



GRIPLINK-PLUGIN FÜR EPSON

Version 4.0.0

April 2023



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
3	Befehlsreferenz.....	6
3.1	Verbindung aufbauen - CONNECT.....	7
3.2	Verbindung beenden – DISCONNECT.....	8
3.3	Gerät aktivieren - ENABLE.....	9
3.4	Gerät deaktivieren - DISABLE.....	10
3.5	Gerätezustand abfragen – DEVSTATE.....	11
3.6	Greifmodul referenzieren - HOME.....	12
3.7	Werkstück greifen - GRIP.....	13
3.8	Gleichzeitiges Greifen von Werkstücken – MGRIP.....	14
3.9	Werkstück freigeben - RELEASE.....	15
3.10	Gleichzeitiges Freigeben von Werkstücken – MRELEASE.....	16
3.11	Sensorwert auslesen - VALUE.....	17
3.12	Greifkraftherhaltung steuern - HOLD.....	19
3.13	Ansteuerung der LED-Anzeige – GRIPLINK LED.....	21
3.14	Greifeinstellungen setzen - GRIPCFG.....	22
3.15	Gerätetyp sicherstellen - DEVASSERT.....	23
Anhang A		
4	Fehlerbehandlung.....	24
	Gerätezustand.....	26

1 Einführung

Mit der GRIPLINK-Technologie können servoelektrische und smarte pneumatische Greifmodule von WEISS ROBOTICS sowie ausgewählte Sensoren und Aktoren von Fremdherstellern 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

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 sind die folgenden EPSON Produkte erforderlich:

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



Kontaktieren Sie EPSON oder ihren EPSON Partner zum Bezug dieser Produkte.



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 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 GRIPLINK verwendet werden soll.
3. Importieren Sie die entpackten Dateien in EPSON RC+ mittels des Menüpunktes „Datei“ -> „Importieren“ in Ihr neues Projekt. Beachten Sie dabei, dass das Paket mehrere 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 Befehlsschnittstelle 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.

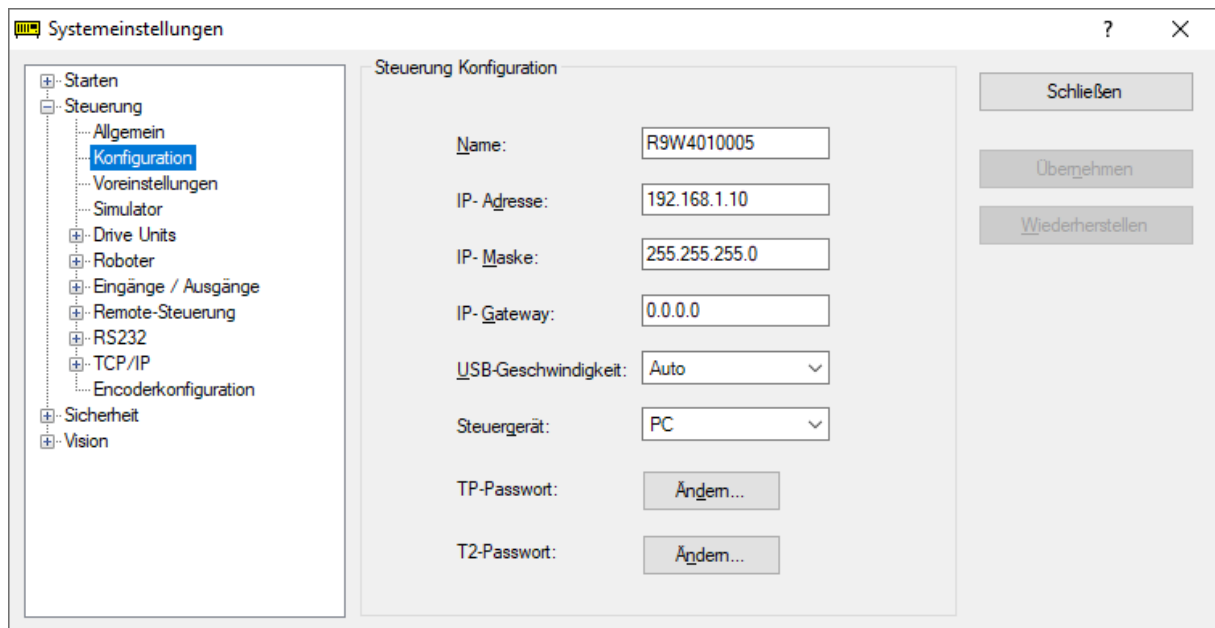


Abbildung 1: Einstellen der IP-Adresse auf der Robotersteuerung

3 Befehlsreferenz

Das GRIPLINK-Plugin stellt dem Anwender eine Sammlung an Funktionen zur Steuerung von Greifmodulen bzw. Aktoren sowie zum Auslesen von Sensormesswerten zur Verfügung. Es stehen sowohl Einzel- wie auch als Mehrfachbefehle zur Verfügung.

Mehrfachbefehle

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

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

1. Verbindung herstellen mit CONNECT
2. Gerät aktivieren mit ENABLE
3. Bei Servogreifmodulen ohne Absolutgeber: Greifmodul referenzieren mit HOME
4. Greifen/Freigeben mit GRIP/MULTIGRIP bzw. RELEASE/MULTIRELEASE

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

Befehl	Beschreibung
Griplink_Connect()	Verbindung zum GRIPLINK aufbauen
Griplink_Disconnect()	Verbindung zum GRIPLINK trennen
Griplink_Enable()	Gerät aktivieren
Griplink_Disable()	Gerät deaktivieren
Griplink_DevState()	Gerätezustand abfragen
Griplink_MultiDevState()	Gerätezustand aller Geräte abfragen
Griplink_Value()	Sensorwert(e) abfragen
Griplink_Home()	Greifmodul referenzieren
Griplink_Grip()	Werkstück greifen
Griplink_MultiGrip()	Gleichzeitiges Greifen mehrerer Werkstücke
Griplink_Release()	Werkstück freigeben
Griplink_MultiRelease()	Gleichzeitiges Freigeben von mehreren Werkstücken
Griplink_Hold()	Greifkrafterhaltung PERMAGRIP® 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.



Zu beachten ist, dass der gewählte Port 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)
```

Parameter

strIPAddr\$	IP-Adresse des Griplink als String
intControllerPortNumber	Port-Nummer der vorkonfigurierten TCP/IP-Verbindung. Muss zwischen 201 und 216 liegen.

Rückgabewert

-

Beispiel

Verbindung zwischen Robotersteuerung und GRIPLINK mit der IP-Adresse 192.168.1.40 herstellen unter Verwendung von TCP/IP-Port Nr. 201.

```
Griplink_Connect("192.168.1.40", 201)
```


3.2 Verbindung beenden – DISCONNECT

Dieser Befehl beendet eine bestehende Verbindung, die mittels Griplink_Connect() aufgebaut wurde.

Signatur

```
Function Griplink_Disconnect()
```

Parameter

-

Rückgabewert

-

Beispiel

```
Griplink_Disconnect()
```

3.3 Gerät aktivieren - ENABLE

Dieser Befehl aktiviert das Gerät am ausgewählten GRIPLINK-Port. Der Befehl löst einen Zustandsübergang nach *DS_ENABLED* aus und aktiviert ggf. das angeschlossene Gerät. Er dient dazu, nach dem Quittieren eines Fehlers mit DISABLE das entsprechende Gerät wieder zu aktivieren.



Für Greifmodule von WEISS ROBOTICS ist zu beachten, dass der Befehl beim Starten des Roboterprogramms nicht mehr zwingend notwendig ist und in jedem Fall erst nach dem Ausführen der Referenzierungsfahrt verwendet werden kann.

Signatur

```
Function Griplink_Enable(ubDevicePort As UByte)
```

Parameter

ubDevicePort	Port-Index (0 bis 31)
--------------	-----------------------

Rückgabewert

-

Beispiel

Aktiviere das Gerät Port 0:

```
Griplink_Enable(0)
```

3.4 Gerät deaktivieren - DISABLE

Dieser Befehl deaktiviert das Gerät am ausgewählten GRIPLINK-Port. Im Fall eines angeschlossenen Greifmoduls kann diese Funktion z. B. genutzt werden, um den Antrieb des Greifmoduls stromlos zu schalten.

Signatur

```
Function Griplink_Disable(ubDevicePort As UByte)
```

Parameter

ubDevicePort	Port-Index (0 bis 31)
--------------	-----------------------

Rückgabewert

-

Beispiel

Greifmodul an Port 1 wechseln:

```
'Verbindung zwischen Roboter und dem GRIPLINK herstellen  
Griplink_Connect("192.168.1.40", 201)  
'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)  
'Programm mit neuem Werkzeug fortsetzen  
'...
```

3.5 Gerätezustand abfragen – DEVSTATE

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

Signatur

```
Function Griplink_DevState(ubDevicePort As UByte)
```

Parameter

ubDevicePort Port-Index (0 bis 31)

Rückgabewert

Aktueller Gerätezustand (vgl. Anhang A)

Beispiel

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

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

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 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 Richtung der Referenzfahrt kann über die Weboberfläche des GRIPLINK Schnittstellenwandlers konfiguriert werden.

Signatur

```
Function Griplink_Home(ubDevicePort As UByte)
```

Parameter

ubDevicePort Port-Index (0 bis 31)

Rückgabewert

-

Beispiel

Referenziere Greifmodul an Port 3:

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

3.7 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(ubDevicePort As UByte, ubGripIndex As UByte) As Integer
```

Parameter

ubDevicePort	Port-Index (0 bis 31)
ubGripIndex	Ausgewählter Griff (Wertebereich abhängig vom Greifmodul)

Rückgabewert

Aktueller Greifzustand (siehe Kapitel 3.16)

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 DS_NO_PART und DS_HOLDING sind in Griplink.inc
'definiert
If intGripState = DS_NO_PART Then
    Print "No part found"
ElseIf intGripState <> DS_HOLDING Then
    Print "Unexpected grip state"
EndIf
```

3.8 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(ByRef boolGrippers() As Boolean, ubGripIndex  
As UByte)
```

Parameter

boolGrippers()	Array von Boolean-Werten zur Auswahl der einzelnen Greifmodule, mit denen gegriffen werden soll. Ein Wert von <i>True</i> bedeutet, dass mit dem jeweiligen Greifmodul gegriffen wird. Die Position im Array entspricht der Port-Nummer. Die Größe des Arrays kann variabel für bis zu 32 Greifmodule ausgelegt werden. Bei kleineren Arrays werden fehlende Ports automatisch auf <i>False</i> gesetzt.
ubGripIndex	Ausgewählter Griff (Wertebereich abhängig vom Greifmodul)

Rückgabewert

-

Beispiele

Greifmodul an Port 0 und 3 greifen Werkstück mit Griff 2 und prüfen, ob ein Teil gehalten wird:

```
#include Griplink.inc  
Boolean boolSelectedGrippers(4)  
boolSelectedGrippers(0) = True  
boolSelectedGrippers(3) = True  
Griplink_MultiGrip(ByRef boolSelectedGrippers(), 2)  
'Die Konstanten DS_NO_PART und DS_HOLDING sind in Griplink.inc  
'definiert  
If Griplink_DevState(0) = DS_HOLDING And Griplink_DevState(3) =  
DS_HOLDING Then  
    Print "Holding parts"  
EndIf
```

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.

Signatur

```
Function Griplink_Release(ubDevicePort As UByte, ubGripIndex As UByte)
```

Parameter

ubDevicePort	Port-Index (0 bis 31)
ubGripIndex	Ausgewählter Griff (Wertebereich abhängig vom Greifmodul)

Rückgabewert

-

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 DS_RELEASED ist in Griplink.inc
'definiert
If intGripState = DS_RELEASED Then
    Print "Part released"
EndIf
```


3.10 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.



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

Signatur

```
Function Griplink_MultiRelease(ByRef boolGrippers() As Boolean,  
ubGripIndex As UByte)
```

Parameter

boolGrippers()	Array von Boolean-Werten zur Auswahl der einzelnen Greifmodule. Ein Wert von <i>True</i> bedeutet, dass das Werkstück im jeweiligen Greifmodul freigegeben wird. Die Position im Array entspricht der Port-Nummer. Die Größe des Arrays kann variabel für bis zu 32 Greifmodule ausgelegt werden. Bei kleineren Arrays werden fehlende Ports automatisch auf <i>False</i> gesetzt.
ubGripIndex	Ausgewählter Griff (Wertebereich abhängig vom Greifmodul)

Rückgabewert

-

Beispiele

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

```
#include Griplink.inc  
Boolean boolSelectedGrippers(8)  
boolSelectedGrippers(1) = True  
boolSelectedGrippers(2) = True  
boolSelectedGrippers(3) = True  
Griplink_MultiRelease(ByRef boolSelectedGrippers(), 3)
```

3.11 Sensorwert auslesen - VALUE

Dieser Befehl gibt den aktuellen Messwert eines angeschlossenen Sensors zurück. Im Fall eines angeschlossenen Greifmoduls von WEISS ROBOTICS kann z. B. die aktuelle Fingerposition in Mikrometern ausgelesen werden. Im Fall von mehreren zur Verfügung stehenden Messwerten kann der gewünschte Messwert über den Parameter *ubIndex* selektiert werden.

Für Greifmodule von Weiss Robotics liefert Index 0 immer die Fingerposition in Mikrometern (μm) zurück.

Signatur

```
Function Griplink_Value(ubDevicePort As UByte, ubIndex As Ubyte) As Long
```

Parameter

ubDevicePort	Port-Index (0 bis 31)
ubIndex	Messwert-Index (Wertebereich geräteabhängig)

Rückgabewert

Sensorwert (z. B. Fingerposition in μm)

Beispiel

Lese die Fingerposition von Greifer 0 aus:

```
Long lngPos
lngPos = Griplink_Value(0, 0)
If lngPos > 80000 Then
    Print "Finger position is 80 mm"
EndIf
```

3.12 Steuerwert setzen – SETVAL

Dieser Befehl setzt einen Wert zur Steuerung des Greifmoduls. Der Befehl ist nicht-blockierend, d.h. es wird nicht auf die Übernahme des Wertes gewartet. Um die Übernahme sicher zu stellen, kann der Befehl WAITVAL verwendet werden. Der Befehl ist funktionsabhängig und nicht auf allen Geräten verfügbar.

Signatur

```
Function Griplink_SetVal (ubDevicePort As UByte, ubIndex As UByte,  
    lngValue As Long)
```

Parameter

ubDevicePort	Port-Index (0 bis 31)
ubIndex	Steuerwert-Index (Wertebereich geräteabhängig)
lngValue	Steuerwert

Rückgabewert

-

Beispiel

Setze die Fingerposition zur Vorpositionierung der Greiferbacken an einem WPG-Greifmodul auf 100 mm. Nach dem Absenden verfahren die Greiferbacken auf eine Öffnungsweite von 100 mm. Der Befehl kommt sofort zurück, d. h. die Zielposition ist nach der Ausführung des Befehls noch nicht sicher erreicht. Um dies sicher zu stellen, muss mit WAITVAL ein weiterer Befehl aufgerufen werden:

```
Griplink_SetVal(0, 0, 100000)  
Griplink_WaitVal(0, 0)
```

3.13 Auf Übernahme von Steuerwert warten - WAITVAL

Dieser Befehl dient dazu, nach dem Setzen eines Steuerwertes mit SETVAL auf die Übernahme dieses Wertes zu warten. Der Befehl ist funktionsabhängig und nicht für alle Geräte verfügbar.

Signatur

```
Griplink_WaitVal(ubDevicePort As UByte, ubIndex As UByte) As Long
```

Parameter

ubDevicePort	Port-Index (0 bis 31)
ubIndex	Steuerwert-Index (Wertebereich geräteabhängig)

Rückgabewert

Übernommener/aktueller Steuerwert (z. B. Fingerposition in μm)

Beispiel

Setze die Fingerposition zur Vorpositionierung der Greiferbacken an einem WPG-Greifmodul auf 63 mm. Nach dem Absenden verfahren die Greiferbacken auf eine Öffnungsweite von 63 mm. Der Befehl kommt sofort zurück, d. h. die Zielposition ist nach der Ausführung des Befehls noch nicht sicher erreicht. Um dies sicher zu stellen, muss mit WAITVAL ein weiterer Befehl aufgerufen werden:

```
Griplink_SetVal(0, 0, 63000)  
Griplink_WaitVal(0, 0)
```

3.14 Greifkraftherhaltung steuern - HOLD

Die von WEISS ROBOTICS entwickelte innovative Greifkraftsicherung PERMAGRIP® erhält die Greifkraft am Werkstück, auch wenn die Stromzufuhr zum Greifmodul unerwartet unterbrochen wird. Des Weiteren ermöglicht PERMAGRIP® dauerhaftes Greifen, ohne dass das Greifmodul dabei überhitzen kann.

Dieser Befehl aktiviert oder deaktiviert die Greifkraftherhaltung bei dem ausgewählten Greifmodul. Bei deaktiviertem Greifmodul (siehe *Griplink_Enable()*, Kapitel 3.2 und *Griplink_Disable()*, Kapitel 3.4) kann mittels dieser Funktion die Bewegung der Greiffinger von Hand erlaubt oder verhindert werden.



Für die Sicherung der Greifkraft im Falle einer Unterbrechung der Stromzufuhr ist diese Funktion unerheblich. Die Greifkraftsicherung bei Stromausfall ist immer aktiv.



PERMAGRIP ist nicht bei allen Greifmodulen verfügbar.

Signatur

```
Function Griplink_Hold(ubDevicePort As UByte, boolEnable As Boolean)
```

Parameter

ubDevicePort	Port-Index (0 bis 31)
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 DS_HOLDING ist in Griplink.inc
'definiert
If intGripState = DS_HOLDING Then
    'PERMAGRIP® aktivieren
    Griplink_Hold(2, True)
EndIf
```

3.15 Ansteuerung der LED-Anzeige – GRIPLINK LED

Dieser Befehl ändert die Farbe und das Muster des Leuchtrings eines selektierten Greifmoduls der CRG-Serie von WEISS ROBOTICS.



Leuchtmuster können über die Weboberfläche des GRIPLINK Schnittstellenwandlers konfiguriert werden.



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

Signatur

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

Parameter

ubDevicePort	Port-Index (0 bis 31)
ubLEDIndex	Index des vordefinierten Leuchtmusters (0 bis 7)

Rückgabewerte

-

Beispiel

Greife mit dem Greifmodul an Port 3 und ändere die Farbe des Leuchtrings auf das Leuchtmuster 0, wenn die Fingerposition danach größer gleich 80 mm und auf Leuchtmusters 1, wenn kleiner:

```
#include "Griplink.inc"
Integer intGripState
Long lngPos
intGripState = Griplink_Grip(3, 0)
If intGripState = DS_HOLDING Then
    lngPos = Griplink_Value(3, 0)
    If lngPos > 80000 Then
        'Setze LED-Muster 0
        Griplink_LED(0)
    Else
        'Setze LED-Muster 1
        Griplink_LED(1)
    EndIf
Else
    Print "Failed to grip part"
EndIf
```

3.16 Greifeinstellungen setzen - GRIPCFG

Setzt die Greifeinstellungen für den gewählten Griff auf dem gewählten Gerät. Jeder Griff besteht aus einem Namen (Zeichenkette) und drei Parametern. Der Name dient der einfacheren Identifikation des Griffs.



Für die Parameter können, je nach Gerät, bestimmte Wertebereiche gelten. Diese entnehmen Sie bitte der Dokumentation des Geräts (z.B. Greifmodul).

Signatur

```
Function Griplink_GripCfg(ubDevicePort As UByte, ubGripIndex As UByte,  
label$ As String, ByRef lngParams() As Long)
```

Parameter

ubDevicePort	Port-Index (0 bis 31)
ubGripIndex	Ausgewählter Griff (Wertebereich abhängig vom Greifmodul)
label\$	Label / Name des Griffs
lngParams	Array aus 8 Konfigurationswerten



Für servoelektrische Greifmodule von Weiss Robotics bilden die ersten drei Parameter die Griffkonfiguration ab. Wert 0 repräsentiert den Grenzwert kein Teil (No Part Limit) in Mikrometern (μm), Wert 1 repräsentiert den Grenzwert Teil freigegeben (Release Limit) ebenfalls in Mikrometern (μm). Wert 2 entspricht der Greifkraft in Prozent der Nominalgreifkraft, multipliziert mit dem Faktor 1000, d.h. z.B. 100000 für Greifkraft 100%.

Rückgabewerte

-

Beispiel

Setze die Konfiguration für Griff 2 auf dem Greifmodul an Port 1:

```
#include "Griplink.inc"  
Long lngParams(8)  
lngParams(0) = 2000 // No Part Limit 2.0 mm  
lngParams(1) = 10000 // Release Limit 10.0 mm  
lngParams(2) = 80000 // Gripping force 80%  
lngParams(3) = 0  
...  
lngParams(7) = 0  
Griplink_GripCfg(0, 1, "Workpiece 1", ByRef lngParams())
```

3.17 Gerätetyp sicherstellen - DEVASSERT

Stellt sicher, dass an dem selektierten Port das gewünschte Gerät verbunden ist.



Die notwendigen Parameter IO-Link Vendor ID und Device ID entnehmen Sie bitte der Beschreibung des Geräts.

Signatur

```
Function Griplink_DevAssert(ubDevicePort As UByte, intVendorID As Integer, intDeviceID As Integer)
```

Parameter

ubDevicePort	Port-Index (0 bis 31)
intVendorID	IO-Link Vendor ID des gewünschten Geräts
intDeviceID	IO-Link Device ID des gewünschten Geräts

Rückgabewerte

-

Beispiel

Prüfe, ob an Port 2 ein Greifmodul IEG55-020 (Device ID 20) von Weiss Robotics (Vendor ID 815) verbunden ist:

```
#include "Griplink.inc"  
Griplink_DevAssert(2, 815, 20)
```


4 Fehlerbehandlung

Tritt innerhalb des GRIPLINK-Plugins ein Problem auf, so wird auf der Robotersteuerung ein Fehler ausgelöst. Sofern dieser Fehler nicht vom Benutzer abgefangen wird, führt dies zu einem Stopp des Roboterprogramms. Dabei werden laufende Roboterbewegungen noch zu Ende geführt, bevor der Roboter anhält.



Das Abfangen dieser Fehler und das anschließende Überführen der Anlage in einen sicheren Zustand obliegt dem Programmierer. Die EPSON-Programmierungsumgebung stellt hierfür geeignete Befehle wie z. B. *OnErr* bereit (vgl. Dokumentation der EPSON-Robotersteuerung).

Um verschiedene Fehlerarten unterscheiden zu können, verwendet das GRIPLINK-Plugin mehrere unterschiedliche Fehlernummern aus dem Wertebereich der benutzerdefinierten Fehlernummern der Robotersteuerung. Tabelle 1 zeigt die vordefinierten Fehlernummern und ihre Bedeutung. Um Konflikte mit bestehenden Roboterprogrammen zu umgehen, können diese Fehlernummern in der Datei *GriplinkConfig.inc* angepasst werden.

Fehler	Fehlernummer	Bedeutung
GRIPLINK_CONNECTION_ERROR	8100	Verbindungsfehler zwischen GRIPLINK-Schnittstellenwandler und Greifmodul
GRIPLINK_COMMAND_ERROR	8101	Fehler beim Ausführen eines Befehls
GRIPLINK_DEVICE_ERROR	8102	Gerätefehler auf angeschlossenem Greifmodul (z. B. Übertemperatur)
GRIPLINK_PARAM_ERROR	8103	Parameter-Fehler beim Aufruf einer Griplink-Funktion (z. B. Wert außerhalb des zulässigen Bereichs)
GRIPLINK_TIMEOUT_ERROR	8104	Überschreitung der zulässigen Ausführungszeit beim Ausführen eines Griplink-Befehls

Tabelle 1: Vordefinierte Fehlernummern

Fehler, die bei der Ausführung der in Kapitel 3 aufgeführten GRIPLINK-Funktionen ausgelöst werden, können über die übliche Fehlerbehandlung von SPEL+ abgefangen werden. Beispiel:

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

EndIf

errorHandler:

MsgBox "Failed to execute grip command"



Weitere Informationen zur Fehlerbehandlung in SPEL+ entnehmen Sie bitte der Dokumentation ihres EPSON Robotersystems.

Gerätezustand

Gerätezustand	Konstante	Code	Beschreibung
NOT CONNECTED	DS_NOT_CONNECTED	0	Port nicht verbunden
NOT INITIALIZED	DS_NOT_INITIALIZED	1	Gerät nicht initialisiert
DISABLED	DS_DISABLED	2	Betriebsbereit, nicht aktiv
RELEASED	DS_RELEASED	3	Werkstück freigegeben
NO PART	DS_NO_PART	4	Kein Werkstück gefunden
HOLDING	DS_HOLDING	5	Werkstück wird gehalten
OPERATING	DS_OPERATING	6	Betriebsbereit, aktiv
FAULT	DS_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.