifkraftherhaltung steuern – hold()

Die von WEISS ROBOTICS entwickelte innovative Greifkraftsicherung erhält die Greifkraft am Werkstück, auch wenn die Stromzufuhr zum Greifmodul unerwartet unterbrochen wird. Dank der integrierten Absolutsensorik kann die Produktion bei Wiederherstellung der Stromversorgung auch ohne Referenzieren gleich weitergehen. Des Weiteren ermöglicht hold() dauerhaftes Greifen, ohne dass das Greifmodul dabei heiß wird.

Dieser Befehl aktiviert oder deaktiviert die Greifkraftsicherung bei dem ausgewählten Greifmodul.



HOLD ist nicht bei allen Greifmodulen verfügbar.

Syntax

```
call GRIPLINK:hold(< num x_nPort >, < num x_nEnable >)
```

Parameter

< num x_nPort > Index des Geräte-Ports (0 bis 3)

< num x_nEnable > Greifkraftherhaltung: 1 = ein, 0 = aus

Beispiel

Aktiviert die Greifkraftherhaltung bei Greifmodul an Port 2, wenn der Greifstatus „HOLDING“ ist:

```
call GRIPLINK:grip(2,0,x_nDevstate)
if x_nDevstate=5
    call GRIPLINK:hold(2,1)
endif
```

3.12 Ansteuerung der LED-Anzeige – LED()

Dieser Befehl ändert die Farbe und das Muster des Leuchtrings eines selektierten Greifmoduls. Diese Funktion ist ausschließlich für Greifmodule der CRG-Serie von WEISS ROBOTICS verfügbar.



Leuchtmuster können über die Weboberfläche des GRIPLINK Controllers konfiguriert werden.

Syntax

```
call GRIPLINK:LED(< num x_nPort >, < num x_nIndex >)
```

Parameter

< num x_nPort > Index des Geräte-Ports (0 bis 3)

< num x_nIndex > Index des vordefinierten Leuchtmusters

Beispiel

Greife mit dem Greifmodul an Port 3 und ändere die Farbe des Leuchtrings auf das Leuchtmuster 0, wenn die Fingerposition (Value Index 0) danach größer gleich 8,1 mm und auf Leuchtmusters 1, wenn kleiner:

```
call GRIPLINK:grip(3,0,x_nDevstate)
call GRIPLINK:value(3,0,x_nValue)
if x_nValue>=8100
    call GRIPLINK:LED(3,0)
else
    call GRIPLINK:LED(3,1)
endif
```

3.13 Abfrage von Positions- und Sensorwerten – value()

Liest einen Messwert vom angeschlossenen Gerät. Abhängig vom Gerät sind ein oder mehrere Messwerte verfügbar, die über den anzugebenden Index ausgewählt werden können.



Bei Greifmodulen und ähnlichen Aktuatoren entspricht Index 0 immer der Fingerposition/Greifweite in Mikrometern.

Syntax

```
call GRIPLINK:value(< num x_nPort >, < num x_nIndex >, < num x_nValue >)
```

Parameter

< num x_nPort >	Index des Geräte-Ports (0 bis 3)
< num x_nIndex >	Index des abzufragenden Messwerts
< num x_nValue >	Referenz für die Rückgabe des Messwerts

Rückgabewert

x_nValue liefert den Messwert zurück

Beispiel 1

Frage den Abstand zum Werkstück (Messwert Index 0) des an Port 1 angeschlossenen Laserdistanzsensors ab und greife mit dem Greifer an Port 3 zu, wenn der Abstand kleiner als 42 mm ist. Anderenfalls rufe das Unterprogramm „skip“ auf:

```
call GRIPLINK:value(1,0,x_nValue)
if x_nValue<42000
  call GRIPLINK:GRIP(3,0,x_nDevstate)
else
  call skip()
endif
```

Beispiel 2

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

```
call GRIPLINK:value(2,0,x_nValue)
if x_nValue>10500
  call prog()
endif
```

Anhang A. Gerätezustand

Gerätezustand	Code	Beschreibung
NOT CONNECTED	0	Kein Gerät angeschlossen
NOT INITIALIZED	1	Nicht initialisiert
DISABLED	2	Betriebsbereit, aber nicht aktiviert
RELEASED	3	Werkstück freigegeben
NO PART	4	Kein Werkstück gefunden
HOLDING	5	Werkstück wird gehalten
OPERATING	6	Betriebsbereit
FAULT	7	Fehlerzustand

© 2020 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.