



GRILINK PLUGIN FOR KUKA

Version 1.1.0



Content

1	Introduction	2
1.1	Notation and symbols.....	2
1.2	Intended use	2
1.3	System requirements.....	2
1.4	License terms	3
2	Installation	4
2.1	Preparation of the robot.....	4
2.2	Configuration of the XML file.....	5
2.3	Network configuration settings	5
2.4	Software Installation.....	6
2.5	Behavior in case of error.....	7
3	Command Reference	7
3.1	Establish connection - CONNECT	8
3.2	Close connection - DISCONNECT	8
3.3	Activate device - ENABLE	9
3.4	Deactivate device - DISABLE	9
3.5	Referencing gripping module - HOME	10
3.6	Grip workpiece - GRIP	11
3.7	Simultaneous gripping of workpieces - MGRIP.....	12
3.8	Release workpiece - RELEASE	13
3.9	Simultaneous release of workpieces - MRELEASE	14
3.10	Configure grip preset - GRIPCFG	15
3.11	Verify device types - DEVASSERT	16
3.12	Control gripping force retention - CLAMP	17
3.13	Control of the LED display - LED.....	18
3.14	Query device state - GETSTATE.....	19
3.15	Query of position and sensor values - VALUE.....	20
3.16	Set value - SETVAL.....	21
3.17	Wait for a value to reach the set value - WAITVAL.....	22
Anhang A	Device states	23

1 Introduction

With the GRIPLINK technology, servo-electric and smart pneumatic gripping modules from WEISS ROBOTICS as well as sensors/actors from selected third-party vendors can be connected to robot systems from leading manufacturers via a network connection. The GRIPLINK plug-in for KUKA is the software link between the GRIPLINK-ET4 interface converter and the robot controller and enables the easy integration of WEISS ROBOTICS' GRIPLINK technology into KUKA robot systems.



This manual describes the functions of the GRIPLINK plug-in. For information about installation and operation of the GRIPLINK-ET4 interface converter please refer to the GRIPLINK-ET4 user's manual. The manual can be found online on www.griplink.de/manuals

1.1 Notation and symbols

For a better understanding, the following symbols are used in this manual:



Functional or safety relevant information. Non-compliance may endanger the safety of personnel and the system, damage the device or impair its function.



Additional information for a better understanding of the described facts.



Reference to further information.

1.2 Intended use

The software "GRIPLINK plug-in" is intended for communication between the GRIPLINK-ET4 interface converter from WEISS ROBOTICS and a robot controller. The requirements of the applicable guidelines as well as the installation and operation instructions in this manual must be noted and adhered to. Any other use or use beyond that is considered improper use. The manufacturer is not liable for any damage resulting from this.

1.3 System requirements

One of the following KUKA robot controllers is required to run this software:

- KRC 4 or 5

The following robot option is required to run the software:

- KUKA Ethernet KRL



Please contact KUKA or your KUKA distributor in order to purchase these products.



The IP addresses of both the robot controller and the GRIPLINK-ET4 interface converter must be located within the same subnet range. Please contact your network administrator if you experience any problems assigning appropriate IP addresses to the devices.

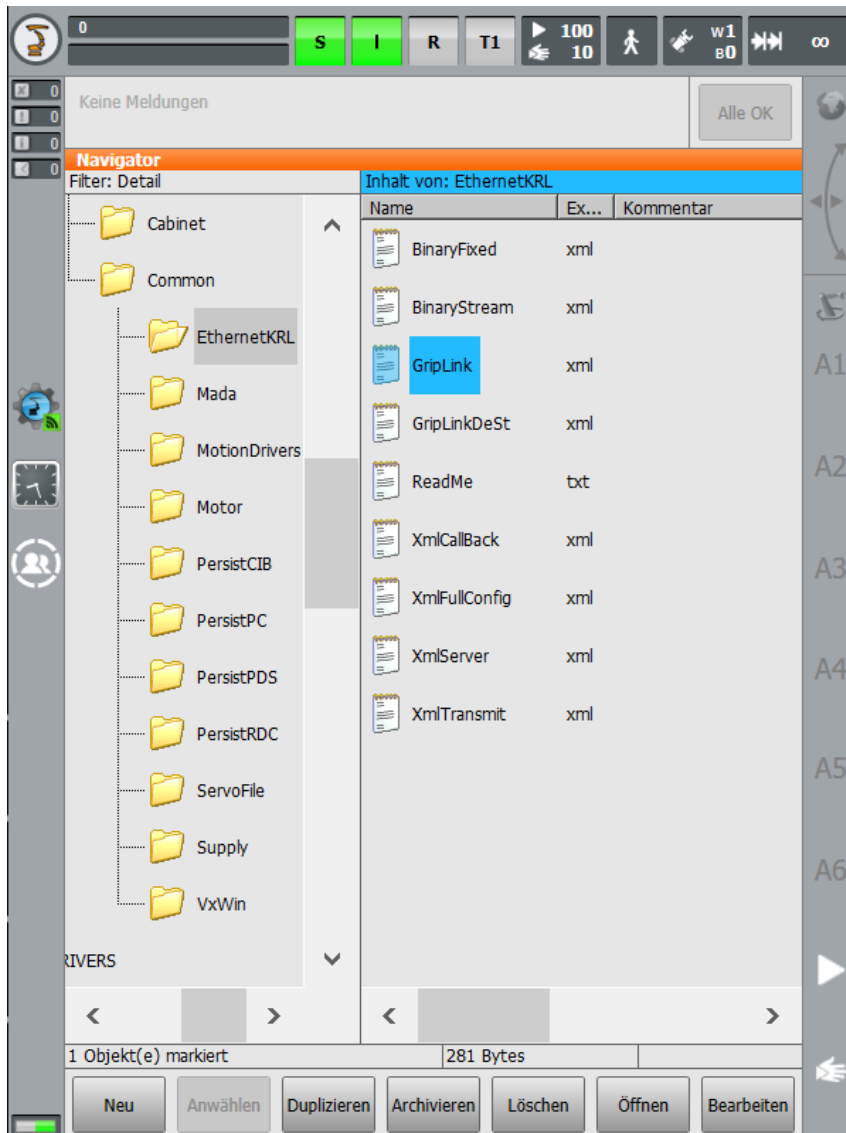
1.4 License terms

The GRIPLINK plug-in is protected by copyright. The software package includes the applicable license terms. By installing and using the GRIPLINK plug-in, the user accepts these license terms.

2 Installation

2.1 Preparation of the robot

Copy the XML file *GripLink.xml* into the folder `C:\KRC\ROBOTER\Config\User\Common\EthernetKRL` of your robot controller. This file path is only visible if you are acting as an expert (level 3) or higher.



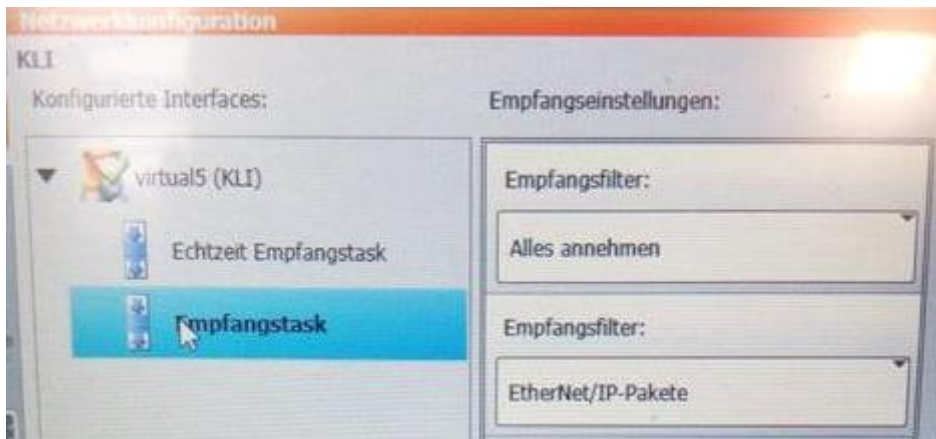
2.2 Configuration of the XML file

In the XML file, enter the IP address of the GRIPLINK (192.168.1.40 by default) in the <IP> field, and the port number 10001 accordingly in the <PORT> field. If you have changed the IP address of the GRIPLINK Controller, adjust the IP address in the XML file accordingly.

```
<ETHERNETKRL>
  <CONFIGURATION>
    <EXTERNAL>
      <IP>192.168.1.40</IP>
      <PORT>10001</PORT>
    </EXTERNAL>
  </CONFIGURATION>
  <RECEIVE>
    <RAW>
      <ELEMENT Tag="GRIPLINK_RESPONSE" Type="STREAM"
        Size="128" EOS="10"/>
    </RAW>
  </RECEIVE>
</SEND/>
</ETHERNETKRL>
```

2.3 Network configuration settings

Configure the receive task within the network configuration to have the following filter settings:



2.4 Software Installation



Make sure that you are using the latest version of the GRIPLINK plugin. The latest version can be downloaded from www.griplink.de/software.

The GRIPLINK plug-in for KUKA can be installed via the KUKA WorkVisual software or by using a USB thumb drive to copy the files directly to the robot controller.

2.4.1 Installation via WorkVisual

Copy the *GripLink_Kuka.dat* and *GripLink_Kuka.src* files in WorkVisual to the program folder. Then copy files to your robot using the "Transfer changes" option.

2.4.2 Direct copy to robot controller

Copy the contents of the ZIP archive to a USB thumb drive. Then connect the USB thumb drive to the robot controller or to the smartPAD and copy the *GripLink_Kuka.src* and *GripLink_Kuka.dat* files to the folder *KRC:\R1\Program* of your robot controller.

If your filter setting is not "File" but "Module", you can copy the *GripLink_Kuka* module accordingly.

2.5 Behavior in case of error

If an error occurs within the GRIPLINK plug-in or during communication with the GRIPLINK controller, the running robot program will be stopped by means of an error. This usually interrupts and aborts any running motion of the robot.

In addition, the connection between the robot controller and the GRIPLINK Controller will be closed and the user will be prompted to deselect the running program.

3 Command Reference

The GRIPLINK plugin provides a collection of generic motion and grip functions as well as functions for reading connected sensors. The commands are defined as globally visible subprograms, which receive their parameters as transfer values. To execute a command, the corresponding subroutine must be called.

The basic program flow with the GRIPLINK plugin is always as follows:

1. Establish connection with GRIPLINK_CONNECT
2. Reference and activate gripping module with GRIPLINK_HOME
3. Grip or release with GRIPLINK_GRIP or GRIPLINK_RELEASE
4. Disconnect with GRIPLINK_DISCONNECT

The commands available with the GRIPLINK plug-in are described below.

3.1 Establish connection - CONNECT

This command establishes the connection between the GRIPLINK controller and the robot controller according to the specifications in the file *GripLink.xml*. The command waits until the connection is established or a timeout occurs. If the connected hardware or software is not compatible, an error will be thrown.

If GRIPLINK-ET4 commands are executed before GRIPLINK_CONNECT, this will raise an error.

Syntax

```
GRIPLINK_CONNECT()
```

Parameters

none

Return value

none

Example

Establish connection between robot and GRIPLINK:

```
GRIPLINK_CONNECT ()
```

3.2 Close connection - DISCONNECT

This command closes the active connection between the GRIPLINK controller and the robot controller according to the specifications in the *GripLink.xml*.

Syntax

```
GRIPLINK_DISCONNECT()
```

Parameter

none

Return value

none

Example

Disconnect the connection between the robot and the GRIPLINK:

```
GRIPLINK_DISCONNECT ()
```

3.3 Activate device - ENABLE

This command activates the device connected to the specified port.

If a port is addressed before it has been activated, an error may be raised and the program may be stopped.

Syntax

```
GRIPLINK_ENABLE(<PORT>)
```

Parameters

<PORT> Index of the device port (0 to 31)

Return value

none

Example

Activate drive and gripping part monitoring of the gripping module at port 0:

```
GRIPLINK_ENABLE (0)
```

3.4 Deactivate device - DISABLE

This command deactivates the device connected to the selected port. This command can be used, for example, to change tools. The device can be reactivated via GRIPLINK_ENABLE.

Syntax

```
GRIPLINK_DISABLE(<PORT>)
```

Parameters

<PORT> Index of the device port (0 to 31)

Return value

none

Example

Change the gripping module at port 0:

```
; Connect to GRIPLINK  
GRIPLINK_CONNECT()  
Activate gripper 0  
GRIPLINK_ENABLE (0)  
; DO something  
; ...  
; Prepare tool change:  
; Disable gripper 0:  
GRIPLINK_DISABLE (0)  
; Now, the gripper can be changed
```

```
;
; - OPERATE THE TOOL CHANGER HERE -
;
; Activate the new gripper:
GRIPLINK_ENABLE (0)
```

3.5 Referencing gripping module - HOME

This command references the selected servo gripper. The command executes a reference run of the gripping module and waits until this is completed. After the Home command has been executed, the fingers of the gripping module are powerless and must be moved to a defined position with a Grip/MGrip or Release/MRelease command.

A gripper has to receive a Home command before it can act without errors.



Not every gripping module requires a referencing cycle. Check gripping module manual!

Syntax

```
GRIPLINK_HOME(<PORT>)
```

Parameters

<PORT> Index of the device port (0 to 31)

Return value

none

Example

Reference the gripping module to port 2:

```
GRIPLINK_HOME (2)
```

3.6 Grip workpiece - GRIP

This command causes the selected gripping module to grip a workpiece using the selected grip configuration. The command is executed until the gripping state corresponds to HOLDING or to NO PART. However, if the device state corresponds to FAULT or a timeout occurs, the running program is stopped.



The gripping parameters can be configured via the web interface of the GRIPLINK controller.

Syntax

```
GRIPLINK_GRIP(<PORT>, <INDEX>, <WSTR>)
```

Parameters

<PORT>	Index of the device port (0 to 31)
<INDEX>	Selected handle (range depends on gripper model)
<WSTR>	Whether or not to wait for state transitions using WSTR

Return value

none

Example

Grip with device at port 0 and grip preset 2. Then, wait for state transitions.

```
GRIPLINK_GRIP(0, 2, TRUE)
```

3.7 Simultaneous gripping of workpieces - MGRIP

This command causes the selected gripping modules to execute a grip. The command waits until all gripping modules addressed have each reached one of the states HOLDING or NO PART. If one of the devices reaches the FAULT state or a timeout occurs, the program will be stopped.



The gripping parameters can be configured via the web interface of the GRIPLINK controller.

Syntax

```
GRIPLINK_MGRIP(<INDEX>, <PORTS>, <MWAITFOR>)
```

Parameters

<INDEX>	Grip preset index
<PORTS>	Selected ports as stringified binary array (e.g. "1,0,0,1" for ports 3 and 0)
<MWAITFOR>	Whether or not to wait for state transitions using MWAITFOR

Return value

none

Example

Perform a synchronous grip with the devices at ports 2 and 3 grip using grip preset 2. Then, wait for state transitions:

```
GRIPLINK_MGRIP (2, "1, 1, 0, 0", TRUE)
```

3.8 Release workpiece - RELEASE

This command causes the selected gripping module to release a gripped workpiece. The command waits until the workpiece has been released. If the device state corresponds to FAULT, the further program will be stopped.



The gripping parameters can be configured via the web interface of the GRIPLINK controller.

Syntax

```
GRIPLINK_RELEASE(<PORT>, <INDEX>, <WSTR>)
```

Parameters

<PORT>	Index of the device port (0 to 31)
<INDEX>	Selected handle (range depends on gripper model)
<WSTR>	Whether or not to wait for state transitions usingWSTR

Return value

none

Example

Release with device at port 0 and grip preset 2. Then, wait for state transitions.

```
GRIPLINK_RELEASE (0, 2, TRUE)
```

3.9 Simultaneous release of workpieces - MRELEASE

This command causes the selected gripping modules to release a gripped workpiece at the same time. The command waits until all addressed gripping modules have each reached the RELEASED state. If one of the devices reaches the FAULT state or a timeout occurs, the program will be stopped.

Syntax

```
GRIPLINK_MRELEASE(<INDEX>, <PORTS>, <MWAITFOR>)
```

Parameters

<INDEX>	Grip preset index
<PORTS>	Selected ports as stringified binary array (e.g. "1,0,0,1" for ports 3 and 0)
<MWAITFOR>	Whether or not to wait for state transitions using MWAITFOR

Return value

none

Example

Perform a synchronous release with the devices at ports 2 and 3 grip using grip preset 2. Then, wait for state transitions:

```
GRIPLINK_MRELEASE (2, "1, 1, 0, 0", TRUE)
```

3.10 Configure grip preset - GRIPCFG

This command parameterizes a grip preset directly from the robot program. This way, the desired grip settings can be saved within the robot program to be transmitted to the GRIPLINK Controller each time the robot program will be started.



The parameters set with this command will not be saved permanently. They will be lost when power-cycling the GRIPLINK Controller. Please use the configuration board on the GRIPLINK's web interface to fully parameterize and permanently save grip presets.

Syntax

```
GRIPLINK_GRIPCFG(<PORT>, <INDEX>, <PARAM0>, <PARAM1>, <PARAM2>, <PARAM3>,  
                 <PARAM4>, <PARAM5>, <PARAM6>, <PARAM7>)
```

Parameter

<PORT>	Index of the device port (0 to 31)
<INDEX>	Index of the grip preset to be configured. The number of available presets depends on the device type.
<PARAM0...7>	Device specific gripping parameters. If one or more parameters are not used, they shall be set to 0. Note: all parameters must be given multiplied by 1000, as the GRIPLINK protocol specifies numbers as floats with a scale of 1/1000!

Specification for grippers from Weiss Robotics:

- Parameter 0: No-part limit (in micrometres)
- Parameter 1: Release limit (in micrometres)
- Parameter 2: Gripping force (in 1/1000 %)
- Parameters 3 - 7: *Not used*



Specification for third party devices: Refer to the device's driver manual!

Return value

none

Example

Configure the grip preset 2 of the gripping module at port 0 as follows:

- No part limit: 3 mm
- Release limit: 10 mm
- Gripping force: 80%

```
GRIPLINK_GRIPCFG(0,2,3000,10000,80000,0,0,0,0,0)
```


3.11 Verify device types - DEVASSERT

This command checks whether the device connected to the specified port corresponds to the expected device. If this is not the case, an error will be raised and the program will be stopped. The check is performed via the IO-Link Vendor and Product ID.

Syntax

```
GRIPLINK_DEVASSERT(<PORT>, <VID>, <PID>)
```

Parameter

<PORT>	Index of the device port (0 to 31)
<VID>	Expected Vendor ID. This is assigned to the manufacturer by the IO-Link user association.
<PID>	Expected Product ID. Identification number of the device assigned by the manufacturer.

Return value

none

Example

Make sure that a gripping module of type IEG 55-020 (Product ID: 20) from Weiss Robotics (Vendor ID: 815) is connected to port 0:

```
GRIPLINK_DEVASSERT (0, 815, 20)
```

3.12 Control gripping force retention - CLAMP

The innovative gripping force retention developed by Weiss Robotics keeps up the gripping force on the workpiece even if the power supply to the gripping module is unexpectedly interrupted. Thanks to the integrated absolute sensor technology, production can be continued immediately when the power supply is restored, even without referencing. Furthermore, CLAMP enables permanent gripping without the gripping module overheating.

This command activates or deactivates the gripping force retention for the selected gripping module.



CLAMP is not available with all gripping modules.

Syntax

```
GRIPLINK_CLAMP(<PORT>, <ENABLE>)
```

Parameter

<PORT>	Index of the device port (0 to 31)
<ENABLE>	Force retention: 1 = on, 0 = off

Return value

none

Example

Activates the gripping force safety device on the gripping module at port 2

```
GRIPLINK_CLAMP (2, 1)
```

3.13 Control of the LED display - LED

This command changes the color and pattern of the light ring of a selected CRG gripping module from Weiss Robotics.

This function is only available for gripping modules of the Weiss Robotics CRG series.



Light patterns can be configured via the web interface of the GRIPLINK controller.



A controllable LED display is not available for all gripping modules.

Syntax

```
GRIPLINK_LED(<PORT>, <PATTERN>)
```

Parameter

<PORT>	Index of the device port (0 to 31)
<PATTERN>	Index of the predefined light pattern (0 to 7)

Return value

none

Example

Activates the light pattern 3 on the gripper connected to port 2:

```
GRIPLINK_LED(2, 3)
```

3.14 Query device state - GETSTATE

This command returns the device state of the device connected to the selected port as an integer value. The possible states can be taken from Appendix A. To make the queries easy to read, constants with the corresponding numerical values have been defined, which can be used instead of numbers. Their names can also be seen in Appendix A.

Syntax

```
GRIPLINK_GETSTATE(<PORT>, <RETURN_VARIABLE>)
```

Parameters

<PORT>	Index of the device port (0 to 31)
<RETURN_VARIABLE>	Name of the return variable
	In case of an error, -1 is stored to this variable

Return value

INT Status code

Example

If the gripper at port 1 holds a part, the robot shall move to the place position:

```
GRIPLINK_DEVSTATE(1,p1_state)
IF (p1_state == DS_HOLDING)
    ; Move to place position
ENDIF
```

3.15 Query of position and sensor values - VALUE

This command reads a measured value from the connected device and returns it as an integer value. Depending on the device, one or more measured values are available, which can be selected via the index to be specified.

Syntax

```
GRIPLINK_VALUE(<PORT>, <INDEX>, <RETURN_VARIABLE>)
```

Parameter

<PORT>	Index of the device port (0 to 31)
<INDEX>	Index of the measured value to be queried
<RETURN_VARIABLE>	Name of the return variable
	In case of an error, -1 is stored to this variable

Return value

INT Status code

Example

Store the finger position (primary value → value index 0) of the gripper connected to port 0 in the variable pos_value:

```
res = GRIPLINK_VALUE(0,0,pos_value)
IF (res != E_SUCCESS)
    ; Perform error handling
ENDIF
```

3.16 Set value - SETVAL

This command sets a device specific numeric value.



The GRIPLINK protocol specifies, that numeric values are transmitted with a factor of 1000.

If the value 1.234 is to be set, it must be transmitted with the factor of 1,000 (corresponds to 1,234).

Syntax

```
GRIPLINK_SETVAL(<PORT>, <INDEX>, <VALUE>)
```

Parameter

<PORT>	Index of the device port (0 to 31)
<INDEX>	Index of the value to be set
<VALUE>	Value to be set

Return value

none

Example

Set the value with index 4 of the device connected to port 2 to the value 3. Wait for value 4 has reached the set target value:

Value to be set 3 => $3 \cdot 1,000 = 3,000$

```
; Set value
GRIPLINK_SETVAL(2,4,3000)
; Wait for value has reached the target value
res = GRIPLINK_WAITVAL(2,4,val)
IF (res != E_SUCCESS)
    ; Perform error handling
ENDIF
```

3.17 Wait for a value to reach the set value - WAITVAL

This command waits until the value with the given index has reached its latest set value. The reached value is then written in the return variable.

An application might consist of a vacuum generator, which was configured using the SETVAL command to generate a specific vacuum level. With WAITVAL it is possible, to wait until the vacuum generator has reached the desired level within a specific time or a timeout occurred.

Syntax

```
GRIPLINK_WAITVAL(<PORT>, <INDEX>, <RETURN_VARIABLE>)
```

Parameter

<PORT>	Index of the device port (0 to 31)
<INDEX>	Index of the value to be waited for
<RETURN_VARIABLE>	Name of the return variable
	In case of an error, -1 is stored to this variable

Return value

INT Status code

Example

Refer to example in 3.16

Anhang A Device states

Gripping state	Code	Constant	Description
NOT CONNECTED	0	DS_NOT_CONNECTED	Device not connected
NOT INITIALIZED	1	DS_NOT_INITIALIZED	Device not initialized
IDLE	2	DS_IDLE	Ready for operation, not active
RELEASED	3	DS_RELEASED	Workpiece released
NO PART	4	DS_NO_PART	No workpiece found
HOLDING	5	DS_HOLDING	Workpiece is held
OPERATING	6	DS_OPERATING	Device ready for operation
FAULT	7	DS_FAULT	Error condition

© 2023 WEISS ROBOTICS GmbH & Co. KG. All rights reserved.

GRIPLINK and PERMAGRIP are registered trademarks of WEISS ROBOTICS GmbH & Co. KG. All other trademarks are the property of their respective owners.

Specifications given in this document are subject to change without notice for the purpose of product improvement. Trademarks are the property of their respective owners. Our products are not intended for use in life support systems or systems where failure could result in personal injury.

