



## GRILINK-PLUGIN FÜR EPSON

Version 1.0.0



# Inhalt

1	Einführung.....	2
1.1	Notation und Symbole .....	2
1.2	Bestimmungsgemäße Verwendung.....	2
1.3	Systemvoraussetzungen .....	2
1.4	Lizenzbestimmungen .....	3
2	Installation .....	4
2.1	Installation der Software .....	4
2.2	Netzwerkconfiguration .....	4
2.3	Greifteilerkennung und -überwachung .....	5
2.4	Verhalten im Fehlerfall .....	6
3	Befehlsreferenz.....	7
3.1	Verbindung aufbauen - CONNECT .....	8
3.2	Antrieb des Greifmoduls einschalten - ENABLE.....	9
3.3	Greifzustand abfragen – GET STATE .....	10
3.4	Abfrage des Greifzustands über globale Variablen .....	11
3.5	Antrieb des Greifmoduls deaktivieren - DISABLE .....	12
3.6	Alle angeschlossenen Greifmodule deaktivieren – DISABLE ALL.....	13
3.7	Greifmodul referenzieren - HOME.....	14
3.8	Mehrere Greifmodule referenzieren – MHOME .....	15
3.9	Werkstück greifen - GRIP .....	16
3.10	Gleichzeitiges Greifen von Werkstücken – MGRIP .....	17
3.11	Werkstück freigeben - RELEASE.....	18
3.12	Gleichzeitiges Freigeben von Werkstücken – MRELEASE.....	19
3.13	Aktuelle Fingerposition auslesen – GET POSITION .....	20
3.14	Greifkrafterhaltung steuern - PERMAGRIP .....	21
3.15	Gleichzeitiges Steuern der Greifkrafterhaltung – MPERMAGRIP .....	22
3.16	Ansteuerung der LED-Anzeige – GRIPLINK LED .....	23
Anhang A	Greifzustand.....	24

# 1 Einführung

Mit der GRIPLINK-Technologie können servoelektrische und smart pneumatische Greifmodule von WEISS ROBOTICS über eine Netzwerkverbindung mit Robotersystemen führender Hersteller verbunden werden. Das GRIPLINK-Plugin für EPSON ist das steuerungsseitige Bindeglied und ermöglicht die einfache Einbindung der GRIPLINK-Technologie von WEISS ROBOTICS in Robotersysteme des Herstellers EPSON.



Diese Anleitung beschreibt die Funktionen des GRIPLINK-Plugins. Informationen über Montage, Inbetriebnahme und Betrieb des GRIPLINK Schnittstellenwandlers entnehmen Sie der Betriebsanleitung des jeweiligen Moduls. Diese finden Sie online unter [www.griplink.de/manuals](http://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 Schnittstellenwandler 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 ein folgende EPSON Entwicklungsumgebung benötigt:

- EPSON Robotersteuerung der Baureihen RC700-A und RC90-B
- EPSON Entwicklungsumgebung RC+ 7.0 in Version 7.4.5 oder höher



Kontaktieren Sie EPSON oder ihren EPSON Partner zum Bezug der Roboter-Optionen.



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

### 2.1 Installation der Software

Zum Betrieb des GRIPLINK-ET4 wird das von Weiss Robotics bereit gestellte GRIPLINK-Plugin auf der Robotersteuerung benötigt. Zur Installation des GRIPLINK-Plugins führen Sie folgende Schritte aus:



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

1. Entpacken Sie das zuvor heruntergeladene ZIP-Archiv mit dem GRIPLINK-Plugin in ein Verzeichnis ihrer Wahl
2. Legen Sie in der Roboter-Entwicklungsumgebung EPSON RC+ ein neues Projekt an oder öffnen Sie ein bestehendes Projekt, in dem der GRIPLNK verwendet werden soll.
3. Importieren Sie die entpackten Dateien mittels des Menüpunktes „Datei“ -> „Importieren“ in Ihr neues Projekt. Beachten Sie dabei, dass das Paket Dateien mit den Endungen \*.prg sowie \*.inc enthält. Beide Dateitypen müssen separat importiert werden.
4. Die notwendigen Funktionen für die Verwendung des GRIPLINK-ET4 in Ihrem Projekt stehen nun zur Verfügung.

### 2.2 Netzwerkkonfiguration

Die Verbindung zwischen GRIPLINK-ET4 und Robotersteuerung wird über eine TCP/IP-Netzwerkverbindung hergestellt. Hierfür muss der Robotersteuerung in den Systemeinstellungen eine IP-Adresse zugewiesen werden. Bitte beachten Sie, dass die IP-Adresse der Robotersteuerung im selben Subnetz liegen muss wie die IP-Adresse des GRIPLINK-ET4.

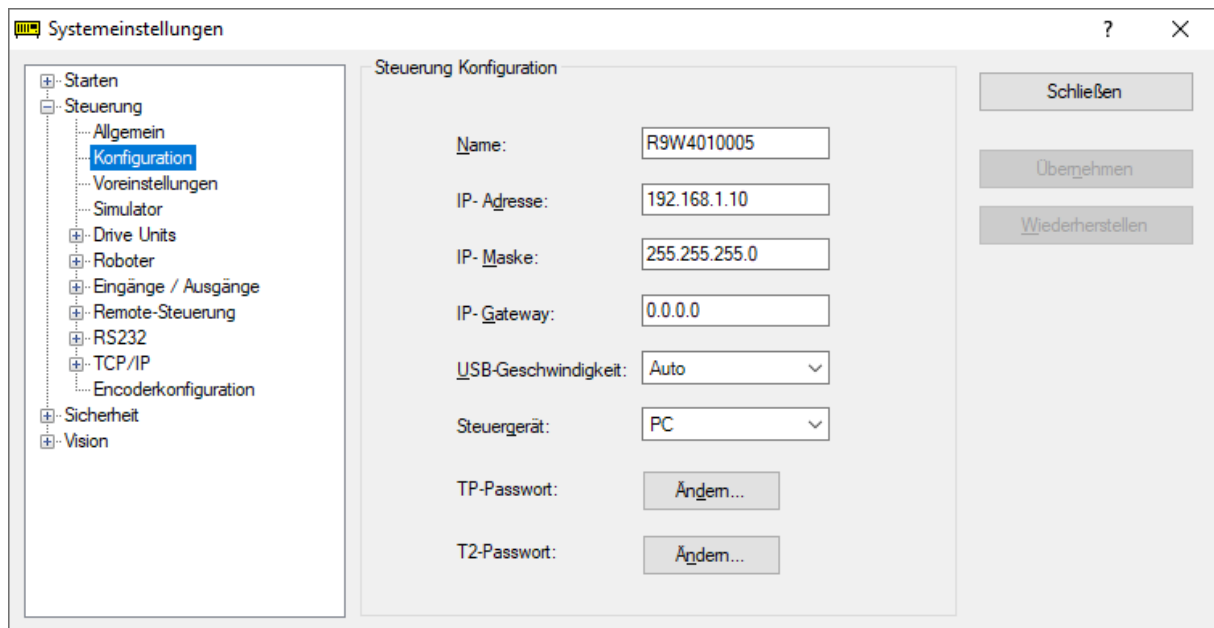
Die IP-Adresse des GRIPLINK ist standardmäßig auf 192.168.1.40 eingestellt und kann über dessen Web-Oberfläche jederzeit angepasst werden. Verbinden Sie dazu den GRIPLINK mit einem PC oder Laptop und öffnen Sie die Web-Oberfläche im Browser ihrer Wahl durch Eingabe der Adresse <http://192.168.1.40>. Die IP-Einstellungen erreichen Sie über den Button „Config“.

Die Befehlschnittstelle des GRIPLINK nimmt eingehende Verbindungen auf Port 10001 (TCP) entgegen.



Weitere Informationen zu den Konfigurationen des GRIPLINKs entnehmen Sie bitte dem zugehörigen Benutzerhandbuch.

Die IP-Adresse der Robotersteuerung kann in der Entwicklungsumgebung EPSON RC+ im Menüpunkt „Einstellungen“ → „Systemeinstellungen“ unter „Steuerung“ → „Konfiguration“ eingestellt werden.



## 2.3 Greifteilerkennung und -überwachung

Greifmodule von WEISS ROBOTICS verfügen über eine integrierte Greifteilerkennung und -überwachung, die es ermöglicht, den gesamten Greifprozess ohne zusätzliche Sensoren permanent zu überwachen und so die Zuverlässigkeit des Handhabungsprozesses wesentlich zu erhöhen.

Das GRIPLINK-Plugin kommuniziert im Hintergrund kontinuierlich mit dem Greifmodul und fragt dessen Zustand permanent ab. Auf diese Weise kann im Roboterprogramm unmittelbar auf nicht vorhandene oder verloren gegangene Werkstücke reagiert werden.

### 2.3.1 Greifteilerkennung

Die Greifteilerkennung ermöglicht es, nach dem Greifen sofort zu erkennen, ob ein Werkstück korrekt gegriffen wurde oder nicht. Sind die Greiffinger innerhalb des vorgegebenen Positionsfensters blockiert und konnte die gewünschte Greifkraft aufgebaut werden, wechselt das Greifmodul in den Zustand HOLDING. Andernfalls wechselt das Greifmodul in den Zustand NO PART. Der Greifzustand nach Ausführung des Griffs wird von der Funktion *Griplink\_Grip()* (vgl. Kapitel 3.9) als Rückgabewert zurück geliefert. Außerdem kann der Greifzustand jederzeit mittels der Funktion *Griplink\_GetState()* (vgl. Kapitel 3.3) abgefragt werden.



Eine detaillierte Beschreibung der Greifzustände finden Sie in der Betriebsanleitung des jeweiligen Greifmoduls.

### 2.3.2 Greifteilüberwachung

Wurde ein Werkstück korrekt gegriffen, d.h. der Zustand HOLDING wurde erreicht, startet auf dem Greifmodul automatisch die integrierte Greifteilüberwachung. Wenn das Werkstück nun verloren geht

oder entnommen wird, bevor der Befehl *Griplink\_Release()* (vgl. Kapitel 3.11) zur Freigabe des Werkstücks ausgeführt wird, wechselt der Zustand des Greifmoduls nach PART LOST.

Das Verhalten des Roboterprogramms im Falle eines Werkstückverlusts kann beim Aktivieren des Greifmoduls mittels des Befehls *Griplink\_Enable()* (vgl. Kapitel 3.2) festgelegt werden. Wird hier die Greifteilüberwachung aktiviert, wird im Falle eines Werkstückverlusts der Roboter angehalten (Befehl *AbortMotion All*) und das Roboterprogramm mittels Fehler gestoppt. Wird die Greifteilüberwachung dagegen nicht aktiviert, wird das GRIPLINK-Plugin den Werkstückverlust ignorieren und ohne Fehlermeldung fortfahren.

## 2.4 Verhalten im Fehlerfall

Tritt innerhalb des GRIPLINK-Plugins ein Fehler auf, so wird das laufende Roboterprogramm grundsätzlich mittels Fehler gestoppt. Dies führt in der Regel dazu, dass laufende Bewegungen des Roboters noch zu Ende gefahren werden, bevor das Programm anhält.

Geht die Verbindung zwischen Robotersteuerung und GRIPLINK oder zwischen GRIPLINK und Greifmodul unerwartet verloren oder tritt auf dem Greifmodul ein schwerer Fehler auf, so werden zusätzlich alle laufenden Roboterbewegungen über den Befehl *AbortMotion All* automatisch angehalten.

Die Fehlernummer für alle durch das GRIPLINK-Plugin ausgelösten Fehler ist auf 8100 vorkonfiguriert. Der Wert kann in der Datei *GriplinkConfig.inc* angepasst werden.

Die folgende Tabelle gibt eine Übersicht über die Fehlerreaktion:

Fehler	Bedingung	Reaktion
Kommunikationsunterbrechung zwischen GRIPLINK Schnittstellenwandler und Roboter	-	Roboter wird angehalten
Kommunikationsunterbrechung zwischen Greifmodul und GRIPLINK Schnittstellenwandler	Greifer aktiviert über ENABLE-Befehl	Roboter wird angehalten
	Greifer nicht aktiviert	keine
Fehler am Greifmodul (Hardwarefehler, Übertemperatur)	Greifer aktiviert über ENABLE-Befehl	Roboter wird angehalten
	Greifer nicht aktiviert	Keine
Greifteilverlust	Greifteilüberwachung aktiv	Roboter wird angehalten
	Greifteilüberwachung nicht aktiv	keine

### 3 Befehlsreferenz

Das GRIPLINK-Plugin stellt dem Anwender eine Sammlung an greifmodulspezifischen Funktionen zur Verfügung. Es stehen sowohl Einzel- wie auch als Mehrfachbefehle zur Verfügung.

#### **Mehrfachbefehle**

Mit den Mehrfachbefehlen (Präfix M) können mehrere Greifmodule gleichzeitig parallel angesprochen werden. Diese Befehle eignen sich insbesondere für die Handhabung großer oder biegeschlaffer Werkstücke mit mehreren Greifmodulen.

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

1. Verbindung herstellen mit CONNECT
2. Greifmodul und Verbindungsüberwachung aktivieren mit ENABLE
3. Bei Servogreifmodulen ohne Absolutgeber: Greifmodul referenzieren mit HOME/MHOME
4. Greifen/Freigeben mit GRIP/MGRIP bzw. RELEASE/MRELEASE

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

Befehl	Beschreibung
Griplink_Connect()	Verbindung aufbauen
Griplink_Enable()	Greifmodul aktivieren
Griplink_Disable()	Greifmodul deaktivieren
Griplink_DisableAll()	Alle angeschlossenen Greifmodule deaktivieren
Griplink_GetState()	Greifzustand abfragen
Griplink_Home()	Greifmodul referenzieren
Griplink_MulitHome	Mehrere Greifmodule gleichzeitig referenzieren
Griplink_Grip()	Werkstück greifen
Griplink_MultiGrip()	Gleichzeitiges Greifen von mehreren Werkstücken
Griplink_Release()	Werkstück freigeben
Griplink_MultiRelease()	Gleichzeitiges Freigeben von mehreren Werkstücken
Griplink_GetPos()	Aktuelle Fingerposition auslesen
Griplink_Permagrip()	Greifkrafterhaltung PERMAGRIP® steuern
Griplink_MultiPermagrip()	Greifkrafterhaltung gleichzeitg steuern
Griplink_LED()	LED-Anzeige steuern



### 3.1 Verbindung aufbauen - CONNECT

Dieser Befehl stellt die Verbindung zwischen GRIPLINK Schnittstellenwandler und der Robotersteuerung her. Für die Kommunikation muss einer von 16 zur Verfügung stehenden Ports (Nr. 201 bis 216) auf der Robotersteuerung ausgewählt werden. Um die Verbindung sowie den Zustand der angeschlossenen Greifmodule permanent überwachen zu können, wird auf der Robotersteuerung ein eigener Task gestartet, deren Identifikationsnummer aus dem zulässigen Wertebereich frei gewählt werden kann.

Darüber hinaus werden jeweils ein Timer und ein SyncLock der Robotersteuerung verwendet. Auch diese können über die übergebene ID jeweils frei gewählt werden.



Zu beachten ist, dass sowohl der gewählte Port als auch Task-ID, Timer-ID und SyncLock-ID exklusiv für die Kommunikation mit dem GRIPLINK zur Verfügung stehen müssen und an keiner anderen Stelle im Roboterprogramm verwendet werden dürfen.



Die IP-Adresse des GRIPLINK kann über die Weboberfläche geändert werden.

#### **Signatur**

```
Function Griplink_Connect(strIPAddr$ As String, intControllerPortNo As Integer, intTaskId As Integer, intTimerId As Integer, intSyncLockId As Integer)
```

#### **Parameter**

strIPAddr\$	IP-Adresse des Griplink als String
intControllerPortNumber	Port-Nummer der vorkonfigurierten TCP/IP-Verbindung. Muss zwischen 201 und 216 liegen.
intTaskId	Task ID, unter der der Task für die Verbindungsverwaltung gestartet werden soll. Muss zwischen 1 und 31 liegen. In der Regel ist Task ID 1 bereits durch das Hauptprogramm belegt, so dass mindestens Task ID 2 verwendet werden muss.
intTimerId	Timer ID des zu verwendenden Timers. Muss zwischen 0 und 63 liegen.
intSyncLockId	Lock ID des zu verwendenden SyncLock. Muss zwischen 0 und 63 liegen.

#### **Rückgabewert**

-

#### **Beispiel**

Verbindung zwischen Robotersteuerung und GRIPLINK mit der IP-Adresse 192.168.0.40 herstellen unter Verwendung von TCP/IP-Port Nr. 201. Task ID: 2, Timer ID: 0 und Lock ID: 0.

```
Griplink_Connect("192.168.0.40", 201, 2, 0, 0)
```

## 3.2 Antrieb des Greifmoduls einschalten - ENABLE

Dieser Befehl aktiviert das Greifmodul und die Verbindungsüberwachung. Wird die Verbindung zum Greifmodul getrennt (z.B. durch einen Kabelbruch), löst dies einen Fehler aus und das Roboterprogramm wird gestoppt. Darüber hinaus werden alle laufenden Roboterbewegungen angehalten. Die Verbindungsüberwachung kann über den Befehl *Griplink\_Disable()* (vgl. Abschnitt 3.4) deaktiviert werden, um z. B. einen Werkzeugwechsel durchzuführen.

### Greifteilüberwachung

Über einen Parameter kann die Greifteilüberwachung ein- bzw. ausgeschaltet werden. Ist die Greifteilüberwachung aktiv und verliert das Greifmodul ein zuvor aufgenommenes Werkstück, löst dies einen Fehler aus und das Roboterprogramm wird gestoppt. Zudem werden in diesem Fall alle in Bewegung befindlichen Roboter über den Befehl *AbortMotion All* unmittelbar angehalten. Ist dieses Verhalten unerwünscht, muss die Greifteilüberwachung deaktiviert werden.

*Griplink\_Enable()* muss nach *Griplink\_Connect()* für alle angeschlossenen Greifmodule ausgeführt werden. Wenn GRIPLINK-Befehle vor der Ausführung von *Griplink\_Enable()* ausgeführt werden, löst dies einen Fehler aus und das Roboterprogramm wird gestoppt.

### **Signatur**

```
Function Griplink_Enable(ubGripperID As UByte, boolEnableGripMonitoring  
As Boolean) As Integer
```

### **Parameter**

ubGripperID	Index des Greifmoduls (0 bis 3)
boolEnableGripMonitoring	Greifteilüberwachung: <i>True</i> = an, <i>False</i> = aus

### **Rückgabewert**

Aktueller Greifzustand (siehe Anhang A)

### **Beispiel**

Aktiviere Antrieb und Greifteilüberwachung von Greifmodul an Port 0:

```
Integer intGripState  
intGripState = Griplink_Enable(0, True)
```

### 3.3 Greifzustand abfragen – GET STATE

Diese Funktion liefert den Greifzustand des ausgewählten Greifmoduls zurück. Der Greifzustand ist als Integer-Wert codiert. Zur Vereinfachung und Verbesserung der Lesbarkeit sind in der Datei *Griplink.inc* Konstanten definiert, die für die Verarbeitung von Greifzuständen verwendet werden können.

#### **Signatur**

```
Function Griplink_GetState(ubGripperID As UByte) As Integer
```

#### **Parameter**

ubGripperID            Index des Greifmoduls (0 bis 3)

#### **Rückgabewert**

Aktueller Greifzustand (siehe Anhang A)

#### **Beispiel**

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

```
#include "Griplink.inc"  
'Greifzustand von Greifmodul an Port 2 abfragen  
Integer intGripState  
'Die Konstante GS_HOLDING ist in Griplink.inc definiert  
Do While intGripState <> GS_HOLDING  
    intGripState = Griplink_GetState(2)  
Loop
```

### 3.4 Abfrage des Greifzustands über globale Variablen

Das GRIPLINK-Plugin stellt vier globale Zustandsvariablen bereit, die bei angeschlossenem und mittels *Griplink\_Enable()* aktiviertem Greifmodul permanent den Greifzustand des jeweiligen Greifmoduls als Ganzzahlwert (Integer) bereit stellen:

1. *Griplink\_intGripperState0* - Greifzustand des Greifmoduls an Port 0
2. *Griplink\_intGripperState1* - Greifzustand des Greifmoduls an Port 1
3. *Griplink\_intGripperState2* - Greifzustand des Greifmoduls an Port 2
4. *Griplink\_intGripperState3* - Greifzustand des Greifmoduls an Port 3

Diese globalen Variablen können z. B. dazu verwendet werden, laufende Bewegungen des Roboters bei Werkstückverlust oder fehlerhaftem Greifmodul zu unterbrechen. Insbesondere wenn die integrierte Greifteilüberwachung (vgl. Kapitel 2.3.2) nicht zur Anwendung kommt, kann auf diese Weise mit Hilfe der Steuerungsbefehle *Sense* und *Till* (vgl. SPEL+ Referenzhandbuch des EPSON-Robotersystems) kundenspezifisches Verhalten bei Werkstückverlust implementiert werden.

#### **Beispiel**

Das folgende Programm unterbricht die mittels *Jump P4* durchgeführte Bewegung, wenn die globale Variable *Griplink\_intGripperState0*, die permanent den Greifzustand des Greifmoduls an Port 0 enthält, einen anderen Wert als *GS\_HOLDING* annimmt. Ob die Bewegung zu Ende geführt oder unterbrochen wurde, kann anschließend mittels des Befehls *JS* festgestellt werden. Wurde die Bewegung unterbrochen, so wird im Beispiel der Fehler 8101 ausgelöst.

```
Sense Griplink_intGripperState0 <> GS_HOLDING
Jump P4 Sense
If JS = True Then
  Error 8101
EndIf
```



Weitere Informationen zu den Befehlen *Jump* und *Sense* entnehmen Sie bitte der Dokumentation des EPSON-Robotersystems.

### 3.5 Antrieb des Greifmoduls deaktivieren - DISABLE

Diese Funktion kann beispielsweise zum Werkzeugwechsel genutzt werden. Sie deaktiviert den Antrieb des ausgewählten Greifmoduls. Ist der Antrieb deaktiviert, ist auch die Verbindungsüberwachung ausgeschaltet und eine Verbindungsunterbrechung zwischen GRIPLINK und Greifmodul führt nicht mehr zu einem Fehler. Die Verbindungsüberwachung kann über *Griplink\_Enable()* wieder aktiviert werden.

#### **Signatur**

```
Function Griplink_Disable(ubGripperID As UByte) As Integer
```

#### **Parameter**

ubGripperID            Index des Greifmoduls (0 bis 3)

#### **Rückgabewert**

Aktueller Greifzustand (siehe Anhang A)

#### **Beispiel**

Greifmodul an Port 1 wechseln:

```
'Verbindung zwischen Roboter und dem GRIPLINK herstellen
Griplink_Connect("192.168.1.40", 201, 2, 0, 0)
'Aktiviere Antrieb und Greifteilüberwachung von Greifmodul an Port 1
Griplink_Enable(1, True)
'Programm ausführen (z. B. Pick & Place)
'...
'Deaktivieren von Greifmodul an Port 1
Griplink_Disable(1)
'Jetzt kann das Greifmodul mechanisch ausgetauscht werden.
'Aktivieren des neuen Greifmoduls.
Griplink_Enable(1, True)
'Programm weiter ausführen
```

### 3.6 Alle angeschlossenen Greifmodule deaktivieren – DISABLE ALL

Diese Funktion deaktiviert den Antrieb aller angeschlossenen Greifmodule. Er kann z. B. im Falle eines Notstopps verwendet werden, um alle Greifmodule kraftlos zu schalten. Ist der Antrieb deaktiviert, ist auch die Verbindungsüberwachung ausgeschaltet und eine Verbindungsunterbrechung zwischen GRIPLINK und Greifmodul führt nicht zu einem Fehler. Die Verbindungsüberwachung kann über *Griplink\_Enable()* (vgl. Abschnitt 3.2) wieder aktiviert werden.

#### **Signatur**

```
Function Griplink_DisableAll As Boolean
```

#### **Parameter**

-

#### **Rückgabewert**

*True* bei erfolgreicher Ausführung

*False* bei Fehler

#### **Beispiel**

Deaktivieren des Antriebs aller Greifmodule bei Notstopp:

```
'Funktion, die bei Notstopp ausgeführt werden soll  
Function EmergencyStop  
    Print "Emergency stop, disabling all grippers"  
    Griplink_DisableAll()  
Fend  
'Führe bei Notstopp die Funktion EmergencyStop aus  
Trap Emergency Xqt EmergencyStop
```

### 3.7 Greifmodul referenzieren - HOME

Referenziert den ausgewählten Servogreifer. Der Befehl führt eine Referenzfahrt des Greifmoduls aus und wartet, bis diese abgeschlossen ist. Nachdem der HOME-Befehl ausgeführt wurde, sind die Finger des Greifmoduls kraftlos und müssen mit *Griplink\_Grip()* / *Griplink\_MultiGrip()* oder *Griplink\_Release()* / *Griplink\_MultiRelease()* in eine definierte Position verfahren werden.



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

#### **Signatur**

```
Function Griplink_Home(ubGripperID As UByte) As Integer
```

#### **Parameter**

ubGripperID            Index des Greifmoduls (0 bis 3)

#### **Rückgabewert**

Aktueller Greifzustand (siehe Anhang A)

#### **Beispiel**

Referenziere Greifmodul an Port 3:

```
#include "Griplink.inc"
Integer intGripState
'Referenziere Greifmodul an Port 3
intGripState = Griplink_Home(3)
If intGripState = GS_IDLE Then
    'Greifmodul ist referenziert
Else
    'Greifmodul ist nicht referenziert
EndIf
```

### 3.8 Mehrere Greifmodule referenzieren – MHOME

Referenziert die ausgewählten Greifmodule. Die Funktion führt bei allen ausgewählten Greifmodulen eine Referenzfahrt aus und wartet, bis diese abgeschlossen ist. Nachdem eine Referenzierungsfahrt ausgeführt wurde, sind die Finger des Greifmoduls kraftlos und müssen mit *Griplink\_Grip()* / *Griplink\_MultiGrip()* oder *Griplink\_Release()* / *Griplink\_MultiRelease()* in eine definierte Position verfahren werden.



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

#### Signatur

```
Function Griplink_MultiHome(boolGripper0 As Boolean, boolGripper1 As Boolean, boolGripper2 As Boolean, boolGripper3 As Boolean)
```

#### Parameter

boolGripper0	<i>True</i>	Greifmodul an Port 0 ist selektiert und wird referenziert
	<i>False</i>	Greifmodul an Port 0 ist nicht selektiert
boolGripper1	<i>True</i>	Greifmodul an Port 1 ist selektiert und wird referenziert
	<i>False</i>	Greifmodul an Port 1 ist nicht selektiert
boolGripper2	<i>True</i>	Greifmodul an Port 2 ist selektiert und wird referenziert
	<i>False</i>	Greifmodul an Port 2 ist nicht selektiert
boolGripper3	<i>True</i>	Greifmodul an Port 3 ist selektiert und wird referenziert
	<i>False</i>	Greifmodul an Port 3 ist nicht selektiert

#### Rückgabewert

-

#### Beispiele

Referenziere gleichzeitig die Greifmodule 0 und 2:

```
'Referenziere Greifer an Port 0 und 2  
Griplink_MultiHome(True, False, True, False)
```



### 3.9 Werkstück greifen - GRIP

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



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

#### **Signatur**

```
Function Griplink_Grip(ubGripperID As UByte, ubGripIndex As UByte) As Integer
```

#### **Parameter**

ubGripperID	Index des Greifmoduls (0 bis 3)
ubGripIndex	Ausgewählter Griff (0 bis 3 bzw. 0 bis 7 bei CRG Greifmodulen)

#### **Rückgabewert**

Aktueller Greifzustand (siehe Anhang A)

#### **Beispiele**

Greifmodul an Port 0 soll Griff 2 ausführen. Wenn kein Teil gefunden wurde soll das Programm eine Nachricht ausgeben:

```
#include Griplink.inc
Integer intGripState
intGripState = Griplink_Grip(0, 2)
'Die Konstanten GS_NO_PART und GS_HOLDING sind in Griplink.inc
'definiert
If intGripState = GS_NO_PART Then
    Print "No part found"
ElseIf intGripState <> GS_HOLDING Then
    Print "Unexpected grip state"
EndIf
```

### 3.10 Gleichzeitiges Greifen von Werkstücken – MGRIP

Diese Funktion führt mit den ausgewählten Greifmodulen einen Greifbefehl aus. Die Funktion 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.

#### Signatur

```
Function Griplink_MultiGrip(boolGripper0 As Boolean, boolGripper1 As Boolean, boolGripper2 As Boolean, boolGripper3 As Boolean, ubGripIndex As UByte)
```

#### Parameter

boolGripper0	<i>True</i>	Greifmodul an Port 0 ist selektiert und wird referenziert
	<i>False</i>	Greifmodul an Port 0 ist nicht selektiert
boolGripper1	<i>True</i>	Greifmodul an Port 1 ist selektiert und wird referenziert
	<i>False</i>	Greifmodul an Port 1 ist nicht selektiert
boolGripper2	<i>True</i>	Greifmodul an Port 2 ist selektiert und wird referenziert
	<i>False</i>	Greifmodul an Port 2 ist nicht selektiert
boolGripper3	<i>True</i>	Greifmodul an Port 3 ist selektiert und wird referenziert
	<i>False</i>	Greifmodul an Port 3 ist nicht selektiert
ubGripIndex		Ausgewählter Griff (0 bis 3 bzw. 0 bis 7 bei CRG Greifmodulen)

#### Rückgabewert

-

#### Beispiele

Greifmodul an Port 1 und 2 greifen Werkstück mit Griff 2 und prüfen, ob ein Teil gehalten wird:

```
#include Griplink.inc
Griplink_MultiGrip(False, True, True, False, 2)
'Die Konstanten GS_NO_PART und GS_HOLDING sind in Griplink.inc
'definiert
If Griplink_GetState(1) = GS_HOLDING And Griplink_GetState(2) =
GS_HOLDING Then
    Print "Holding parts"
EndIf
```

### 3.11 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.

#### **Signatur**

```
Function Griplink_Release(ubGripperID As UByte, ubGripIndex As UByte) As Integer
```

#### **Parameter**

ubGripperID	Index des Greifmoduls (0 bis 3)
ubGripIndex	Ausgewählter Griff (0 bis 3 bzw. 0 bis 7 bei CRG Greifmodulen)

#### **Rückgabewert**

Aktueller Greifzustand (siehe Anhang Anhang A)

#### **Beispiele**

Greifmodul an Port 0 soll ein mit Griff 2 gehaltenes Werkstück freigeben:

```
#include Griplink.inc
Integer intGripState
intGripState = Griplink_Release(0, 2)
'Die Konstante GS_RELEASED ist in Griplink.inc
'definiert
If intGripState = GS_RELEASED Then
    Print "Part released"
EndIf
```

### 3.12 Gleichzeitiges Freigeben von Werkstücken – MRELEASE

Gibt die mit den ausgewählten Greifmodulen gegriffenen Werkstück wieder frei. Die Funktion wartet, bis alle Greifmodule jeweils den Zustand „RELEASED“ erreicht haben.

#### **Signatur**

```
Function Griplink_MultiRelease(boolGripper0 As Boolean, boolGripper1 As Boolean, boolGripper2 As Boolean, boolGripper3 As Boolean, ubGripIndex As UByte)
```

#### **Parameter**

boolGripper0	<i>True</i>	Greifmodul an Port 0 ist selektiert und wird referenziert
	<i>False</i>	Greifmodul an Port 0 ist nicht selektiert
boolGripper1	<i>True</i>	Greifmodul an Port 1 ist selektiert und wird referenziert
	<i>False</i>	Greifmodul an Port 1 ist nicht selektiert
boolGripper2	<i>True</i>	Greifmodul an Port 2 ist selektiert und wird referenziert
	<i>False</i>	Greifmodul an Port 2 ist nicht selektiert
boolGripper3	<i>True</i>	Greifmodul an Port 3 ist selektiert und wird referenziert
	<i>False</i>	Greifmodul an Port 3 ist nicht selektiert
ubGripIndex		Ausgewählter Griff (0 bis 3 bzw. 0 bis 7 bei CRG Greifmodulen)

#### **Rückgabewert**

-

#### **Beispiele**

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

```
#include Griplink.inc  
Griplink_MultiRelease(True, True, True, False, 3)
```

### 3.13 Aktuelle Fingerposition auslesen – GET POSITION

Dieser Befehl gibt die aktuelle Fingerposition des ausgewählten Greifmoduls zurück.

#### **Signatur**

```
Function Griplink_GetPos(ubGripperID As UByte) As Real
```

#### **Parameter**

ubGripperID          Index des Greifmoduls (0 bis 3)

#### **Rückgabewert**

Fingerposition in mm

#### **Beispiel**

Lese die Fingerposition von Greifer 0 aus:

```
Real rlPos  
rlPos = Griplink_GetPos(0)
```

### 3.14 Greifkraftherhaltung steuern - PERMAGRIP

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 Referenzierungsfahrt direkt fortgesetzt werden. Des Weiteren ermöglicht PERMAGRIP dauerhaftes Greifen, ohne dass das Greifmodul dabei überhitzen kann.

Dieser Befehl aktiviert oder deaktiviert die PERMAGRIP-Greifkraftherhaltung bei dem ausgewählten Greifmodul.



PERMAGRIP ist nicht bei allen Greifmodulen verfügbar.

#### **Signatur**

```
Function Griplink_Permagrip(ubGripperID As UByte, boolEnable As Boolean)
```

#### **Parameter**

ubGripperID	Index des Greifmoduls (0 bis 3)
boolEnable	PERMAGRIP aktivieren ( <i>True</i> ) oder deaktivieren ( <i>False</i> )

#### **Rückgabewert**

-

#### **Beispiel**

Greife Werkstück mit Griff 0 auf Greifer 2. Aktiviere PERMAGRIP, wenn ein Teil gehalten wird:

```
#include Griplink.inc
Integer intGripState
'Teil greifen
intGripState = Griplink_Grip(2, 0)
'Die Konstante GS_HOLDING ist in Griplink.inc
'definiert
If intGripState = GS_HOLDING Then
    'PERMAGRIP aktivieren
    Griplink_Permagrip(2, True)
EndIf
```

### 3.15 Gleichzeitiges Steuern der Greifkraftherhaltung – MPERMAGRIP

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 PERMAGRIP dauerhaftes Greifen, ohne dass das Greifmodul dabei überhitzen kann.

Dieser Befehl aktiviert oder deaktiviert PERMAGRIP bei den ausgewählten Greifmodulen.



PERMAGRIP® ist nicht bei allen Greifmodulen verfügbar.

#### Signatur

```
Function Griplink_MultiPermagrip(boolGripper0 As Boolean, boolGripper1  
As Boolean, boolGripper2 As Boolean, boolGripper3 As Boolean, boolEnable  
As Boolean)
```

#### Parameter

boolGripper0	<i>True</i>	Greifmodul an Port 0 ist ausgewählt
	<i>False</i>	Greifmodul an Port 0 ist nicht ausgewählt
boolGripper1	<i>True</i>	Greifmodul an Port 1 ist ausgewählt
	<i>False</i>	Greifmodul an Port 1 ist nicht ausgewählt
boolGripper2	<i>True</i>	Greifmodul an Port 2 ist ausgewählt
	<i>False</i>	Greifmodul an Port 2 ist nicht ausgewählt
boolGripper3	<i>True</i>	Greifmodul an Port 3 ist ausgewählt
	<i>False</i>	Greifmodul an Port 3 ist nicht ausgewählt
boolEnable		PERMAGRIP aktivieren ( <i>True</i> ) oder deaktivieren ( <i>False</i> )

#### Rückgabewerte

keine

#### Beispiele

Deaktiviere PERMAGRIP bei Greifmodul an Port 0 und 1:

```
#include Griplink.inc  
'PERMAGRIP deaktivieren  
Griplink_MultiPermagrip(True, True, False, False, False)
```

### 3.16 Ansteuerung der LED-Anzeige – GRIPLINK LED

Dieser Befehl ändert die Farbe und das Muster des Leuchtrings eines selektierten CRG Greifmoduls.



Leuchtmuster können über die Weboberfläche des GRIPLINK Schnittstellenwandlers konfiguriert werden.



Diese Funktion ist ausschließlich für Greifmodule der CRG-Serie verfügbar.

#### Signatur

```
Function Griplink_LED(ubGripperID As UByte, ubLEDIndex As UByte)
```

#### Parameter

ubGripperID	Index des Greifmoduls (0 bis 3)
ubLEDIndex	Index des vordefinierten Leuchtmusters (0 bis 7)

#### Rückgabewerte

keine

#### Beispiel

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

```
#include "Griplink.inc"
Integer intGripState
Real rlPos
intGripState = Griplink_Grip(3, 0)
If intGripState = GS_HOLDING Then
    rlPos = Griplink_GetPos(3)
    If rlPos > 8.1 Then
        'Setze LED-Muster 0
        Griplink_LED(0)
    Else
        'Setze LED-Muster 1
        Griplink_LED(1)
    EndIf
Else
    Print "Failed to grip part"
EndIf
```



## Anhang A Greifzustand

Greifzustand	Konstante	Code	Beschreibung
NOT CONNECTED	GS_NOT_CONNECTED	0	Greifmodul nicht verbunden
NOT INITIALIZED	GS_NOT_INITIALIZED	1	Greifmodul nicht initialisiert
IDLE	GS_IDLE	2	Betriebsbereit, nicht aktiv
RELEASED	GS_RELEASED	3	Werkstück freigegeben
NO PART	GS_NO_PART	4	Kein Werkstück gefunden
HOLDING	GS_HOLDING	5	Werkstück wird gehalten
PART LOST	GS_PART_LOST	6	Werkstück verloren
FAULT	GS_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.