



## GRIPLINK-PLUGIN FÜR DENSO

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	Vorbereitung des Roboters.....	4
2.2	Installation der Software .....	6
2.3	Verhalten im Fehlerfall .....	7
3	Befehlsreferenz.....	8
3.1	Einbinden des Plugins - #Include .....	9
3.2	Verbindung aufbauen - Connect.....	10
3.3	Gerät aktivieren - Enable .....	11
3.4	Greifzustand abfragen – GetState .....	13
3.5	Gerät deaktivieren - Disable .....	13
3.6	Greifmodul referenzieren - Home .....	15
3.7	Werkstück greifen - Grip.....	16
3.8	Gleichzeitiges Greifen von Werkstücken - MGRIP .....	17
3.9	Werkstück freigeben - Release .....	18
3.10	Gleichzeitiges Freigeben von Werkstücken - MRelease .....	19
3.11	Greifrezepte parametrieren - SetGripCfg .....	20
3.12	Gerätetypen verifizieren - DevAssert .....	21
3.13	Abfrage von Positions- und Sensorwerten - Value .....	21
3.14	Greifkraftherhaltung steuern - Clamp .....	23
3.15	Ansteuerung der LED-Anzeige - LED .....	26
3.16	Beenden der Verbindung von GRIPLINK und Roboter - Disconnect.....	11
Anhang A	Greifzustand.....	27

# 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 Denso ist das steuerungsseitige Bindeglied und ermöglicht die einfache Einbindung der GRIPLINK-Technologie von WEISS ROBOTICS in Robotersysteme des Herstellers Denso.



Diese Anleitung beschreibt die Funktionen des GRIPLINK-Plugins. Informationen über Montage, Inbetriebnahme und Betrieb des GRIPLINK Controllers entnehmen Sie der Betriebsanleitung des jeweiligen Controller-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 Controller von WEISS ROBOTICS und einer Denso 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 Denso Robotersteuerungen benötigt:

- RC8
- RC8A
- RC9



Die IP-Adresse des GRIPLINK Controllers muss im selben Subnetz liegen wie die Robotersteuerung. Die Änderung der IP-Adresse des GRIPLINK Controllers erfolgt über dessen Web-Oberfläche und ist in der Betriebsanleitung ausführlich beschrieben.

## **1.4 Lizenzbestimmungen**

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

## 2 Installation

### 2.1 Vorbereitung des Roboters

Um die Kommunikation zwischen dem Roboter und dem GRIPLINK Controller zu ermöglichen, muss ein sog. Client-Channel der Denso-Robotersteuerung konfiguriert werden. Die Einstellungen für die Datenkommunikation können über das Teach Pendant vom Hauptbildschirm aus im Menüpunkt „Einstellungen“ (F6) -> „Kommunik. Bedienrechte“ (F5) -> „Datenkommu.“ (F3) vorgenommen werden (Abbildung 1).

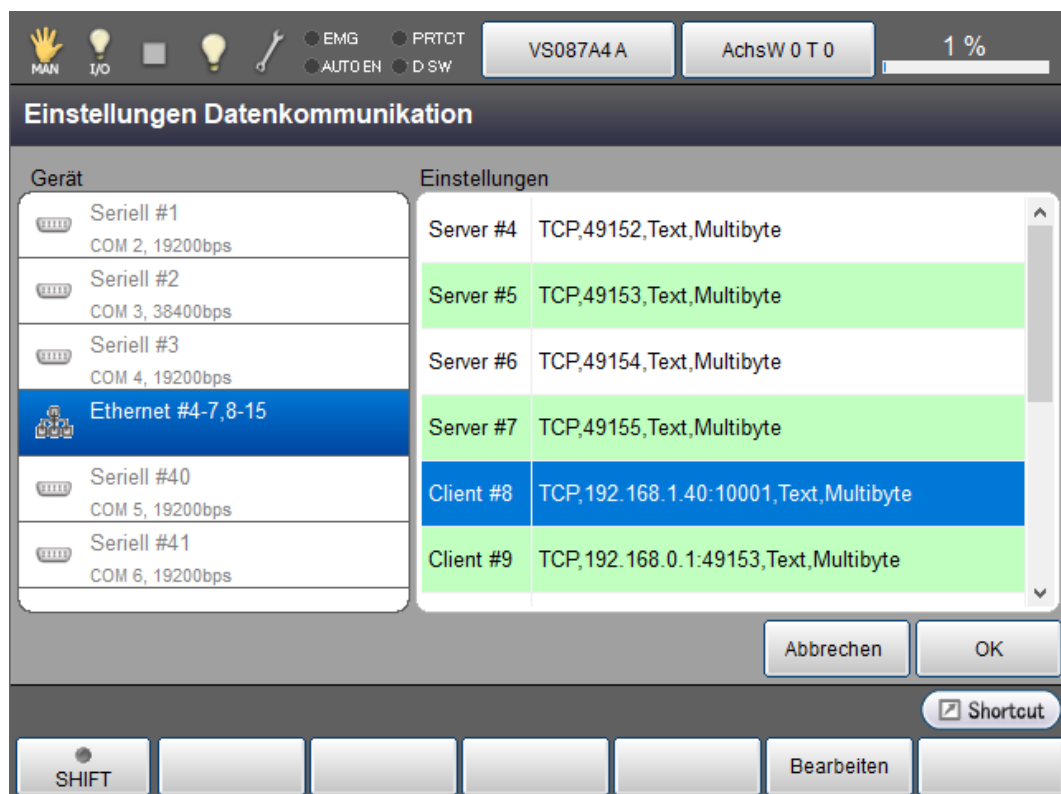


Abbildung 1: Einstellungen Datenkommunikation auf der Robotersteuerung

Für einen beliebigen Client-Port muss die IP-Adresse des GRIPLINK Controllers eingetragen werden. Diese ist standardmäßig auf 192.168.1.40 eingestellt, kann aber über die Web-Oberfläche des GRIPLINK Controllers angepasst werden.

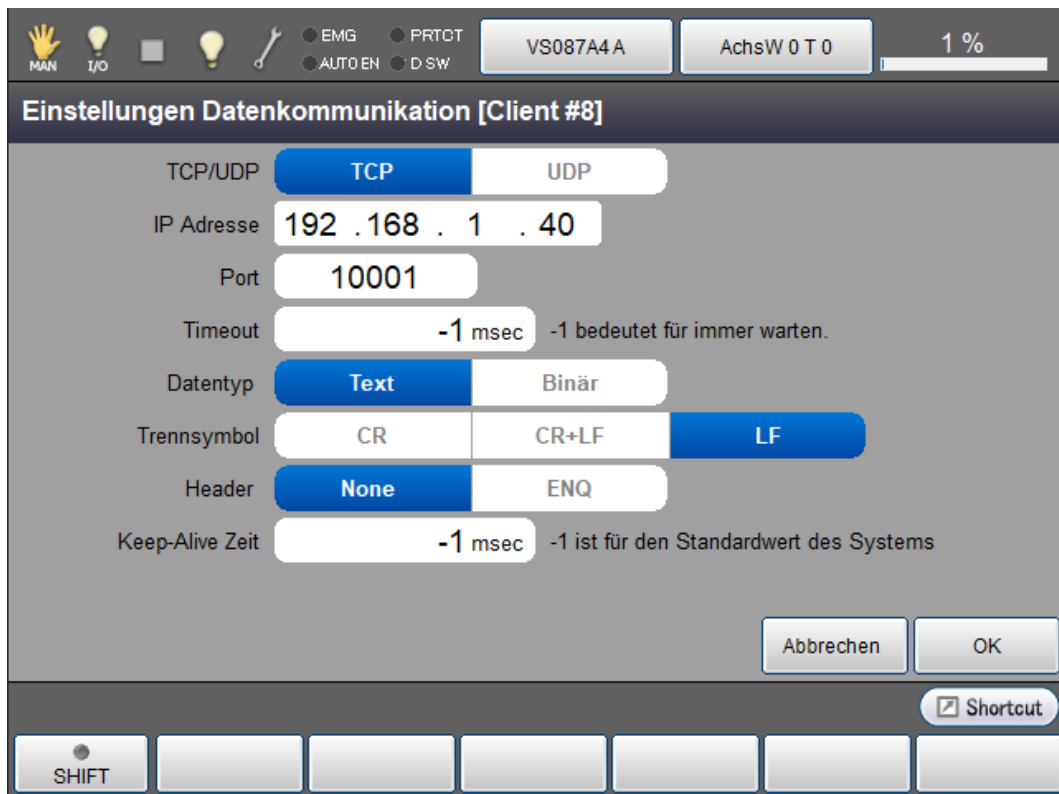


Abbildung 2: Einstellungen auf der Robotersteuerung zur Verbindung mit dem GRIPLINK Controller

Durch Anwählen von „Bearbeiten“ in Abbildung 1 können die Schnittstelleneigenschaften angepasst werden. Abbildung 2 zeigt exemplarisch die notwendigen Einstellungen zur Verbindung der Robotersteuerung mit dem GRIPLINK Controller. Die IP-Adresse entspricht der Einstellung des GRIPLINK Controllers im Auslieferungszustand und muss bei Änderung der Adresse am GRIPLINK Controller entsprechend geändert werden. Port-Nummer 10001, Datentyp „Text“ sowie Trennsymbol „LF“ sind unveränderlich und müssen in jedem Fall auf die angegebenen Werte eingestellt werden. Der Timeout-Wert kann für die jeweilige Anwendung individuell angepasst werden, allerdings verfügt das Plugin selbst über Timeout-Grenzen (3 Sekunden für die Kommunikation und 10 Sekunden für das Abschließen operativer Aufgaben nach Annahme des Befehls).

## 2.2 Installation der Software



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.

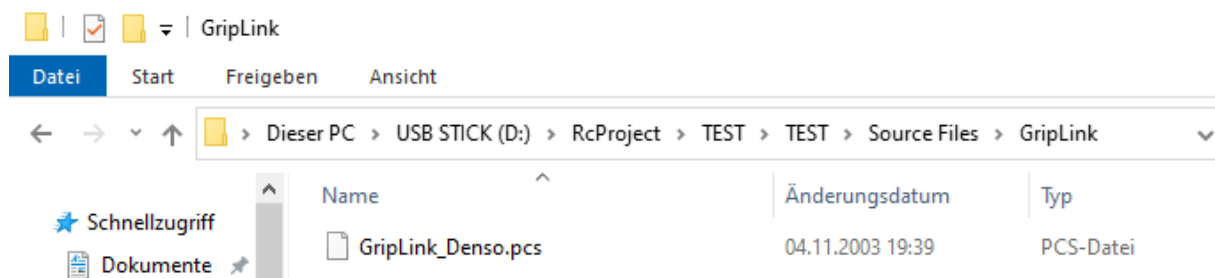
Das GRIPLINK-Plugin für Denso kann sowohl über die Denso-Software „WINCAPS III“ als auch mithilfe eines Datenträgers wie eines USB-Sticks installiert werden.

### 2.2.1 Über WINCAPS III

Kopieren Sie das Verzeichnis *XWrobGriplink* mit allen enthaltenen Unterordnern und Dateien in WINCAPS III in das Roboterprojekt. Dann können Sie das Plugin mit dem Befehl „Daten übertragen“ an die Robotersteuerung übergeben.

### 2.2.2 Per Datenträger

Schließen Sie den Datenträger, auf dem sich das GRIPLINK-Plugin befindet, an die Robotersteuerung an. Öffnen Sie in den Menüpunkt „Daten lesen und schreiben“, indem Sie im Hauptmenü folgende Tasten drücken: F6 -> F6 -> F1. Der eingesteckte Datenträger sollte automatisch ausgewählt sein. Drücken Sie jetzt „Speichern“ und geben einen beliebigen Namen ein, z. B. *GriplinkPluginDenso*. Entfernen Sie den Datenträger von der Robotersteuerung und schließen ihn an einen Computer an. Wenn Sie den Datenträger öffnen, enthält er jetzt einen Ordner namens *RcProject*. Kopieren Sie den Ordner *XWrobGriplink* mit allen Unterverzeichnissen und Dateien unter dem Pfad, den Sie im Bild sehen, auf Ihren Datenträger.



Den Ordner *Griplink* müssen Sie dabei erzeugen (er erleichtert später den Zugriff auf das Plugin), statt *TEST* wird dort der Name stehen, den Sie eingegeben haben. Entfernen Sie den Datenträger vom Computer und schließen ihn wieder an die Steuerung an. Dort gelangen Sie per F6 -> F6 -> F1 wieder in den Menüpunkt „Daten lesen und schreiben“, wählen die zuvor erzeugte Projektdatei aus und drücken auf „Lesen“.

### **2.3 Verhalten im Fehlerfall**

Tritt innerhalb des GRIPLINK-Plugins oder bei der Kommunikation mit dem GRIPLINK Controller ein Fehler auf, so wird das laufende Roboterprogramm grundsätzlich mittels Fehler gestoppt. Dies führt in der Regel dazu, dass laufende Bewegungen des Roboters abgebrochen werden.

Zusätzlich wird die Verbindung zwischen der Robotersteuerung und dem GRIPLINK Controller getrennt. Eine Ausnahme bilden syntaktische Fehleingaben, die vom Compiler nicht direkt erkannt werden (bspw. das Weglassen von Anführungszeichen bei einem String als Argument).



### 3 Befehlsreferenz

Das GRIPLINK-Plugin stellt eine Sammlung an generischen Bewegungs- und Greiffunktionen sowie Funktionen zum Auslesen angeschlossener Sensoren bereit. Die Befehle sind als global sichtbare Unterprogramme definiert, welche ihre Parameter als Übergabewerte erhalten. Zur Ausführung eines Befehls muss das entsprechende Unterprogramm aufgerufen werden.

***Der prinzipielle Programmablauf mit dem GRIPLINK-Plugin ist im Regelfall wie folgt:***

1. Einbinden des Plugins mit #Include (typischerweise vor Beginn der Hauptprozedur)
2. Verbindung herstellen mit *GriplinkConnect*
3. Greifmodul referenzieren und aktivieren mit *GriplinkHome*
4. Greifen bzw. Freigeben mit *GriplinkGrip* bzw. *GriplinkRelease*
5. Verbindung trennen mit *GriplinkDisconnect*

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

### 3.1 Einbinden des Plugins - #Include

Dieser Befehl ist kein Teil des Plugins, trotzdem muss jede Programmdatei, in der GRIPLINK-Befehle verwendet werden, diesen Befehl enthalten. Ansonsten kommt es zu Compiler-Fehlern, da die verwendeten Subroutinen und Funktionen nicht bekannt sind.

#### **Signatur**

```
#Include < path >
```

#### **Parameter**

< path >                    Dateipfad zum Plugin

#### **Beispiel**

Einbinden des Plugins:

```
#Include "\\XWrobGriplink\GriplinkLib\WrobGriplink.pcs"
```



Wenn Sie einen anderen Speicherort gewählt haben, ändert sich auch die Angabe des Pfades. Konsultieren Sie in diesem Fall das Denso Robot User Manual, das zu Ihrer Steuerung passt.

## 3.2 Verbindung aufbauen - Connect

Dieser Befehl stellt die Verbindung zwischen GRIPLINK Controller und der Robotersteuerung entsprechend den Angaben in den Kommunikationseinstellungen her. Der Befehl wartet, bis die Verbindung hergestellt ist oder es zu einem Timeout kommt. Sollten die angeschlossene Hardware oder Software nicht kompatibel sein, wird ein Fehler geworfen.

Wenn GRIPLINK-ET4-Befehle vor *GriplinkConnect* ausgeführt werden, löst dies einen Fehler aus.

### **Signatur**

```
Sub GriplinkConnect(ByVal intChannelIdx as Integer)
```

### **Parameter**

intChannelIdx	Index des gewünschten Client-Channels
---------------	---------------------------------------

### **Rückgabewert**

-

### **Beispiel**

Verbindung zwischen Roboter und dem GRIPLINK Controller per Channel 8 herstellen:

```
GriplinkConnect 8
```

### 3.3 Verbindung trennen - Disconnect

Dieser Befehl beendet die Verbindung zwischen GRIPLINK Controller und der Robotersteuerung, die per *GriplinkConnect* hergestellt wurde.

#### **Signatur**

Sub GriplinkDisconnect

#### **Parameter**

-

#### **Rückgabewert**

-

#### **Beispiel**

Verbindung zwischen Roboter und dem GRIPLINK trennen:

```
GriplinkDisconnect
```

### 3.4 Gerät aktivieren - Enable

Dieser Befehl aktiviert das am angegebenen Port angeschlossene Gerät.

Wird ein Port angesprochen, bevor er aktiviert wurde, kann dies zu einem Fehler führen, der das Programm stoppt.

#### **Signatur**

```
Sub GriplinkEnable(ByVal intPort as Integer)
```

#### **Parameter**

intPort	Index des Geräte-Ports (0 bis 31)
---------	-----------------------------------

#### **Rückgabewert**

-

#### **Beispiel**

Aktiviere Antrieb und Greifteilüberwachung des Greifmoduls an Port 0:

```
GriplinkEnable 0
```

### 3.5 Gerät deaktivieren - Disable

Dieser Befehl deaktiviert das am gewählten Port angeschlossene Gerät. Dieser Befehl kann beispielsweise zum Werkzeugwechsel genutzt werden. Das Gerät kann über Griplink\_Enable wieder aktiviert werden.

#### **Signatur**

```
Sub GriplinkDisable(ByVal intPort as Integer)
```

#### **Parameter**

intPort	Index des Geräte-Ports (0 bis 3)
---------	----------------------------------

#### **Rückgabewert**

-

#### **Beispiel**

Greifmodul an Port 0 wechseln:

```
' Connect to GRIPLINK
GriplinkConnect 8
' Activate gripper 0
GriplinkEnable 0
' DO something
' ...
' Prepare tool change:
' Disable gripper 0:
GriplinkDisable 0
' Now, the gripper can be changed
'
' - OPERATE THE TOOL CHANGER HERE -
'
'Activate the new gripper:
GriplinkEnable 0
```

### 3.6 Greifzustand abfragen – GetState

Dieser Befehl gibt den Gerätezustand des ausgewählten Geräte-Ports als ganzzahligen Wert zurück. Die möglichen Zustände lassen sich Anhang A entnehmen. Um die Abfragen gut lesbar gestalten zu können, wurden Konstanten mit den entsprechenden Zahlenwerten definiert, die man statt der Zahlen verwenden kann. Auch deren Namen lassen sich in Anhang A einsehen.

#### **Signatur**

```
Function GriplinkGetState(ByVal intPort as Integer) as Integer
```

#### **Parameter**

intPort	Index des Geräte-Ports (0 bis 31)
---------	-----------------------------------

#### **Rückgabewert**

Integer	Greifzustand
---------	--------------

#### **Beispiel**

Greife Werkstück und prüfe, ob es korrekt gegriffen wurde:

```
Dim intDevState
' Grip Workpiece
GriplinkGrip 1, 0
' Get Device State
intDevState = GriplinkGetState(1)
' Proceed depending on device state
If intDevState = DS_HOLDING Then
    ...
Else
    ...
End If
```

### 3.7 Greifmodul referenzieren - Home

Dieser Befehl 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 einem Grip/MGrip oder Release/MRelease in eine definierte Position verfahren werden.

Ein Greifer muss einen Home-Befehl erhalten, bevor er fehlerfrei agieren kann.



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

#### **Signatur**

```
Sub GriplinkHome (ByVal intPort as Integer)
```

#### **Parameter**

intPort	Index des Geräte-Ports (0 bis 31)
---------	-----------------------------------

#### **Rückgabewert**

-

#### **Beispiel**

Referenziere das Greifmodul an Port 2:

```
GriplinkHome 2
```



### 3.8 Werkstück greifen - Grip

Durch diesen Befehl greift das ausgewählte Greifmodul ein Werkstück unter Verwendung der ausgewählten Griffkonfiguration. Der Befehl wird ausgeführt, bis der Greifzustand „Holding“ oder „No\_Part“ entspricht. Sollte der Gerätezustand aber „Fault“ entsprechen oder eine Zeitüberschreitung auftreten, wird das laufende Programm gestoppt.



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

#### **Signatur**

```
Sub GriplinkGrip(ByVal intPort as Integer, ByVal intGripPreset as Integer)
```

#### **Parameter**

intPort	Index des Geräte-Ports (0 bis 31)
intGripPreset	Ausgewählter Griff (Wertebereich hängt von Greifermodell ab)

#### **Rückgabewert**

-

#### **Beispiel**

Das Greifmodul an Port 0 soll mit Griffkonfiguration 2 greifen.

```
GriplinkGrip 0, 2
```

### 3.9 Gleichzeitiges Greifen von Werkstücken - MGRIP

Durch diesen Befehl führen die ausgewählten Greifmodule einen Griff aus. Der Befehl wartet, bis alle angesprochenen Greifmodule jeweils einen der Zustände „Holding“ oder „No\_Part“ erreicht haben. Sollte eines der Geräte den Zustand „Fault“ erreichen oder eine Zeitüberschreitung auftreten, wird der weitere Programmablauf gestoppt.



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

#### **Signatur**

```
Sub GriplinkMGrip(ByVal intGripPreset as Integer, ByRef intPortMask() as Integer)
```

#### **Parameter**

intGripPreset	Ausgewählter Griff (Bereich hängt von Greifermodell ab)
intPortMask	Zu verwendende Ports. Array aus Werten 0 (Port nicht verwenden) und 1 (Port verwenden)

#### **Rückgabewert**

-

#### **Beispiel**

Die Greifmodule an den Ports 0 und 1 greifen mit Griffkonfiguration 2:

```
Dim intPorts(3)  
intPorts(0) = 1  
intPorts(1) = 1  
intPorts(2) = 0  
GriplinkMGrip 2, intPorts
```

### 3.10 Werkstück freigeben - Release

Durch diesen Befehl gibt das ausgewählte Greifmodul ein gegriffenes Werkstück wieder frei. Der Befehl wartet, bis das Werkstück freigegeben wurde. Falls der Gerätezustand „Fault“ entspricht, wird der weitere Programmablauf gestoppt.



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

#### **Signatur**

```
Sub GriplinkRelease(ByVal intPort as Integer, ByVal intGripPreset as Integer)
```

#### **Parameter**

intPort	Index des Geräte-Ports (0 bis 31)
intGripPreset	Ausgewählter Griff (Wertebereich hängt von Greifermodell ab)

#### **Rückgabewert**

-

#### **Beispiel**

Gebe das mit Greifmodul an Port 0 gegriffene Werkstück mit der Griffkonfiguration 2 wieder frei:

```
GriplinkRelease 0, 2
```

### 3.11 Gleichzeitiges Freigeben von Werkstücken - MRelease

Durch diesen Befehl geben die ausgewählten Greifmodule ein gegriffenes Werkstück gleichzeitig wieder frei. Der Befehl wartet, bis alle angesprochenen Greifmodule jeweils den Zustand „Released“ erreicht haben. Sollte eines der Geräte den Zustand „Fault“ erreichen oder eine Zeitüberschreitung auftreten, wird der weitere Programmablauf gestoppt.

#### **Signatur**

```
Sub GriplinkMRelease(ByVal intGripPreset as Integer, ByRef intPortMask as Integer)
```

#### **Parameter**

intGripPreset	Ausgewählter Griff (Bereich hängt von Greifermodell ab)
intPortMask	Zu verwendende Ports. Array aus Werten 0 (Port nicht verwenden) und 1 (Port verwenden)

#### **Rückgabewert**

-

#### **Beispiel**

Greifmodule an den Ports 1, 2 und 3 geben ein Werkstück mit Griffkonfiguration 3 frei:

```
Dim intPorts(5)
intPorts(0) = 0
intPorts(1) = 1
intPorts(2) = 1
intPorts(3) = 1
intPorts(4) = 0
GriplinkMRelease 3, intPorts
```

### 3.12 Greifrezepte parametrieren - SetGripCfg

Dieser Befehl parametriert ein Greifrezept direkt aus dem Roboterprogramm. Der GRIPLINK Controller speichert diese Einstellungen nicht dauerhaft, sie gehen beim Ausschalten verloren. Wenn man ein Rezept dauerhaft auf dem Controller speichern möchte, kann man das über die Weboberfläche des Controllers tun. Dieser Befehl ist dazu gedacht, bestimmte Werkzeugeinstellungen mit dem Roboterprogramm zu verknüpfen und damit sicherzustellen, dass die Werkzeugeinstellungen zum Programm passen.



Nutzen Sie die Einstellmöglichkeiten über die Weboberfläche von GRIPLINK, um Greifrezepte vollständig zu parametrieren und dauerhaft zu speichern.

#### Signatur

```
Sub GriplinkSetGripCfg(ByVal intPort as Integer, ByVal intGripPreset as Integer, ByVal strTag as String, ByRef intParams() as Integer)
```

#### Parameter

intPort	Index des Geräte-Ports (0 bis 31)
intGripPreset	Index des zu konfigurierenden Greifrezepts. Die Anzahl der verfügbaren Rezepte hängt vom Gerätetyp ab.
strTag	Name der Griffkonfiguration
intParams(7)	Gerätespezifische Greifparameter Welche Parameter verwendet werden, hängt vom Gerät ab. Sollten einer oder mehrere nicht verwendet werden, sollten diese mit 0 angegeben werden.

Für Greifer von Weiss Robotics gilt beispielsweise:

- Parameter 0: Kein-Teil-Position (in Mikrometern)
- Parameter 1: Freigabe-Position (in Mikrometern)
- Parameter 2: Anteil der maximalen Greifkraft (in tausendstel Prozent)
- Parameter 3 bis 7: nicht verwendet

#### Rückgabewert

-

#### Beispiel

Parametrisiere die Griffkonfiguration 2 des Weiss Robotics IEG 55-020 Greifmoduls an Port 0 wie folgt:

- Kein-Teil-Position: 3 mm
- Freigabe-Position: 10 mm
- Greifkraft: 80%
- Name: GehäuseOben

```
GriplinkSetGripCfg 0, 2, "GehaeuseOben", 3000, 10000, 80000, 0, 0, 0, 0, 0
```

### 3.13 Gerätetypen verifizieren - DevAssert

Dieser Befehl prüft, ob das am angegebenen Geräte-Port angeschlossene Gerät dem erwarteten Gerät entspricht. Ist das nicht der Fall, wird eine Warnung ausgegeben und das Programm gestoppt. Die Überprüfung erfolgt über die IO-Link Vendor und Product ID.

#### **Signatur**

```
Sub GriplinkDevAssert (ByVal intPort as Integer, ByVal intVid as Integer,  
ByVal intPid as Integer)
```

#### **Parameter**

intPort	Index des Geräte-Ports (0 bis 31)
intVid	Erwartete Hersteller Ident-Nummer. Diese wird an den Hersteller von der IO-Link Nutzervereinigung vergeben.
intPid	Erwartete Geräte Ident-Nummer. Vom Hersteller zugewiesene Identifikationsnummer des Geräts.

#### **Rückgabewert**

-

#### **Beispiel**

Stelle sicher, dass an Port 0 ein Greifmodul des Typs IEG 55-020 (Product ID: 20) von Weiss Robotics (Vendor ID: 815) angeschlossen ist:

```
GriplinkDevAssert 0, 815, 20
```

```
Dim intPorts(5)  
intPorts(0) = 0  
intPorts(1) = 1  
intPorts(2) = 0  
intPorts(3) = 1  
intPorts(4) = 1
```

### 3.14 Abfrage von Positions- und Sensorwerten - Value

Dieser Befehl liest einen Messwert vom angeschlossenen Gerät und gibt ihn als ganzzahligen Wert zurück. Abhängig vom Gerät sind ein oder mehrere Messwerte verfügbar, die über den anzugebenden Index ausgewählt werden können.

#### **Signatur**

```
Function GriplinkValue (ByVal intPort as Integer, ByVal intGripPreset as Integer) as Integer
```

#### **Parameter**

intPort	Index des Geräte-Ports (0 bis 31)
intGripPreset	Index des abzufragenden Messwertes

Für Greifer von Weiss Robotics gilt:

- Index = 0: Fingerposition des Greifers (In Mikrometern)
- Index = 1: Temperatur des Greifers (in zehntel Grad Celsius)

#### **Rückgabewert**

Auszulesender Wert (geräteabhängig)

#### **Beispiel**

Speichere die Fingerposition des an Port 0 angeschlossenen Greifers in der Variable *value*:

```
value = GriplinkValue(0, 0)
```

### 3.15 Setzen von gerätespezifischen Steuerwerten – SETVAL

Mit diesem Befehl können gerätespezifische Steuerwerte gesetzt werden. Welche Steuerwerte zur Verfügung stehen ist gerätespezifisch.

#### **Signatur**

```
Sub GriplinkSetVal (ByVal intPort as Integer, ByVal intValueIdx as Integer,  
ByVal intValue as Integer)
```

#### **Parameter**

intPort	Index des Geräte-Ports (0 bis 31)
intValueIdx	Index des zu setzenden Steuerwerts
intValue	Zu setzender Wert

#### **Rückgabewert**

-

#### **Beispiel**

Setze Steuerwert 4 des Geräts an Port 3 auf den Wert 1000:

```
GriplinkSetVal 3, 4, 1000
```



### 3.16 Warten auf gerätespezifischen Steuerwert – WAITVAL

Mit diesem Befehl kann auf die Übernahme eines zuvor mit SETVAL gesetzten Steuerwertes gewartet werden. Der Befehl liefert den übernommenen Steuerwert (oder den aktuellen Wert nach Übernahme des Steuerwertes) zurück.

#### **Signatur**

```
Function GriplinkWaitVal(ByVal intPort as Integer, ByVal intValueIdx as Integer) as Integer
```

#### **Parameter**

intPort	Index des Geräte-Ports (0 bis 31)
intValueIdx	Index des zu setzenden Steuerwertes

#### **Rückgabewert**

Steuerwert nach Übernahme

#### **Beispiel**

Setze Steuerwert 4 des Geräts an Port 3 auf den Wert 1000:

```
Dim intVal as Integer
GriplinkSetVal 3, 4, 1000
intVal = GriplinkWaitVal(3, 4)
If intVal <> 1000 Then
    ...
End If
```

### 3.17 Greifkraftherhaltung steuern - Clamp

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 unmittelbar fortgesetzt werden. Des Weiteren ermöglicht Clamp dauerhaftes Greifen, ohne dass das Greifmodul dabei heiß wird.

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



Clamp ist nicht bei allen Greifmodulen verfügbar.

#### **Signatur**

```
Sub GriplinkClamp(ByVal intPort as Integer, ByVal intEnable as Integer)
```

#### **Parameter**

intPort	Index des Geräte-Ports (0 bis 31)
intEnable	Kraftherhaltung: 1 = ein, 0 = aus

#### **Rückgabewert**

-

#### **Beispiel**

Aktiviert die Greifkraftsicherung beim Greifmodul an Port 2

```
GriplinkClamp 2,1
```

### 3.18 Ansteuerung der LED-Anzeige - LED

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

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



Leuchtmuster können über die Weboberfläche des GRIPLINK Controllers konfiguriert werden.



Eine steuerbare LED-Anzeige ist nicht bei allen Greifmodulen verfügbar.

#### **Signatur**

```
Sub GriplinkLED(ByVal intPort as Integer, ByVal intLEDPreset as Integer)
```

#### **Parameter**

intPort	Index des Geräte-Ports (0 bis 31)
intLEDPreset	Index des vordefinierten Leuchtmusters (0 bis 7)

#### **Rückgabewert**

-

#### **Beispiel**

Aktiviert das Leuchtmuster 3 am an Port 2 angeschlossenen Greifer:

```
GriplinkLED 2, 3
```

## Anhang A Greifzustand

Greifzustand	Code	Konstante
NOT CONNECTED	0	GS_Not_Connected
NOT INITIALIZED	1	GS_Not_Inizialized
IDLE	2	GS_Idle
RELEASED	3	GS_Released
NO PART	4	GS_No_Part
HOLDING	5	GS_Holding
PART LOST	6	GS_Part_Lost
FAULT	7	GS_Fault

© 2022 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.