



GRIPLINK PLUGIN FOR FANUC

Version 4.1.0

August 2026



Contents

1	Introduction	4
1.1	Notation and Symbols.....	4
1.2	Intended Use.....	4
1.3	System Requirements.....	4
1.4	License Terms	5
2	Structure of the GRIPLINK Plugin	6
3	Changes from Previous Versions	7
4	Installation	9
4.1	Preparing the robot	9
4.2	Installing the GRIPLINK Core plugin for standard robots.....	10
4.3	Installing the GRIPLINK CRX plugin	12
4.4	Uninstalling the Software	13
4.5	Configuration of the Socket Messaging Interface	14
4.6	Behavior in the Event of an Error	14
5	Command Reference (Core Plugin).....	16
5.1	Establish Connection – CONNECT (1).....	18
5.2	Disconnect – BYE (2)	19
5.3	Check Connected Device – ASRTDEV (3)	20
5.4	Enable Device – ENABLE (11)	21
5.5	Disable Device – DISABLE (12)	22
5.6	Home Reference – HOME (13).....	23
5.7	Grip Workpiece via Preset – GRIP (20)	24
5.8	Release Workpiece via Preset – RELEASE (22).....	26
5.9	Grip Workpiece via Parameters – FLEXGRIP (26)	27
5.10	Release Workpiece via Parameters – FLEXRELEASE (27).....	28
5.11	Configuring a Grip Preset – SGRIPCFG (60).....	29
5.12	Query current Device Status – STATE (10).....	30
5.13	Wait for a Device State Change – WSTR (80).....	31
5.14	Query Device Value – VALUE (41)	32
5.15	Set Device Value – SETVAL (42)	33
5.16	Wait for a Device Value – WAITVAL (43)	34
5.17	Control Force Retention – CLAMP (24).....	35
5.18	Controlling the Status LED – LED (25)	36
5.19	Deprecated Commands	37
5.19.1	Simultaneous Gripping of Workpieces – MGRIP (21)	37
5.19.2	Simultaneous Release of Workpieces – MRELEASE (23)	39
5.19.3	Wait for Multiple Devices to Change Their State – MWAITFOR (81)	40
5.19.4	Read current Finger Position – POS (40).....	41

6	Graphical programming with the FANUC CRX plugin	42
6.1	Functionality	42
6.2	Hardware Commissioning	43
6.3	Preparing the Plugin	44
6.4	Creating Applications with Instructions.....	46
6.5	Stack Size.....	46
6.6	Recommended Program Structure	47
6.7	Overview of Available Commands	47
6.7.1	Establishing a Connection – CONNECT	49
6.7.2	Disconnect – BYE.....	50
6.7.3	Check connected Devices – ASSERTDEV	51
6.7.4	Enable Device – ENABLE	52
6.7.5	Disable Device – DISABLE.....	53
6.7.6	Reference – HOME.....	54
6.7.7	Grip Workpiece via Preset – GRIP	55
6.7.8	Release Workpiece via Preset – RELEASE	56
6.7.9	Grip workpiece via Parameters – FLEXGRIP.....	57
6.7.10	Release workpiece via Parameter – FLEXRELEASE	59
6.7.11	Configuring a Grip Preset – SGRIPCFG	61
6.7.12	Reading the Device State – DEVSTATE.....	63
6.7.13	Wait for a Device State Change – WAITSTATE.....	64
6.7.14	Query Device Value – VALUE	65
6.7.15	Set Device Value – SETVALUE	66
6.7.16	Wait for a Device Value – WAITVALUE	67
6.7.17	Control Force Retention – CLAMP	68
6.7.18	Controlling the Status LED – LED	69
7	Troubleshooting.....	70
7.1	TELNET Output.....	70
7.2	Error Messages	71
7.2.1	Installation Error Message „MEMO-094 Parameters are different“	71
7.2.2	System Error Message „EtherNet Adapter Error (1)“	71
7.2.3	Message: „Could send command [command]: not connected“	72
7.2.4	Message „GRIPLINK: Cmd [command] failed: E_NOT_OPEN“	72
7.2.5	Message „GRIPLINK Version not supported“	73
7.2.6	Message: „GL_COMM 190 : undefined built-in“	73
7.2.7	Message: „GRIPLINK could not connect via SM slot Cx:“	74
7.2.8	Message „Listen failed:“	74
7.2.9	Message „Device x not connected“	74
7.2.10	Message: „Stack Overflow“	74
7.2.11	Message „GRIPLINK: Failed to listen on socket“	75
7.2.12	Connection between robot and GRIPLINK Controller cannot be disconnected.....	75

Appendix A Device Status.....	76
Appendix B Status Codes.....	77

1 Introduction

GRIPLINK technology enables IO-Link-compatible automation components to be connected to robot systems from leading manufacturers via a network connection. The GRIPLINK plugin for FANUC serves as the control-side interface and facilitates the seamless integration of Weiss Robotics' GRIPLINK technology into robot systems from FANUC.

The term „GRIPLINK Controller“ is used in this manual to refer to all products that have integrated GRIPLINK technology.



These include both standalone GRIPLINK Controllers (e.g., GRIPLINK-ET4) as well as devices with embedded GRIPLINK functionality, such as WPG or INTRAPAL servo grippers from Weiss Robotics. All of these products implement the same GRIPLINK protocol and can therefore be controlled in the same way via the plugin.



This manual describes the functions of the GRIPLINK plugin. For information on installation, commissioning, and operation of the GRIPLINK Controller, please refer to the Operating Manual for the respective module. These can be found online at **www.griplink.de**

1.1 Notation and Symbols

For clarity, the following symbols are used in this manual:



Function- or safety-related note. Failure to observe these instructions may endanger the safety of personnel and the system, damage the device, or impair the device's function.



Additional information to aid in understanding the described topic.



Reference to further information.

1.2 Intended Use

The „GRIPLINK-Plugin“ software is intended for communication between the GRIPLINK Controller from Weiss Robotics and a robot controller. It enables the integration and control of GRIPLINK-compatible devices via the standardized GRIPLINK communication interface.

The requirements of the applicable guidelines as well as the installation and operating instructions in this manual must be observed and complied with. Any other or additional use is considered improper. The manufacturer is not liable for any resulting damage.

1.3 System Requirements

This plugin is compatible with **GRIPLINK protocol version 3** and higher.

One of the following FANUC robot controllers is required for operation:

- R-30iB with software version 8.30 (or higher)
- R-30iB plus with software version 9.10 (or higher)

To run the plugin on a CRX series robot, the following robot controller is required:

- R-30iB plus with software version V9.40P/06 (or higher) with tablet teach pendant

The following robot options are required to run the software:

- R632 KAREL
- R648 User Socket Msg



Contact your FANUC dealer to purchase the robot options.



The IP address of the GRIPLINK Controller must be on the same subnet as that of the robot controller. The GRIPLINK Controller manual describes the exact procedure for changing the IP address.

1.4 License Terms

The GRIPLINK plugin is protected by copyright. The applicable license terms are included with the software package. By installing the software, you accept these license terms.

2 Structure of the GRIPLINK Plugin

The GRIPLINK plugin for FANUC consists of two interdependent software components:

Core Plugin

The Core Plugin provides the basic communication interface between the robot controller and the GRIPLINK Controller. It implements the GRIPLINK commands as text-based function calls that can be invoked from robot programs.

The Core Plugin is designed for classic FANUC industrial robot controllers without a tablet teach pendant and enables full control of GRIPLINK-compatible devices via the command reference described in the chapter 5.

CRX Plugin

An extended plugin is also available for robots in the FANUC CRX series. This builds on the Core plugin and provides its functions in the form of graphical instruction blocks in the program editor of the tablet teach pendant.

The instructions can be inserted into robot programs via drag-and-drop and simplify the creation of applications without direct textual programming. Internally, these instructions rely on the functions of the core plugin. Therefore, both programming methods—graphical and text-based—are available in parallel. The use of the CRX plugin is described in Chapter 6.

3 Changes from Previous Versions

New in Version 4.1.0

Version 4.1.0 builds on version 4.0.0 and brings optimizations for the CRX user interface as well as internal stability improvements. All commands, parameters, and signatures remain unchanged. Existing programs continue to run without modification.

GRIPLINK commands now in the Plugin category

The GRIPLINK commands are now located in the Plugin category of the command selection (previously I/O).



After updating from 4.0.0 to 4.1.0, it is recommended to open each GRIPLINK program node once and verify the return register and the command parameters. Nodes without a valid return register are indicated at runtime by a clear error message.

New in Version 4.0.0

With version 4.0.0, the GRIPLINK plugin has been fundamentally revised. The changes made affect both the parameterization and the available commands and their behavior.

Important Note: This version is **not fully backward-compatible**. Existing applications must be reviewed and adapted.

Switch to floating-point parameters (REAL)

The scaled Integer parameters used previously have been replaced by floating-point values (REAL). In earlier versions, physical values (e.g., positions or velocities) were transmitted with a scaling factor (typically 1000).



Starting with version 4.0.0, values are transferred directly in physical units.

This change affects the following commands:

- FLEXGRIP
- FLEXRELEASE
- SGRIPCFG
- VALUE
- SETVAL
- WAITVAL

Effects

- Scaling factors (e.g., factor 1000) may no longer be used
- Parameter values must be converted to actual physical units (e.g., mm instead of mm × 1000)

Example

Up to version 3.x.x	From version 4.0.0
POSITION = 19500	POSITION = 19.5

Commands that are still available (deprecated)

The following commands are still available but are classified as **deprecated**:

- MGRIP
- MRELEASE
- MWAITFOR
- POS

These commands are retained for backward compatibility, but should no longer be used for new projects. For new applications, it is recommended to use the following commands:

- GRIP / RELEASE in combination with DOBLOCK = 0 instead of MGRIP/MRELEASE
- WSTR for status monitoring instead of MWAITFOR
- VALUE for querying device values instead of POS

The new approach offers greater flexibility in program design, as processes and logic can be individually adapted to the specific application. At the same time, it enables better control over program flow and the synchronization of multiple devices. Furthermore, the use of a uniform interface ensures consistent control of different device types.

Note on Migration

Due to the changes in version 4.0.0, existing programs are **not fully backward compatible**. Existing applications must be reviewed and adapted, particularly with regard to the following points:

- Conversion from Integer to REAL parameters
- Removal of scaling factors



It is recommended to perform a complete functional test of the application after migration.

4 Installation



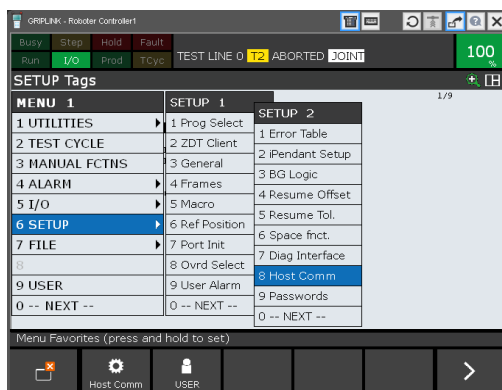
Ensure that you are using the latest version of the GRIPLINK plugin. The latest version can be downloaded at www.griplink.de/software.

4.1 Preparing the robot

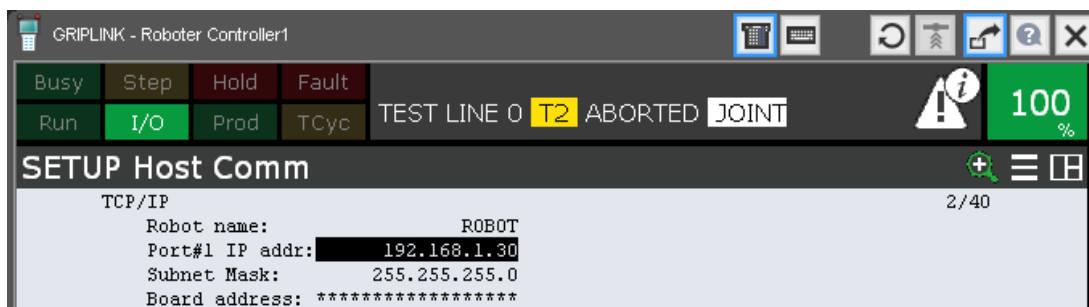
Turn on the robot and configure the IP address (e.g., 192.168.1.30). Make sure that the robot and the GRIPLINK Controller are on the same subnet.

To do this, follow these steps:

1. Select MENU → 6 (SETUP) → 0 (--NEXT--) → 0 (--NEXT--) → 7 (Host Comm).
2. Select „TCP/IP“ and press ENTER.



3. Enter the robot's IP address and subnet mask (in the example, 192.168.1.30, subnet 255.255.255.0).



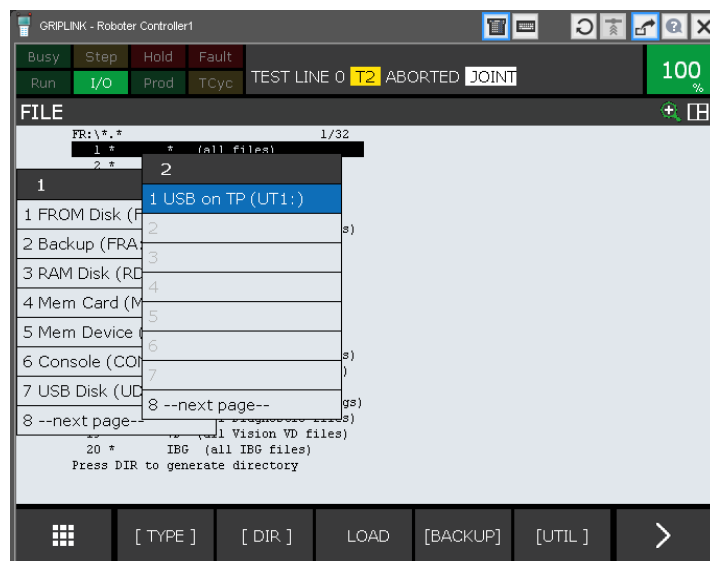
4.2 Installing the GRIPLINK Core plugin for standard robots

For standard robots without Tablet teach pendent, install the Core plugin directly via the teach pendent of your robot controller.

1. Download the plugin file „griplink_fanuc_<Version>.zip“.
2. Extract the ZIP archive to the root directory of a USB flash drive and insert it into the USB port of the teach pendant.
3. Press the Menu button → 7 (FILE) → 1 (File)



4. Press the [UTIL] softkey → 1 (Set Device)
5. Select 8 (-- next page --) → 1 (USB on TP (UT1:))



6. Select „All Files.“ You should now see, among other things, the installation file „SETUP.CM.“

7. Run the installation file. For R-30iB series controllers, you must perform a cold start afterward to apply the parameters.

Verifying the Installation

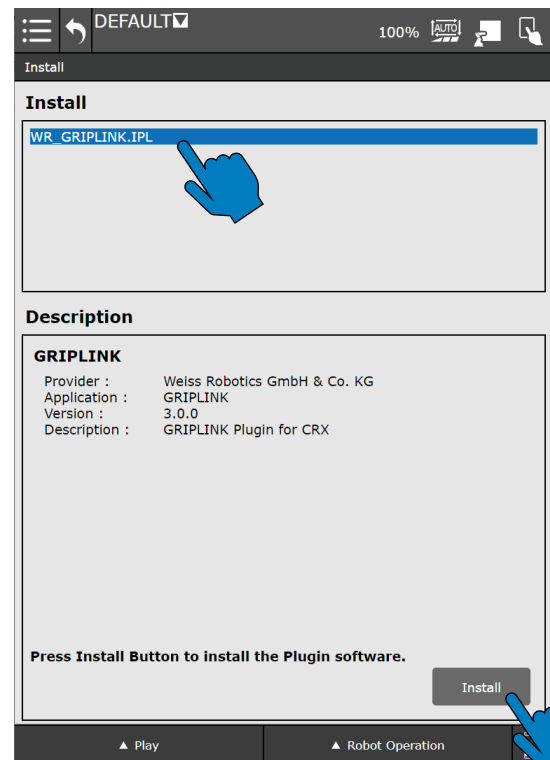
After completing the installation process, press SELECT on the teach pendant. You should now see the „GRIPLINK.PC“ program.

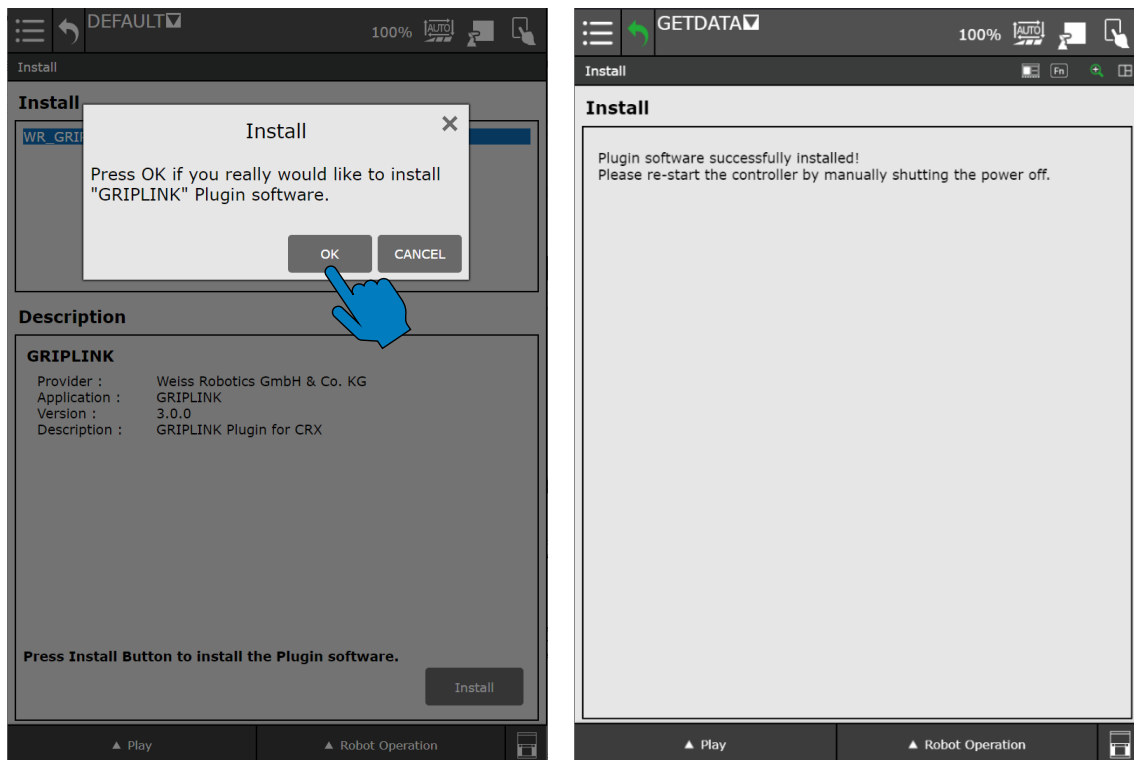
4.3 Installing the GRIPLINK CRX plugin

For FANUC CRX series robots, we recommend installing the CRX-specific GRIPLINK plugin, which provides instruction blocks for graphical programming and thus enables a particularly intuitive integration of GRIPLINK technology into your application.

1. Download the plugin file „griplink_fanuc_<Version>.zip“.
2. Unzip the ZIP archive
3. Copy the file **GRIPLINK.IPL** to the root directory of a USB flash drive and insert it into the USB slot of the teach pendant (UD1).

To install, first start the robot and turn on the tablet teach pendant. Wait until the Tablet TP app opens and the teach pendant has connected to the controller. Then follow these steps:





After being prompted to restart the robot controller, turn it off at the main switch and turn it back on after about ten seconds, or use the restart function via the teach pendant (for more information, refer to the robot controller manual). The teach pendant does not need to be restarted. Wait until the teach pendant has reconnected to the robot controller.



The CRX plugin contains all components of the Core plugin. Therefore, when using the CRX plugin, the Core plugin does not need to be installed separately.

4.4 Uninstalling the Software

To remove the GRIPLINK plugin from your robot, follow these steps.

Uninstallation on standard robots

Run the uninstallation script. This is also part of the software distribution. To do this, follow the steps in the chapter 4.2 and select the uninstallation script „UNINSTALL.CM“ instead of the installation script „SETUP.CM“.

Uninstalling the CRX Plugin

To uninstall the CRX plugin, use the graphical user interface on the tablet teach pendant. In the menu, navigate to the entry „PLUGINS“ -> „Plugin List.“ Select the GRIPLINK plugin and click „Uninstall.“

4.5 Configuration of the Socket Messaging Interface

The Socket Messaging interface is required for communication between the robot and the GRIPLINK Controller. It must be configured as follows:

4. Press the „MENU“ button and navigate to 6 (SETUP) → (Type) → 0 (NEXT) → 8 (HOST COMM). The „SETUP Protocols“ menu appears. Select 3 (SM Socket Messaging Device) and press the SHOW softkey. Select 2 (Clients).
5. Socket Messaging provides you with eight TAGs for free use. Select the TAG you wish to use for communication with the GRIPLINK Controller (TAG 1 in the example).
6. Enter the following data in the input field:

Comment:	GRIPLINK (freely selectable)
Protocol:	SM
Startup State:	START
Server IP/Hostname:	IP address of the GRIPLINK Controller (default setting 192.168.1.40)
Remote Port:	10001



The „Remote Path/Share,“ „Port,“ „Username,“ and „Password“ fields are not required and should be left blank.

7. Use the ACTION soft key to select the value 1 (DEFINE).
„DEFINED“ should now be displayed in the Current State field
8. Use the ACTION soft key to select the value 3 (START).
The „Current State“ field should now display „STARTED“

4.6 Behavior in the Event of an Error

If an error occurs within the GRIPLINK plugin or during communication with the GRIPLINK Controller, the execution of the currently running robot program is always terminated with an error message. As a result, any robot movements currently in progress are also stopped. This prevents the robot from continuing its movement even though a communication or device error has occurred.

An error can occur, for example, if the connection to the GRIPLINK Controller is interrupted, an invalid parameter is transmitted, or a connected device reports a fault state. The same behavior occurs if the addressed device is already in the FAULT state or switches to this state during the execution of a command.

All errors that occur are also logged in the robot controller's log system. The corresponding messages can be accessed via the teach pendant. To do so, navigate to **MENU → 4 (ALARM) → 4 (Appl Log)**. This log stores the error messages generated by the plugin along with additional diagnostic information.

Some messages are also output in greater detail in the robot controller's **USER log**. This additional information can be helpful when analyzing communication problems or unexpected device behavior.

For a more detailed diagnosis during commissioning or troubleshooting, the robot controller's TELNET output can also be used. Instructions for activating and using this output can be found in the section 7.1.

An overview of typical error messages, their causes, and possible solutions can be found in the section 7 - Troubleshooting.

5 Command Reference (Core Plugin)

The GRIPLINK plugin provides a range of functions for controlling and monitoring GRIPLINK-compatible devices. These commands can be used to activate and control gripper modules, as well as to query their statuses and process values.

The commands are executed via the **GRIPLINK.PC** interface program, which is integrated into the robot program. The command to be executed is passed to the interface program along with the required parameters. The following sections describe the available commands of the GRIPLINK plugin, as well as their parameters, return values, and typical application examples.



To use the plugin on controllers of the CRX robot series, the function calls defined in the CRX plugin (beginning with „IPL_WR_GRIPLINK“) should be used.

Calling a GRIPLINK Command

GRIPLINK commands are called via the GRIPLINK.PC interface program. The desired command is passed to the interface program along with the corresponding parameters.

```
CALL GRIPLINK(<command>, <parameter>)
```

Depending on the robot type and software version, the command can be selected and the parameters passed via the Argument Wizard.

Argument Wizard

On controllers with software version 9.10 or higher (R-30iB plus series), the command and parameters can be selected directly via the Argument Wizard.

On controllers with software version 8.30 (R-30iB series), the command can be selected via the Argument Wizard. The parameter values required for the command must be entered via parameters A through K. Unnecessary parameters can be set to 0.

Return Values

The return values of the commands are stored in a register. The index of this register or the start index of a register area can be specified via the „RETREGIDX“ (Return Register Index) parameter.

Basic Program Flow

1. Establish a connection to the GRIPLINK Controller using **CONNECT**.
This command must be executed once at the beginning of the program.
2. Optional: Initialize or reference devices.
For certain device types (e.g., servo grippers without absolute encoders), a homing run with **HOME** may be required. Other device types, such as sensors, do not require referencing.
3. Activate devices using **ENABLE** and configure them, provided the device in question supports this.

4. Execute device functions, e.g., gripping or releasing for gripper modules (GRIP, RELEASE), or query device statuses and values.
5. Disconnect from the GRIPLINK Controller using **BYE** if the interface must be physically disconnected (e.g., during a tool change) or when the program ends.

To ensure reliable communication and optimal performance, the application should be implemented so that the connection to the GRIPLINK Controller remains active throughout the entire program runtime. In this case, **CONNECT** is executed once at the start of the program, and **BYE** is called only at the end of the program or in the event of an error.

Example: Minimal program

```
! Connect to GRIPLINK Controller
CALL GRIPLINK(CONNECT=1,COM_TAG=1)

! Enable gripper
CALL GRIPLINK(ENABLE=11,PORT=0,RETREGIDX=1)

! Grasp part
CALL GRIPLINK(GRIP=20,PORT=0,PRESETIDX=1,DOBLOCK=1)

! Release part
CALL GRIPLINK(RELEASE=22,PORT=0,PRESETIDX=1,DOBLOCK=1)

! Disconnect
CALL GRIPLINK(BYE=2)
```

The available commands of the GRIPLINK plugin are described below.

5.1 Establish Connection – CONNECT (1)

This command establishes the connection between the GRIPLINK Controller and the robot controller. The index of the COM tag previously configured according to the chapter 4.5 is passed as a parameter. The command waits until the connection is established. If the GRIPLINK Controller cannot be reached at the specified address, an error is triggered and program execution is halted.

The command also checks whether the connected GRIPLINK Controller meets the minimum requirements of the GRIPLINK protocol (see section 1.3).

Syntax

R-30iB plus:	GRIPLINK(CONNECT=1, <COM_TAG>)
R-30iB:	GRIPLINK(CONNECT=1, A=<COM_TAG>, B=0, C=0, D=0, E=0, F=0, G=0, H=0, I=0, J=0, K=0)

Parameters

<COM_TAG>	Index of the COM tag (1 to 8)
-----------	-------------------------------

Return Values

None

Common causes of errors

COM tag not initialized correctly

→ Check settings

GRIPLINK Controller not connected to the robot

→ Check network connection and power supply

GRIPLINK Controller configured differently

→ Check the GRIPLINK Controller's network settings via the web interface

The GRIPLINK Controller's protocol version does not match the plugin

→ Update the GRIPLINK Controller's firmware

Example

Establish a connection between the robot and the GRIPLINK using COM port 1:

```
CALL GRIPLINK (CONNECT=1, COM_TAG=1)
```



If GRIPLINK commands are executed before a CONNECT, this triggers an error. This command must be called before any other GRIPLINK command.

5.2 Disconnect – BYE (2)

This command disconnects the connection between the robot controller and the GRIPLINK Controller. This may be necessary if the network connection to the GRIPLINK Controller is interrupted by a tool changer.



Disconnecting the connection is not required during normal program execution and should only be used with caution.

Syntax

R-30iB plus:	GRIPLINK(BYE=2)
R-30iB:	GRIPLINK(BYE=2, A=0, B=0, C=0, D=0, E=0, F=0, G=0, H=0, I=0, J=0, K=0)

Parameters

None

Return Values

None

Example

Disconnect the connection between the robot and the GRIPLINK Controller when a tool change is required:

```
CALL GRIPLINK (BYE=2)
```

```
! -> Change tool here!
```

```
CALL GRIPLINK (CONNECT=1, COM_TAG=1)
```

5.3 Check Connected Device – ASRTDEV (3)

This command checks whether the intended device is connected to the selected port. If the device connected to the port does not have the specified vendor and product IDs, the robot program is paused.

Syntax

R-30iB plus:	GRIPLINK(ASRTDEV=3, <PORT>, <VID>, <PID>)
R-30iB:	GRIPLINK(ASRTDEV=3, A=<PORT>, B=<VID>, C=<PID>, D=0, E=0, F=0, G=0, H=0, I=0, J=0, K=0)

Parameters

<PORT>	Device index (0 to 31)
<VID>	Vendor ID specified in the IODD of the expected device
<PID>	Product ID specified in the IODD of the expected device

Return values

None

Example

Ensure that the IEG 55-020 device from Weiss Robotics (VID 815, PID 20) is connected to port 3:

```
CALL GRIPLINK (ASRTDEV=3, PORT=3, VID=815, PID=20)
```



You can find the Vendor and Product IDs in the device description provided by the manufacturer or in the IODD.

5.4 Enable Device – ENABLE (11)

This command activates the device connected to the selected port. IO-Link gripper modules then switch to the RELEASED state and execute the release action of the currently selected grip preset.

Syntax

R-30iB plus:	GRIPLINK(ENABLE=11, <PORT>, <RETREGIDX>)
R-30iB:	GRIPLINK(ENABLE=11, A=<PORT>, B=<RETREGIDX>, C=0, D=0, E=0, F=0, G=0, H=0, I=0, J=0, K=0)

Parameters

<PORT>	Port index (0 to 31)
<RETREGIDX>	Index of the return value register (1 to 200)

Return values

R[<RETREGIDX>]	Current device status
----------------	-----------------------

Example

Enable the device on port 0:

```
CALL GRIPLINK (ENABLE=11, PORT=0, RETREGIDX=1)
```



Behavior may vary depending on the connected device. Please refer to the instructions in the respective device driver manual!

5.5 Disable Device – DISABLE (12)

This command disables the device connected to the selected port. The command can be used, for example, to release the fingers of a gripper module or to temporarily disable the laser beam of a distance sensor.

Error Acknowledgment Using DISABLE

DISABLE can also be used to acknowledge device errors, for example, if the connected device is in the FAULT state.

Syntax

R-30iB plus:	GRIPLINK(DISABLE=12, <PORT>, <RETREGIDX>)
R-30iB:	GRIPLINK(DISABLE=12, A=<PORT>, B=<RETREGIDX>, C=0, D=0, E=0, F=0, G=0, H=0, I=0, J=0, K=0)

Parameters

<PORT>	Port index (0 to 31)
<RETREGIDX>	Index of the return value register (1 to 200)

Return values

R[<RETREGIDX>]	Current device status
----------------	-----------------------

Example

A servo gripper is connected to Port 2. At system startup, the system should check for any error conditions before referencing and acknowledge them:

```
CALL GRIPLINK (STATE=10, PORT=2, RETREGIDX=1)

! Test for FAULT state
IF (R[1]=7) THEN

    ! Gripper in FAULT state (7)
    CALL GRIPLINK (DISABLE=12, PORT=2, RETREGIDX=1)
    WAIT 0.1 (sec)
    CALL GRIPLINK (ENABLE=11, PORT=2, RETREGIDX=1)
ENDIF
CALL GRIPLINK (HOME=13, PORT=2, RETREGIDX=1)
```

5.6 Home Reference – HOME (13)

Performs a homing cycle/reference drive on the selected device. The command executes the device's built-in home-referencing function and waits until it is complete.

Syntax

R-30iB plus:	GRIPLINK(HOME=13, <PORT>, <RETREGIDX>)
R-30iB:	GRIPLINK(HOME=13, A=<PORT>, B=<RETREGIDX>, C=0, D=0, E=0, F=0, G=0, H=0, I=0, J=0, K=0)

Parameters

<PORT>	Port index (0 to 31)
<RETREGIDX>	Index of the return value register (1 to 200)

Return values

R[<RETREGIDX>]	Current device status
----------------	-----------------------

Example

Reference the gripper module on port 2:

```
CALL GRIPLINK (HOME=13, PORT=2, RETREGIDX=1)
```



Behavior may vary depending on the connected device. Please refer to the instructions in the respective device driver manual!

5.7 Grip Workpiece via Preset – GRIP (20)

The GRIP command performs a grip using the specified grip preset. The optional DOBLOCK parameter can be used to specify whether the command returns immediately after the movement starts or only after it is complete:

DOBLOCK = 0 → The command returns immediately after the movement starts (non-blocking).

DOBLOCK = 1 → The command waits until the gripping process is complete (blocking).

The current state of the gripping module can be queried at any time using the STATE command. In non-blocking mode (DOBLOCK = 0), the WSTR command can also be used to wait for a state change to detect the end of the gripping process.



The gripping parameters can be configured via the GRIPLINK Controller's web interface.

Syntax

R-30iB plus:	GRIPLINK(GRIP=20, <PORT>, <PRESETIDX>, <DOBLOCK (opt)>)
R-30iB:	GRIPLINK(GRIP=20, A=<PORT>, B=<PRESETIDX>, C=<DOBLOCK (opt)>, D=0, E=0, F=0, G=0, H=0, I=0, J=0, K=0)

Parameters

<PORT>	Port index (0 to 31)
<PRESETIDX>	Grip preset index (valid range depends on the device)
<DOBLOCK (opt)>	Optional. If 0, the command returns immediately after the motion starts. If 1, GRIPLINK waits until the gripping operation is complete. Default value: 0

Return values

None

Example

The gripper module on port 0 should grip using grip preset 2. If no workpiece is found, the gripper should wait, reopen, and try again:

```
LBL[1]

! Grip
CALL GRIPLINK (GRIP=20, PORT=0, PRESETIDX=2)

! Wait until the gripper changed its state
CALL GRIPLINK (WSTR=80, PORT=0, RETREGIDX=1)
IF (R[1]=5) THEN
    ! Holding a part
    JMP LBL[2]
ENDIF
```

```

! Release and try again...
CALL GRIPLINK(RELEASE=22,PORT=0,PRESETIDX=2)
! Wait until the gripper changed its state
CALL GRIPLINK(WSTR=80,PORT=0,RETREGIDX=1)

WAIT 0.5(sec)
JMP LBL[1]
LBL[2]
! Workpiece gripped!

```

The same example with blocking commands:

```

LBL[1]

! Grip
CALL GRIPLINK(GRIP=20,PORT=0,PRESETIDX=2,DOBLOCK=1)

! Read device state state
CALL GRIPLINK(STATE=10,PORT=0,RETREGIDX=1)
IF (R[1]=5) THEN
    ! Holding a part
    JMP LBL[2]
ENDIF

! Release and try again...
CALL GRIPLINK(RELEASE=22,PORT=0,PRESETIDX=2,DOBLOCK=1)
WAIT 0.5(sec)
JMP LBL[1]

LBL[2]
! Workpiece gripped!

```

5.8 Release Workpiece via Preset – RELEASE (22)

The RELEASE command releases a previously gripped workpiece using the parameters of the specified grip preset. The optional DOBLOCK parameter can be used to control whether the command returns immediately after the start of the movement or only after its completion:

DOBLOCK = 0 → The command returns immediately after the movement starts (non-blocking).

DOBLOCK = 1 → The command waits until the release is complete (blocking).

The current status of the gripper module can be queried at any time using the STATE command. In non-blocking mode (DOBLOCK = 0), the WSTR command can also be used to wait for a state change to detect the end of the release.



The gripping parameters can be configured via the GRIPLINK Controller's web interface.

Syntax

R-30iB plus:	GRIPLINK(RELEASE=22, <PORT>, <PRESETIDX>, <DOBLOCK (opt)>)
R-30iB:	GRIPLINK(RELEASE=22, A=<PORT>, B=<PRESETIDX>, C=<DOBLOCK (opt)>, D=0, E=0, F=0, G=0, H=0, I=0, J=0, K=0)

Parameter

<PORT>	Port index (0 to 31)
<PRESETIDX>	Grip preset index (valid range depends on the device)
<DOBLOCK (opt)>	Optional. If 0, the command returns immediately after the motion starts. If 1, GRIPLINK waits until the release motion is complete. Default value: 0

Return values

None

Example

See example in chapter 5.7

5.9 Grip Workpiece via Parameters – FLEXGRIP (26)

The FLEXGRIP command performs a gripping operation with freely selectable motion parameters.

Unlike the GRIP command, no predefined grip configuration (preset) is used here; instead, the motion sequence is determined entirely by the command parameters.

This allows the user to specify individual target positions, forces, speeds, and accelerations without having to rely on a saved grip preset.

The command can be executed either in blocking or non-blocking mode.

In blocking mode, the command returns only after the gripping process is complete.

After the command is called, the gripping operation starts with the specified motion parameters. If the parameter DOBLOCK = 0 is set, the command returns immediately while the motion is executed in the background. If DOBLOCK = 1, the command waits until the operation is complete (workpiece gripped or no workpiece detected).

The current status can be queried at any time using the STATE command. Alternatively, the completion of the command can be detected with WSTR if the command was executed in non-blocking mode.

Syntax

R-30iB plus:	GRIPLINK(FLEXGRIP=26, <PORT>, <POSITION>, <FORCE>, <SPEED>, <ACCELERATION>, <DOBLOCK>)
R-30iB:	GRIPLINK(FLEXGRIP=26, A=<PORT>, B=<POSITION>, C=<FORCE>, D=<SPEED>, E=<ACCELERATION>, F=<DOBLOCK>, G=0, H=0, I=0, J=0, K=0)

Parameter

<PORT>	Port index (0 – 31)
<POSITION>	Target position for „No Part Limit“ in mm
<FORCE>	Gripping force in N
<SPEED>	Gripping speed in mm/s (0 = use automatic calculation)
<ACCELERATION>	Finger acceleration in mm/s ² (0 = use default value)
<DOBLOCK>	0 = non-blocking, 1 = blocking

Example

To grip a workpiece with the gripper on port 0 and 60 N force, no-part-limit = 15 mm, speed = 45.5 mm/s, standard acceleration, non-blocking:

```
CALL GRIPLINK(FLEXGRIP=26, PORT=0, POSITION=15.0, FORCE=60.0,
SPEED=45.5, ACCELERATION=0, DOBLOCK=0)
```

Blocking variant:

```
CALL GRIPLINK(FLEXGRIP=26, PORT=0, POSITION=15.0, FORCE=60.0,
SPEED=45.5, ACCELERATION=0, DOBLOCK=1)
```

5.10 Release Workpiece via Parameters – FLEXRELEASE (27)

The FLEXRELEASE command releases a previously gripped workpiece using freely selectable motion parameters. It can be used both for simple release and for targeted pre-positioning of a servo gripper. The motion parameters are defined entirely via the command arguments.

After the command is called, the gripper module opens the fingers to the specified target position. In non-blocking execution, the command returns immediately while the movement runs in the background. In blocking execution, the command waits until the movement is complete.

Progress can be monitored using the STATE command or the WSTR query command.

The state typically changes to DS_RELEASED upon completion.

Syntax

R-30iB plus:	GRIPLINK(FLEXRELEASE=27, <PORT>, <POSITION>, <SPEED>, <ACCELERATION>, <DOBLOCK>)
R-30iB:	GRIPLINK(FLEXRELEASE=27, A=<PORT>, B=<POSITION>, C=<SPEED>, D=<ACCELERATION>, E=<DOBLOCK>, F=0, G=0, H=0, I=0, J=0, K=0)

Parameter

<PORT>	Port index (0 – 31)
<POSITION>	Target position (release limit) in mm
<SPEED>	Opening speed in mm/s (0 = use default value)
<ACCELERATION>	Finger acceleration in mm/s ² (0 = use default value)
<DOBLOCK>	0 = non-blocking, 1 = blocking

Example

Pre-positioning the gripper at Port 0 to 19.5 mm using default parameters, blocking:

```
CALL GRIPLINK(FLEXRELEASE=27, PORT=0, POSITION=19.5, SPEED=0,
ACCELERATION=0, DOBLOCK=1)
```

Non-blocking:

```
CALL GRIPLINK(FLEXRELEASE=27, PORT=0, POSITION=19.5, SPEED=0,
ACCELERATION=0, DOBLOCK=0)
```


5.12 Query current Device Status – STATE (10)

This command returns the device status of the selected device.

Syntax

R-30iB plus:	GRIPLINK(STATE=10, <PORT>, <RETREGIDX>)
R-30iB:	GRIPLINK(STATE=10, A=<PORT>, B=<RETREGIDX>, C=0, D=0, E=0, F=0, G=0, H=0, I=0, J=0, K=0)

Parameters

<PORT>	Port index (0 to 31)
<RETREGIDX>	Index of the return value register (1 to 200)

Return values

R[<RETREGIDX>]	Current device status
----------------	-----------------------

Example

Wait until the device status of the gripper module on port 2 is „HOLDING“ (4):

```
LBL[1]
```

```
! Get device state
```

```
CALL GRIPLINK(STATE=10, PORT=2, RETREGIDX=1)
```

```
IF R[1]=5, JMP LBL[2]
```

```
WAIT 0.01(sec)
```

```
! Do some timeout processing, e.g. using a counter variable
```

```
JMP LBL[1]
```

```
...
```

```
LBL[2]
```

```
! Gripper at port 2 is in HOLDING state
```

5.13 Wait for a Device State Change – WSTR (80)

The WSTR command waits for a change in the device state at the specified port. It can be used, for example, to check whether an asynchronously executed release motion (RELEASE with DOBLOCK=0) has been completed.



If the state change does not occur within a specified time interval, a timeout error is triggered and program execution is interrupted.

Syntax

R-30iB plus: GRIPLINK(WSTR=80, <PORT>, <RETREGIDX>)
R-30iB: GRIPLINK(WSTR=80, A=<PORT>, B=<RETREGIDX>, C=0, D=0, E=0, F=0,
 G=0, H=0, I=0, J=0, K=0)

Parameters

<PORT> Port index (0 to 31)
<RETREGIDX> Return value register index (1 to 200)

Return values

R[<RETREGIDX>] Current device status

Example

Pre-position the gripper fingers using Preset 2 while the robot moves to the pickup position P[1]. Upon arrival, wait until the fingers are open, then move to the pickup position P[2] to grip the workpiece.

```
! Start pre-positioning the fingers
CALL GRIPLINK(RELEASE=22, PORT=0, PRESETIDX=2, DOBLOCK=0)

! Move to approach point
J P[1]...

! Wait until the gripper changes state
CALL GRIPLINK(WSTR=80, PORT=0, RETREGIDX=1)
IF (R[1]<>3) THEN
  ! Not in RELEASED (3) state. Handle ERROR here!
ENDIF

! Move to stack point
J P[2]...

! Grasp workpiece
CALL GRIPLINK(GRIP=20, PORT=0, PRESETIDX=2, DOBLOCK=1)
```

5.14 Query Device Value – VALUE (41)

This command reads an indexed device value.



The return value is written to the register with the specified index.



The valid value range for the value index is device-specific and can be found in the driver or device documentation for the respective device.

Syntax

R-30iB plus:	GRIPLINK(VALUE=41, <PORT>, <VALUEIDX>, <RETREGIDX>)
R-30iB:	GRIPLINK(VALUE=41, A=<PORT>, B=<VALUEIDX>, C=<RETREGIDX>, D=0, E=0, F=0, G=0, H=0, I=0, J=0, K=0)

Parameter

<PORT>	Port index (0 to 31)
<VALUEIDX>	Index of the value to be read (permissible value range depends on the device)
<RETREGIDX>	Index of the return value register (1 to 200)

Return Values

R[<RETREGIDX>]	Current device value
----------------	----------------------

Example

Read the primary value (Value Index 0, here: finger position) from the device at Port 3, a servo gripper, and store the value in Register 31. Use this for an IO/NIO check.

```
! Grip workpiece
CALL GRIPLINK (GRIP=20, PORT=3, PRESETIDX=0, DOBLOCK=1)

! Get part width
CALL GRIPLINK (VALUE=41, PORT=3, VALUEIDX=0, RETREGIDX=31)

! Is part width 30±0.1?
IF (R[31]<29.9 OR R[31]>30.1) THEN

    ! Part is NIO -> Move to drop position
    J P[99]
    CALL GRIPLINK (RELEASE=22, PORT=3, PRESETIDX=0, DOBLOCK=1)
ELSE

    ! Part is IO -> Move to Use position
    J P[20]
    CALL GRIPLINK (RELEASE=22, PORT=3, PRESETIDX=0, DOBLOCK=1)
ENDIF
```

5.15 Set Device Value – SETVAL (42)

This command can be used to set a device-specific value. The value can be selected via an index, which has a valid value range that depends on the device. The value to be written can have different meanings depending on the connected device.



Refer to the driver manual for the respective device for the value ranges of the value index and the value to be written.

Syntax

R-30iB plus:	GRIPLINK(SETVAL=42, <PORT>, <VALUEIDX>, <VALUE>)
R-30iB:	GRIPLINK(SETVAL=42, A=<PORT>, B=<VALUEIDX>, C=<VALUE>, D=0, E=0, F=0, G=0, H=0, I=0, J=0, K=0)

Parameter

<PORT>	Port index (0 to 31)
<VALUEIDX>	Index of the value to be written (permissible value range is device-dependent)
<VALUE>	Value to be written (permissible value range depends on the device)

Return values

None

Example

Set the value with index 2 to 42.5 on port 0 of the device and wait until the value is within the range of ± 0.1 , but wait no longer than 10 seconds:

```
CALL GRIPLINK (SETVAL=42, PORT=0, VALUEIDX=2, VALUE=42.5)
```

```
! Wait until the value reaches the set value
```

```
CALL GRIPLINK (WAITVAL=43, PORT=0, VALUEIDX=2, THRESHOLD=42.5,  
WINDOWSIZE=0.1, TIMEOUT=10000)
```

5.16 Wait for a Device Value – WAITVAL (43)

The WAITVAL command waits until a specific device value reaches the previously set target value or falls within a defined tolerance range. It can be used, for example, to check whether a sensor or position value has reached a target range before the program continues.

The command blocks until one of the following conditions is met:

- The monitored value is within the specified tolerance window, or
- the timeout has been exceeded.



The WAITVAL command is compatible with GRIPLINK protocol version 3. The monitored value corresponds to the index value that can be read via VALUE.



If the value to be monitored does not reach the detection window within the specified TIMEOUT time interval, a timeout error is triggered and program execution is interrupted.

Syntax

R-30iB plus:	GRIPLINK(WAITVAL=43, <PORT>, <VALUEIDX>, <THRESHOLD>, <WINDOWSIZE>, <TIMEOUT>)
R-30iB:	GRIPLINK(WAITVAL=43, A=<PORT>, B=<VALUEIDX>, C=<THRESHOLD>, D=<WINDOWSIZE>, E=<TIMEOUT>, F=0, G=0, H=0, I=0, J=0, K=0)

Parameters

<PORT>	Port index (0 to 31)
<VALUEIDX>	Index of the value to be monitored (permissible value range depends on the device)
<THRESHOLD>	Target value in the unit of the measured value
<WINDOWSIZE>	Width of the tolerance window symmetrical to the target value
<TIMEOUT>	Timeout value in milliseconds

Return values

None

Example

See example in chapter 5.15

5.17 Control Force Retention – CLAMP (24)

With the CLAMP command, gripper modules with integrated gripping force retention can de-energize the drive in the HOLDING state to reduce the gripper module's power consumption. The workpiece continues to be held securely.

If CLAMP is activated, the drive switches off in the HOLDING state. If CLAMP is deactivated, gripper force control remains active in the HOLDING state and the gripper module controller continuously adjusts the gripper force.



The CLAMP command is not available on all gripping modules.

Syntax

R-30iB plus:	<code>GRIPLINK(CLAMP=24, <PORT>, <ENABLE>)</code>
R-30iB:	<code>GRIPLINK(CLAMP=24, A=<PORT>, B=<ENABLE>, C=0, D=0, E=0, F=0, G=0, H=0, I=0, J=0, K=0)</code>

Parameters

<PORT>	Port index (0 to 3)
<ENABLE>	Grip force maintenance: 1 = enabled, 0 = disabled

Return values

None

Example

Enables grip force maintenance in the gripper module on port 2:

```
CALL GRIPLINK (CLAMP=24, PORT=2, ENABLE=1)
```

5.18 Controlling the Status LED – LED (25)

This command activates a predefined LED preset for the status LED of a connected device. This allows the operating status or progress of a process to be visually indicated directly on the end effector.



LED presets can be configured via the GRIPLINK Controller's web interface.



The LED command is not supported by all devices.

Syntax

R-30iB plus:	GRIPLINK(LED=25, <PORT>, <PRESETIDX>)
R-30iB:	GRIPLINK(LED=25, A=<PORT>, B=<PRESETIDX>, C=0, D=0, E=0, F=0, G=0, H=0, I=0, J=0, K=0)

Parameter

<PORT>	Port index (0 to 31)
<PRESETIDX>	Index of the LED preset (0 to 7)

Return values

None

Example

Execute a grasping command with the gripper module at port 3 and read the grasping position. Change the color of the light ring to LED preset 1 if the finger position is greater than or equal to 8.1 mm, and to LED preset 2 if it is less:

```
! Grasp with the gripper at port 3 and preset 0, wait for state transition
```

```
CALL GRIPLINK (GRIP=20, PORT=3, PRESETIDX=0, DOBLOCK=0)
```

```
CALL GRIPLINK (WSTR=80, PORT=3, RETREGIDX=1)
```

```
! Read finger position and update LED
```

```
CALL GRIPLINK (VALUE=41, PORT=3, VALUEIDX=0, RETREGIDX=2)
```

```
IF (R[2] >= 8.1) THEN
```

```
    CALL GRIPLINK (LED=25, PORT=3, PRESETIDX=1)
```

```
ELSE
```

```
    CALL GRIPLINK (LED=25, PORT=3, PRESETIDX=2)
```

```
ENDIF
```

5.19 Deprecated Commands

This section lists commands that are still included in the GRIPLINK plugin for backward compatibility reasons, but whose use is no longer recommended.

These commands have been replaced by functionally equivalent but more flexible mechanisms. In particular, the newer commands allow for finer control of the program flow and better adaptation to different applications and device types.

Therefore, these deprecated commands should not be used in new programs. Instead, it is recommended to use the corresponding current commands and programming patterns.

5.19.1 Simultaneous Gripping of Workpieces – MGRIP (21)

This command executes a simultaneous gripping command using the selected gripping modules. The MWAITFOR command can then be used to wait for a change in the status of the gripping modules.



This command is deprecated and is provided only for compatibility reasons. For new applications, it is recommended to use GRIP with DOBLOCK = 0 in combination with WSTR instead.

Syntax

R-30iB plus:	CALL GRIPLINK(MGRIP=21, <PORTS>, <PRESETIDX>)
R-30iB:	CALL GRIPLINK(MGRIP=21, A=<PORTS>, B=<PRESETIDX>, C=0, D=0, E=0, F=0, G=0, H=0, I=0, J=0, K=0)

Parameter

<PORTS>	Selected gripper modules as a 32-bit vector: Bit x = 1: Device at port (1 <<x>) is selected Bit x = 0: Device at port (1 <<x>) is not selected Example: Select grippers at ports 0 and 3 →Set bits 0 and 3 →Ports = (1 << 0) + (1 << 3) = 1 + 8 = 9
<PRESETIDX>	Index of the grip preset (permissible value range depends on the device)

Return values

None

Example

Gripper modules at ports 0 and 3 grip the workpiece using grip preset 2 and then wait for state changes:

! Grip with grippers at ports 0 and 3

```
CALL GRIPLINK(MGRIP=21,PORTS=9,PRESET=2)
CALL GRIPLINK(MWAITFOR=81,PORTS=9)
! Get states of the grippers
CALL GRIPLINK(STATE=10,PORT=0,RETREGIDX=1)
CALL GRIPLINK(STATE=10,PORT=3,RETREGIDX=4)
...
! Release grippers at ports 0 and 3
CALL GRIPLINK(MRELEASE=23,PORTS=9,PRESET=2)
CALL GRIPLINK(MWAITFOR=81,PORTS=9)
! Get actual states of the grippers
CALL GRIPLINK(STATE=10,PORT=0,RETREGIDX=1)
CALL GRIPLINK(STATE=10,PORT=3,RETREGIDX=4)
```

5.19.2 Simultaneous Release of Workpieces – MRELEASE (23)

This command executes a simultaneous release command with the selected gripper modules. The MWAITFOR command can then be used to wait for a state change of the gripper modules.



This command is deprecated and is provided only for compatibility reasons. For new applications, it is recommended to use RELEASE with DOBLOCK = 0 in combination with WSTR instead.

Syntax

R-30iB plus:	CALL GRIPLINK(MRELEASE=23, <PORTS>, <PRESETIDX>)
R-30iB:	CALL GRIPLINK(MRELEASE=23, A=<PORTS>, B=<PRESETIDX>, C=0, D=0, E=0, F=0, G=0, H=0, I=0, J=0, K=0)

Parameter

<PORTS>	Selected gripper modules as a 32-bit vector: Bit x = 1: Device at port (1 <<x>) is selected Bit x = 0: Device at port (1 <<x>) is not selected
---------	--

Example: Select grippers at ports 0 and 3

→ Set bits 0 and 3

→ Ports = (1 << 0) + (1 << 3) = 1 + 8 = 9

<PRESETIDX>	Index of the grip preset (valid range depends on the device)
-------------	--

Return values

None

Example

See example in 5.19.1

5.19.3 Wait for Multiple Devices to Change Their State – MWAITFOR (81)

The MWAITFOR command waits for multiple devices to change their state.



If the state change does not occur on at least one selected port within a specified time interval, a timeout error is triggered and program execution is interrupted.



This command is deprecated and is provided only for compatibility reasons. For new applications, it is recommended to use WSTR instead.

Syntax

R-30iB plus:	GRIPLINK(MWAITFOR=81, <PORTS>)
R-30iB:	GRIPLINK(MWAITFOR=81, A=<PORTS>, B=0, C=0, D=0, E=0, F=0, G=0, H=0, I=0, J=0, K=0)

Parameters

<PORTS>	Selected gripper modules as a 32-bit vector: Bit x = 1: Device at port (1 <<x>) is selected Bit x = 0: Device at port (1 <<x>) is not selected
---------	--

Example: Select grippers at ports 0 and 3

→ Set bits 0 and 3

→ Ports = (1 << 0) + (1 << 3) = 1 + 8 = 9

Return values

None

Example

See example in 5.19.1

5.19.4 Read current Finger Position – POS (40)

This command reads the current finger position of the selected gripper module and stores it in the register with the given index.



This command is deprecated. Use VALUE with VALUEIDX = 0 instead.

Syntax

R-30iB plus:	CALL GRIPLINK(POS=40, <PORT>, <RETREGIDX>)
R-30iB:	CALL GRIPLINK(POS=40,A=<PORT>, B=<RETREGIDX>, C=0, D=0, E=0, F=0, G=0, H=0, I=0, J=0, K=0)

Parameters

<PORT>	Gripper module index (0 to 31)
<RETREGIDX>	Index of the return value register (1 to 100)

Return values

R[<RETREGIDX>]	Finger position in mm
----------------	-----------------------

Example

Execute the subroutine „PROG“ if the finger position of the gripper module on port 2 is greater than 10.5 mm:

```
CALL GRIPLINK(POS=40, PORT=2, RETREGIDX=42)
IF R[42]>10.5, CALL PROG
```

6 Graphical programming with the FANUC CRX plugin

For robots in the FANUC CRX series, an extended plugin is available in addition to the core plugin, which enables graphical (instruction-based) programming of GRIPLINK functions.

The plugin provides GRIPLINK commands as instruction blocks in the program editor of the tablet teach pendant. These can be inserted into a robot program via drag-and-drop and then configured using a detailed view.

This allows GRIPLINK-compatible devices to be integrated quickly and clearly into robot programs without having to manually program the underlying function calls of the core plugin.

The following sections describe how to set up the plugin, perform hardware commissioning, and use the available instructions in the robot program.

6.1 Functionality

The CRX plugin consists of a configuration page and a set of program instructions through which the robot controller interacts with a connected GRIPLINK device.

The configuration page is used to configure basic settings of the plugin, such as selecting the COM port used for communication with the GRIPLINK Controller.

Instructions provide the commands of the core plugin in the form of graphical function blocks and serve as a configuration and execution interface in the program editor.

The return values of the instructions are generally stored in numeric registers (hereinafter referred to as „registers“) of the robot controller. These registers can then be evaluated in the robot program, for example for status monitoring or process control. The register index can be set individually for each command. Alternatively, a default value for the register index can be defined via the plugin configuration.

The CRX plugin internally utilizes the functions of the Core plugin, which handles the actual communication with the GRIPLINK Controller. The graphical instructions thus serve as a user-friendly interface for the commands implemented in the Core plugin. Applications can therefore be created both graphically via instructions and textually via function calls. Both variants access the same communication interface.

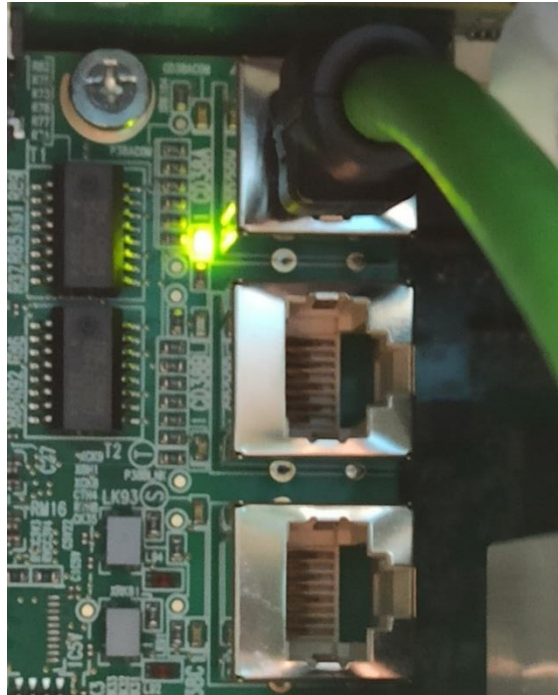
6.2 Hardware Commissioning



BE SURE TO DISCONNECT THE ROBOT CONTROLLER FROM THE POWER SUPPLY BEFORE OPENING IT!

Risk of injury and damage!

Connect the GRIPLINK Controller to the upper network port on your robot controller as shown in the following figure.

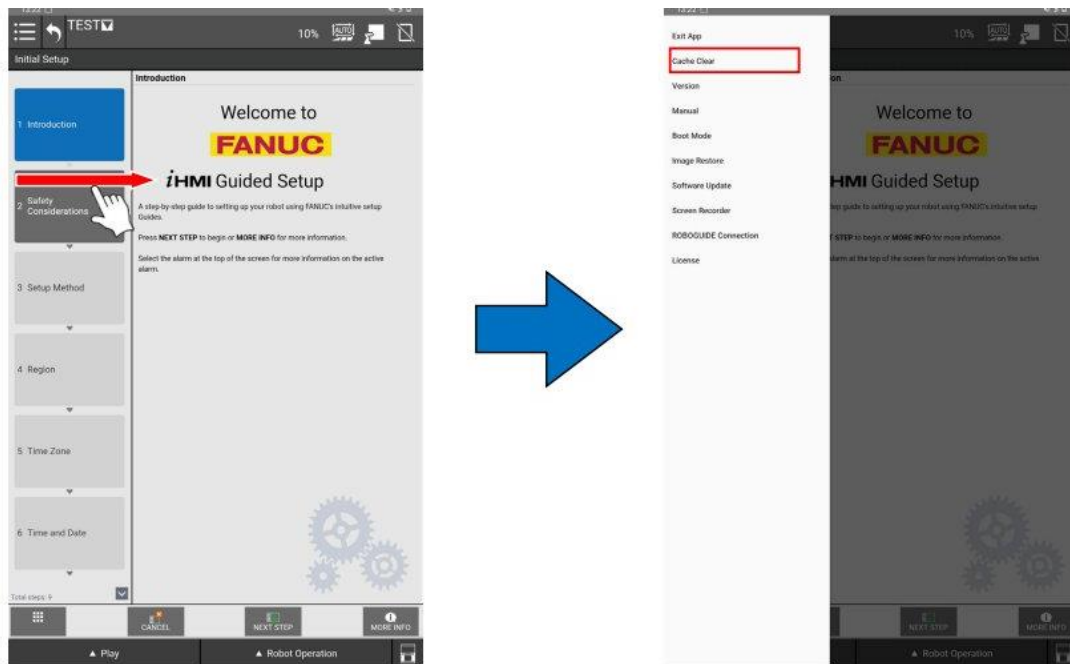


6.3 Preparing the Plugin



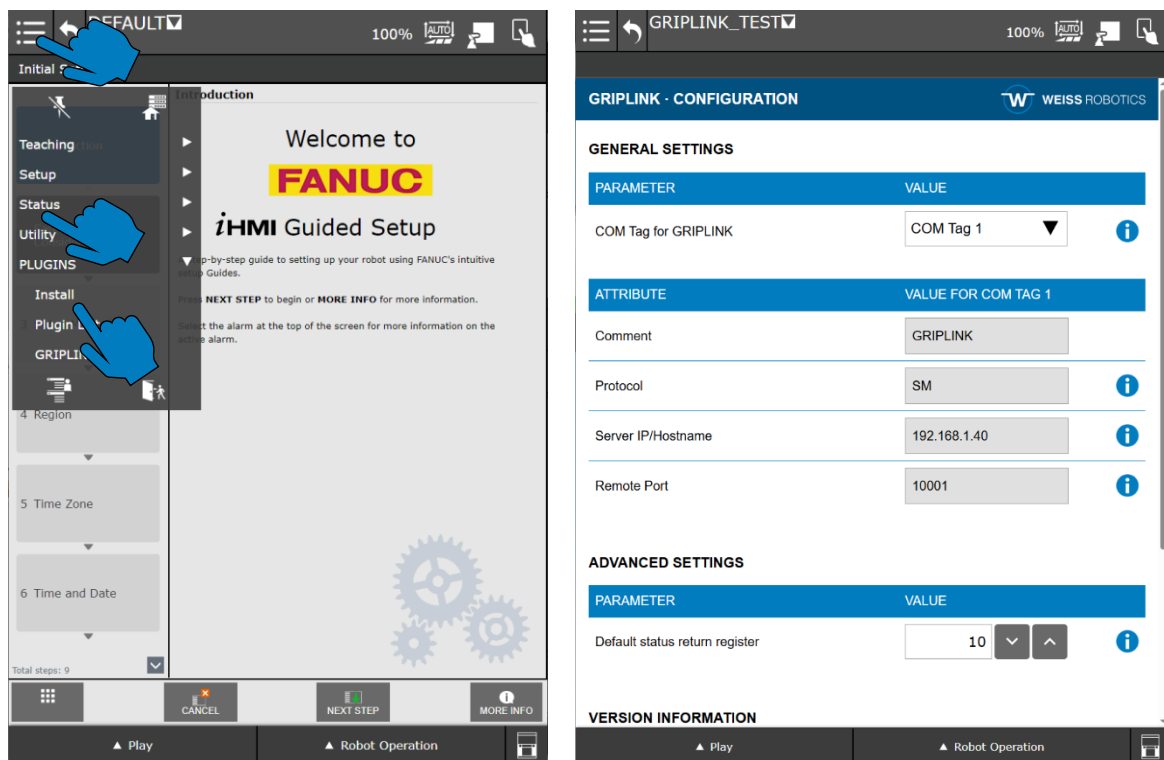
We recommend clearing the Tablet TP cache before setting up the plugin. This prevents display problems on the configuration page and with the graphical instructions that can be caused by cached data.

To clear the cache, slide the left edge of the screen towards the right to open the menu and select „Cache Clear“. Confirm the dialog „Clear the cache and exit the Tablet TP“ with „OK“. The Tablet teach pendant then closes.



A detailed description of the cache clear procedure can be found in the FANUC „Tablet UI Operator’s Manual“ (document no. B-84274EN/05), section 2.7 „When Tablet TP connect to another controller“.

Open the configuration page by selecting the „Plugins“ tab in the menu and tapping the „GRIPLINK“ entry.

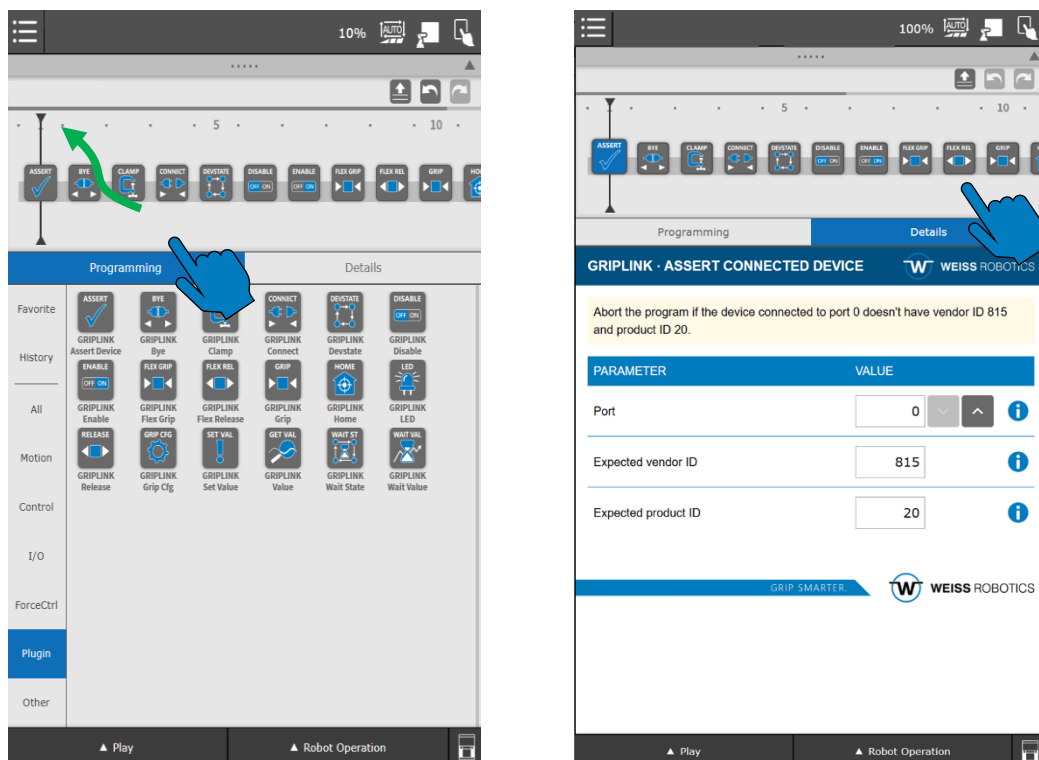


Use the „COM Tag“ dropdown field to set the previously configured COM tag. If a COM tag is selected that is configured incorrectly, an error message will be displayed.

6.4 Creating Applications with Instructions

Instructions are predefined function blocks that encapsulate individual GRIPLINK commands in graphical form. They enable the configuration and execution of device functions without direct textual programming. Each instruction corresponds to a command of the Core plugin and makes its parameters available in a structured user interface. This allows even complex processes to be created quickly and with minimal errors.

The instructions of the GRIPLINK plugin can be found under the „I/O“ tab in the program editor. They can be easily dragged and dropped into the program.



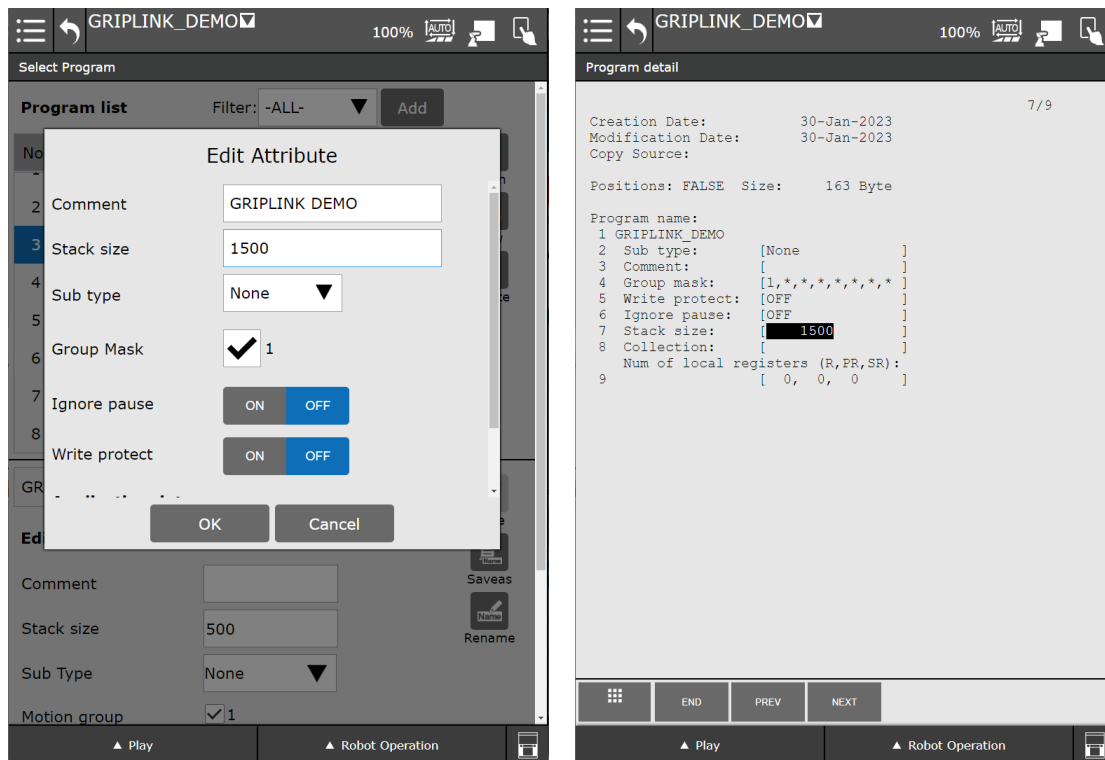
When you select an instruction, the „Details“ tab takes you to a detailed view of the command, where you can configure specific settings depending on the command. In the detailed view of an instruction, all available parameters are clearly displayed. Parameterization is performed via input fields, selection options, and register assignments.

The „i“ button displays context-sensitive descriptions for the individual parameters, providing additional information on their use and meaning.

6.5 Stack Size

Since GRIPLINK instructions perform additional function calls and data processing internally, a sufficiently large stack size should be configured for corresponding robot programs. In particular, with extensive programs or the combination of multiple instructions, an insufficient stack size can lead to runtime errors.

If a „Stack Overflow“ error occurs during program execution, the program's stack size must be increased accordingly in the program properties.



6.6 Recommended Program Structure

Programs that use GRIPLINK instructions should generally have the following structure:

1. Establish a connection using the GRIPLINK Connect instruction
2. Check or initialize devices (e.g., Assert Device, Enable, Home)
3. Execute device functions (e.g., Grip, Release, Value, Wait Value)
4. Optionally, disconnect using GRIPLINK Bye if the physical connection must be terminated

In most applications, the connection remains active for the entire duration of the program.

6.7 Overview of Available Commands

The instructions provide a graphical configuration and execution interface for the commands of the Core plugin. They encapsulate the underlying function calls and allow their parameters to be configured in a structured and clear manner via the tablet teach pendant.

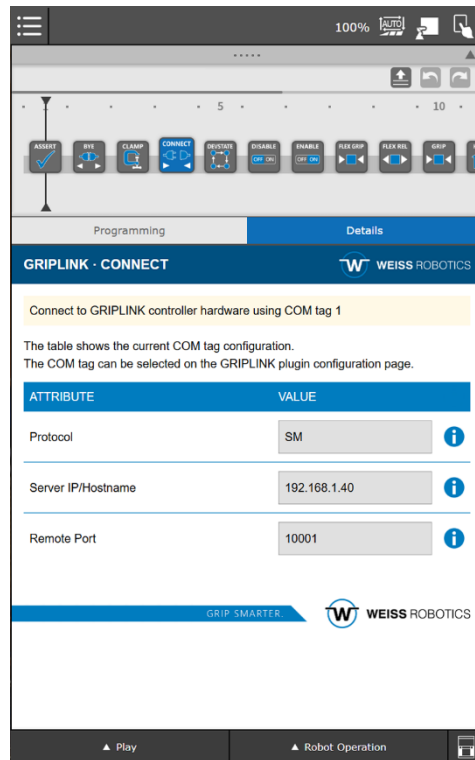
Each instruction corresponds to a specific command of the Core plugin and fully maps its functionality. Instead of manually entering function parameters, these are provided via input fields, selection options, and predefined parameter structures. This simplifies the creation of robot programs and reduces the likelihood of errors.

Internally, the parameters set in the instruction are passed to the interface program and executed there as the corresponding Core command. Thus, the underlying communication logic remains unchanged, while the user is provided with a convenient and intuitive user interface. The following general notes apply:

- For all port-related commands, „Port“ is an integer port index whose value range depends on the connected GRIPLINK Controller.
- Register indices are displayed in the UI as R[n] and are consistently in the range R[1] to R[200].
- Several commands use a return register for the device status. If no valid register index is specified in an instruction, the default value defined on the plugin configuration page is used.
- Parameters with the const/reg switch can either be stored as a constant in the command or read from a numeric register at runtime.
- The internal format of such parameters is either a numeric value, for example 12.5, or a register reference in the format R[n], for example R[42].
- Return values of the device status type are status codes of the connected device; see Appendix A.

6.7.1 Establishing a Connection – CONNECT

Establishes the connection between the robot controller and the GRIPLINK Controller based on the global plugin configuration.



Parameters

No editable instruction parameters. CONNECT implicitly uses the COM tag selected on the GRIPLINK configuration page. An error is displayed if the communication configuration is invalid.

Description

None.

Script mode syntax

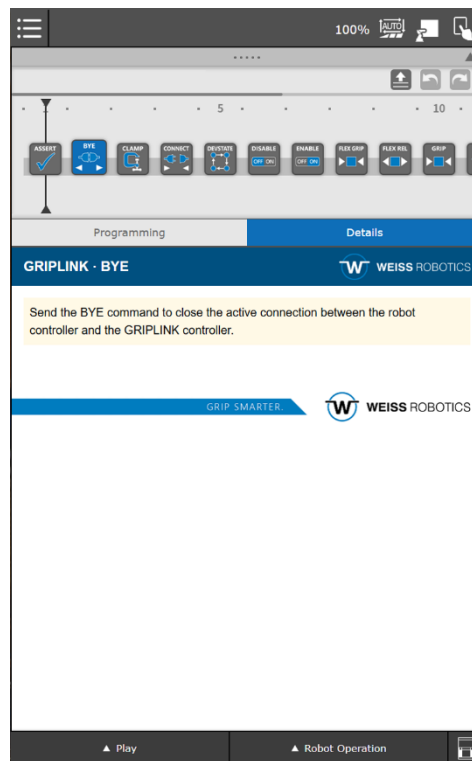
CALL IPL_WR_GRIPLINK_CONNECT

6.7.2 Disconnect – BYE

Sends the BYE command to cleanly close the active connection between the FANUC controller and the GRIPLINK Controller.



Use this instruction within a robot program only if the physical data connection to the GRIPLINK Controller is to be interrupted, for example during a tool change.



Parameters

None

Return Values

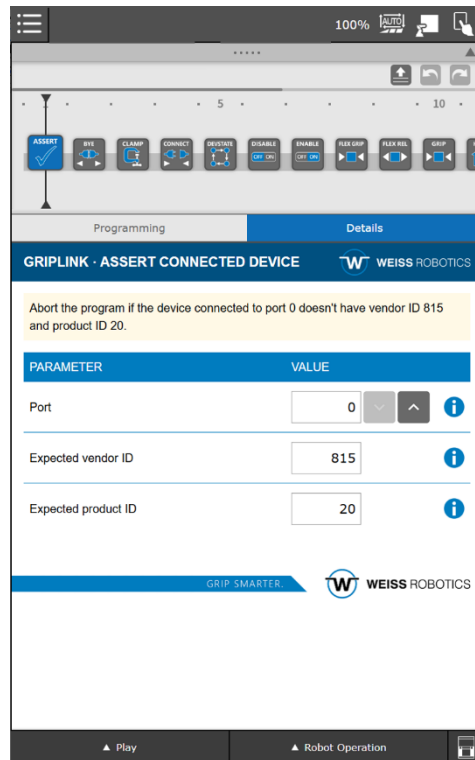
None

Script mode syntax

```
CALL IPL_WR_GRIPLINK_BYE
```

6.7.3 Check connected Devices – ASSERTDEV

Checks whether the expected device is connected to the specified port. The TP program is terminated if the vendor ID or product ID does not match.



Parameters

Port	Port index (0 to 31)
Expected vendor ID	Expected vendor ID from the IODD file.
Expected product ID	Expected product ID from the IODD file.

Return values

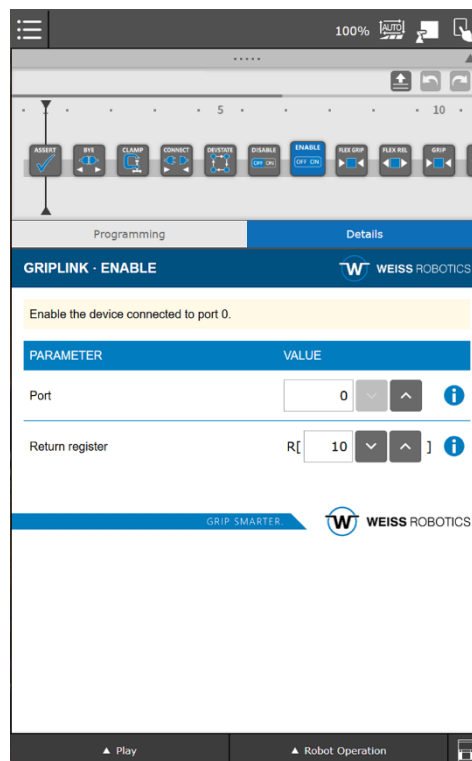
If there is a discrepancy, the program terminates instead of returning a normal result.

Script mode syntax

```
CALL IPL_WR_GRIPLINK_ASSERTDEV(<Port>,<VID>,<PID>)
```

6.7.4 Enable Device – ENABLE

Enables the device connected to the selected port.



Parameters

Port	Port index (0 to 31)
Return register	Destination register for the resulting device state

Return values

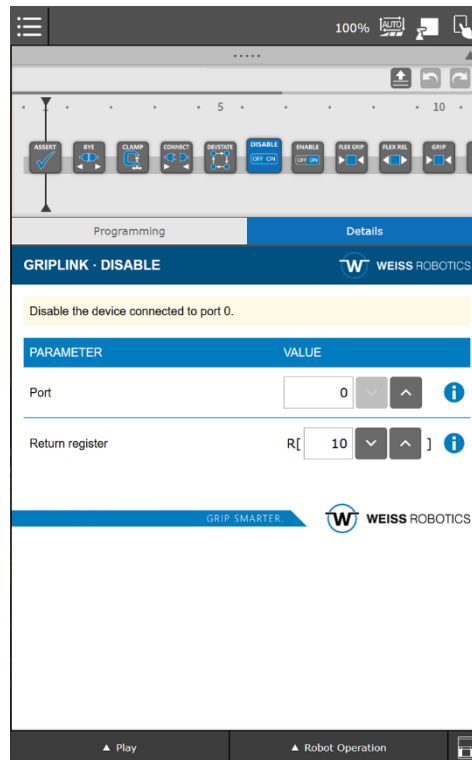
Writes the device state to R[n].

Script mode syntax

```
CALL IPL_WR_GRIPLINK_ENABLE(<Port>,<Register Index>)
```

6.7.5 Disable Device – DISABLE

Disables the device connected to the selected port.



Parameters

Port	Port index (0 to 31)
Return register	Destination register for the resulting device state

Return values

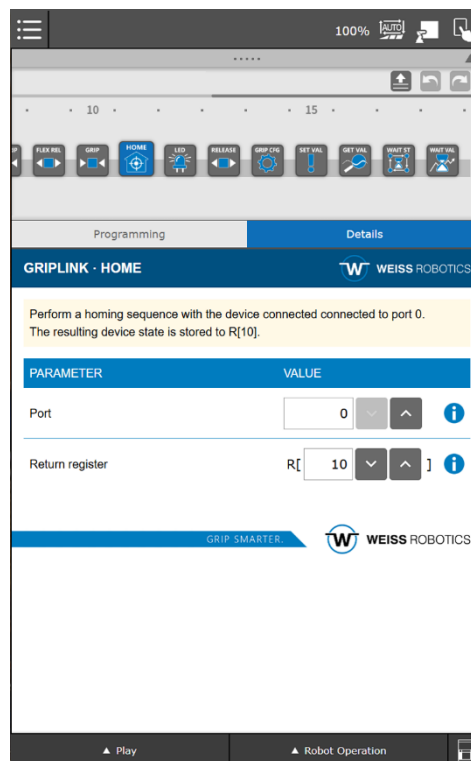
Writes the device state to R[n].

Script mode syntax

```
CALL IPL_WR_GRIPLINK_DISABLE(<Port>,<Register Index>)
```

6.7.6 Reference – HOME

Executes a homing sequence for the device.



Parameters

Port	Port index (0 to 31)
Return register	Destination register for the device status after homing.

Return values

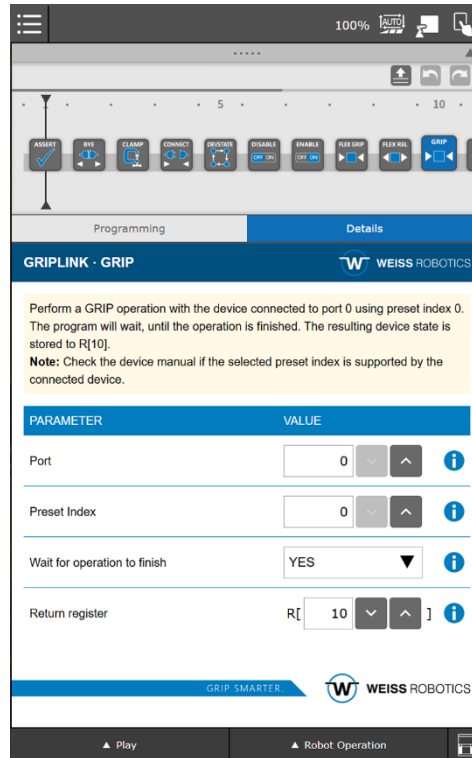
Writes the resulting device status to R[n].

Script mode syntax

```
CALL IPL_WR_GRIPLINK_HOME(<Port>,<Register Index>)
```

6.7.7 Grip Workpiece via Preset – GRIP

Performs a GRIP operation with a specified preset. Optionally, the program can wait for the operation to complete before proceeding.



Parameters

Port:	Port index (0 to 31)
Preset Index:	Integer 0 to 9999. Preset for the gripping configuration.
Wait for operation...:	Select NO or YES. NO: Starts the operation and returns immediately. YES: Waits until the end of the operation.
Return register:	Relevant only for blocking mode; destination register for the resulting device state.

Return values

If Wait = YES, the resulting device state is written to R[n].

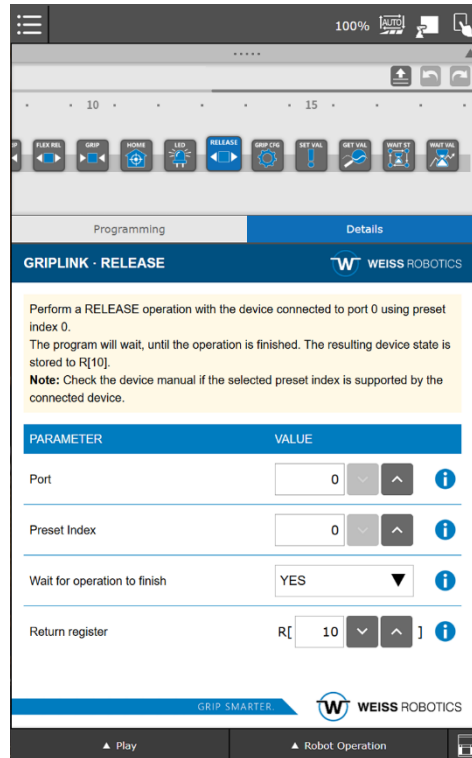
If Wait = NO, there is no synchronously evaluated status return by the instruction itself.

Script mode syntax

```
CALL IPL_WR_GRIPLINK_GRIP(<Port>,<Preset Index>,  
    <do block: 0=no,1=yes>,<Register Index>)
```

6.7.8 Release Workpiece via Preset – RELEASE

Performs a RELEASE operation using a specified preset. Optionally, the system can wait for the operation to complete before proceeding.



Parameters

Port:	Port index (0 to 31)
Preset Index:	Integer 0 to 9999. Preset of the release configuration.
Wait for operation...:	Specifies whether to wait for the operation to complete. NO: Starts the operation and returns immediately. YES: Waits until the operation is complete.
Return register:	Relevant only for blocking mode; destination register for the resulting device state.

Return values

If Wait = YES, the resulting device state is written to R[n].

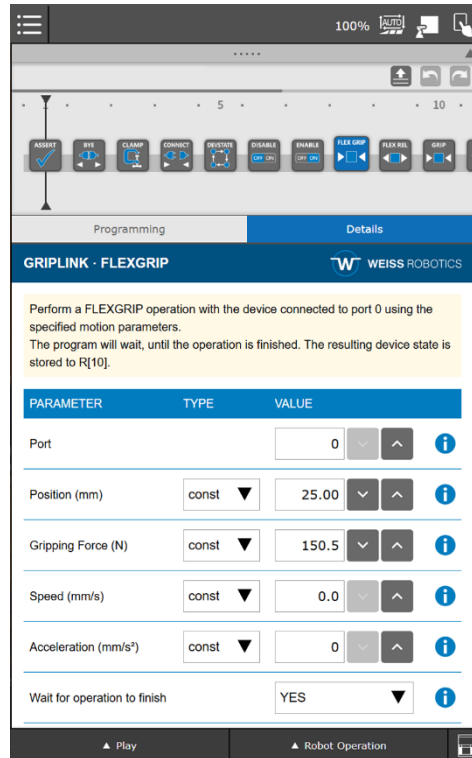
If Wait = NO, there is no synchronously evaluated status return by the instruction itself.

Script mode syntax

```
CALL IPL_WR_GRIPLINK_RELEASE(<Port>,<Preset Index>,  
    <do block: 0=no,1=yes>,<Register Index>)
```

6.7.9 Grip workpiece via Parameters – FLEXGRIP

Starts a gripping motion with freely configurable motion values. The operation can be executed non-blocking or blocking.



Parameter

Port:	Port Index (0 to 31)
Position (mm):	Position value of the no-part limit in mm at which the gripper switches to the NO PART state during gripping. Can be set either as a constant value (select „const“) or at runtime via a register (select „reg“).
Gripping Force (N):	Gripping force in Newtons. Can be set either as a constant value (select „const“) or at runtime via a register (select „reg“).
Speed (mm/s):	Finger speed in mm/s. Can be set either as a constant value (select „const“) or at runtime via a register (select „reg“). Speed = 0 uses the device's default setting.
Acceleration (mm/s²):	Finger acceleration while gripping in mm/s². Can be set either as a constant value (select „const“) or at runtime via a register (select „reg“). Acceleration = 0 uses the device's default setting.
Wait for operation...:	Specifies whether to wait for the operation to complete. NO: Starts the operation and returns immediately. YES: Waits until the operation is complete.

Return register: Only relevant for blocking mode: Specifies the return register for the device status.

Return values

If Wait = YES, the resulting device state is written to R[n].

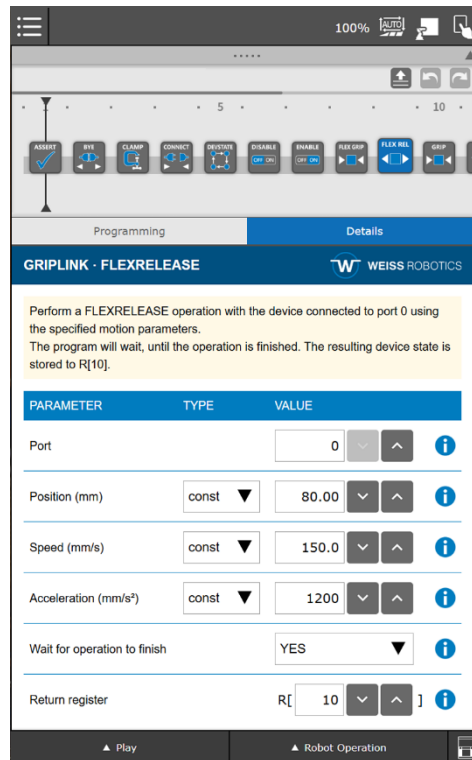
If Wait = NO, the motion starts and the instruction returns immediately.

Script mode syntax

```
CALL IPL_WR_GRIPLINK_FLEXGRIP(<Port>,<Position>,<Gripping Force>,  
    <Speed>,<Acceleration>,<do block: 0=no,1=yes>,<Register Index>)
```

6.7.10 Release workpiece via Parameter – FLEXRELEASE

Releases a previously gripped workpiece with freely configurable motion values. The operation can also be used to pre-position the gripper fingers. It can be executed in both non-blocking and blocking modes.



Parameters

Port:	Port index (0 to 31)
Position (mm):	Release position (release limit) in mm. When the fingers reach this position, the gripper switches to the RELEASED state. Can be set either as a constant value (select „const“) or at runtime via a register (select „reg“).
Speed (mm/s):	Finger speed in mm/s. Can be set either as a constant value (select „const“) or at runtime via a register (select „reg“). Speed = 0 uses the device's default setting.
Acceleration (mm/s ²):	Finger acceleration during release in mm/s ² . Can be set either as a constant value (select „const“) or at runtime via a register (select „reg“). Acceleration = 0 uses the device's default setting.
Wait for operation...:	Specifies whether to wait for the operation to complete. NO: Starts the operation and returns immediately. YES: Waits until the operation is complete.
Return register:	Only relevant for blocking mode: Specifies the return register for the device status.

Return values

If Wait = YES, the resulting device state is written to R[n].

If Wait = NO, the motion starts and the instruction returns immediately.

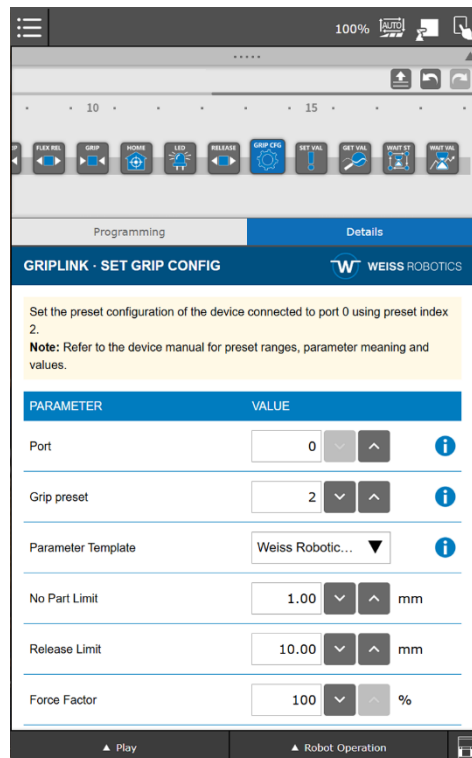
Script mode syntax

```
CALL IPL_WR_GRIPLINK_FLEXRELEASE(<Port>,<Position>,<Speed>,  
    <Acceleration>,<do block: 0=no,1=yes>,<Register Index>)
```

6.7.11 Configuring a Grip Preset – SGRIPCFG

On the detail page of the „SGRIPCFG“ instruction, the port of the desired gripper and the index of the preset can be selected.

The Template parameter can be used to display the meaning of the individual parameters. For certain devices, irrelevant parameters are hidden. Depending on the template, permissible value ranges are also checked.



Parameter

Port:	Port index (0 to 31)
Grip preset:	Preset index to be overwritten.
Parameter Template:	Selection of the interpretation scheme

Return values

None

Script mode syntax

```
CALL IPL_WR_GRIPLINK_SGRIPCFG(<Port>,<Preset Index>,<Template>,  
<Param 1>,<Param 2>,<Param 3>,<Param 4>,<Param 5>,<Param 6>,<Param 7>,<Param 8>)
```



The „Template“ parameter is required for decoding the parameters in the graphical view. To switch between textual and graphical display, it must have one of the following values:

0: Show generic UI template

- 1: Show Weiss Robotics UI template (default for all Weiss Robotics IO-Link grippers)
- 2: Show Weiss Robotics UI template (default for Weiss Robotics WPG grippers)
- 3: Show generic vacuum suction cup UI template

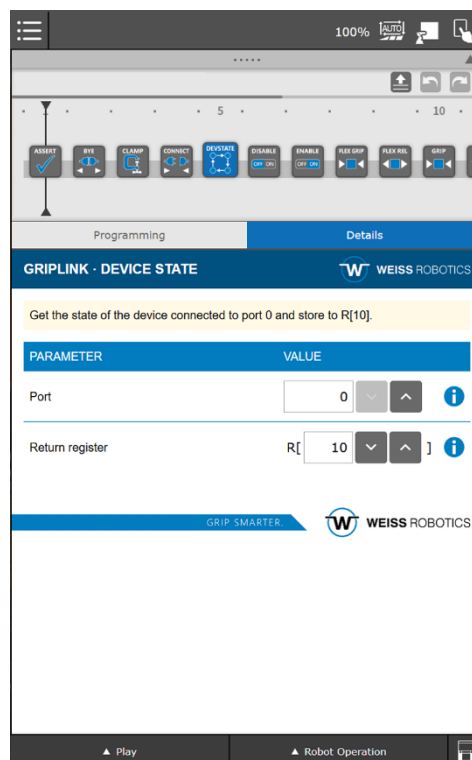


Note that the number of configurable grips depends on the device type.

6.7.12 Reading the Device State – DEVSTATE

To read the device status of a connected device, the „DEVSTATE“ instruction can be used. This queries the current status of the device at the selected port and stores it in the specified register.

Each device always has a unique device state, which can change due to actions such as gripping, releasing, activating, or deactivating. This command allows the robot program to react directly to changes in a device's state. For example, it can be determined whether a gripper has successfully gripped a component or whether no component was detected. A complete list of possible device states can be found in Appendix A .



Parameters

Port:	Port index (0 to 31)
Return register:	Specifies the return register for the device state.

Return values

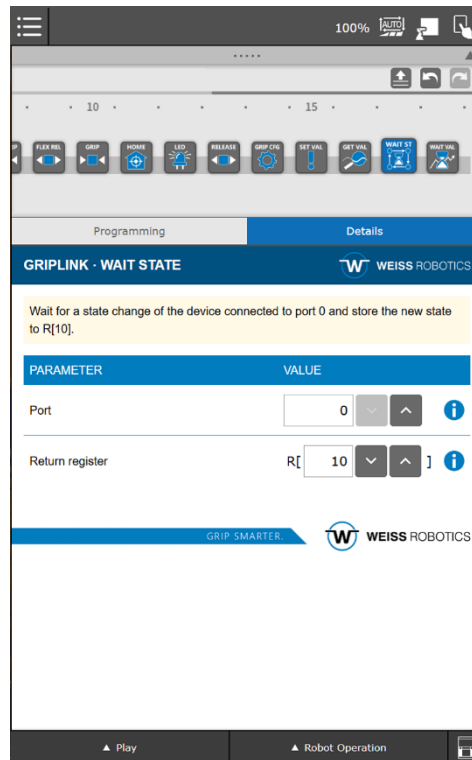
Writes the currently read device status to R[n].

Script mode syntax

```
CALL IPL_WR_GRIPLINK_DEVSTATE(<Port>,<Register Index>)
```

6.7.13 Wait for a Device State Change – WAITSTATE

Blocks until the device state changes at the specified port. This can be used, for example, to check whether an asynchronously executed release movement (RELEASE with the option „Wait for operation to finish“=NO) has been completed.



Parameters

Port:	Port index (0 to 31)
Return register:	The new device state is written to this register as soon as the state changes.

Return values

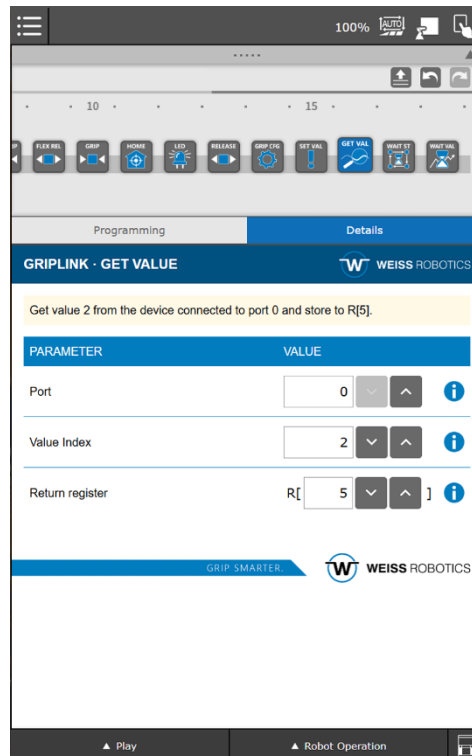
Writes the new device status to R[n].

6.7.14 Query Device Value – VALUE

To use a measurement value from a connected device in the robot program, you can use the „VALUE“ instruction. This instruction allows you to read measurement or process values from a device, store them in a register, and use them later in the program.



The index used has a device-specific meaning. The values available via the individual indices can be found in the documentation for the respective device.



Parameter

Port:	Port index (0 to 31)
Value Index:	Index of the value to be queried. Value Index = 0 always returns the primary value of the device. Other indexes are device-dependent.
Return register:	Destination register for the read value.

Return values

Writes the read value to R[n].

Script mode syntax

CALL IPL_WR_GRIPLINK_VALUE(<Port>,<Value Index>,<Register Index>)

6.7.15 Set Device Value – SETVALUE

The „SETVALUE“ instruction can be used to transfer a value to a connected device. The value to be written can either be entered directly as a constant floating-point number or defined indirectly via the contents of a register.

The index used determines which device value is written. The meaning of the individual indices is device-specific and must be taken from the documentation for the respective device.



Parameters

Port:	Port index (0 to 31)
Value Index:	Device-dependent value index.
Value:	Value to be set. Can be set either as a constant value (select „const“) or at runtime via a register (select „reg“).

Return Values

None.

Script mode syntax

```
CALL IPL_WR_GRIPLINK_SETVALUE(<Port>,<Value Index>,<Value>)
```

6.7.16 Wait for a Device Value – WAITVALUE

The „WAITVALUE“ instruction allows the robot program to wait until a specific measurement or process value from a connected device has been reached.

The instruction continuously monitors a selected device value and does not resume program execution until the measured value falls within a defined tolerance window around the target value. This allows the robot program to reliably respond to conditions reported by external devices or sensors.

Typical use cases include:

- Waiting until a sensor reaches a specific measured value
- Checking whether a process value is within an acceptable range
- Synchronizing the robot program with an external device

If the desired value is not reached within the specified timeout period, the instruction terminates execution with a timeout error, causing the robot program to abort.

Parameters

Port:	Port index (0 to 31)
Value Index:	Index of the value to be monitored (the permissible value range depends on the device).
Threshold Value:	Target value in the unit of the measured value
Window Size:	Width of the tolerance window symmetrical to the target value
Timeout (ms):	Timeout value in milliseconds

Return values

None. If the time limit is exceeded, a timeout error is triggered and the TP program is terminated.

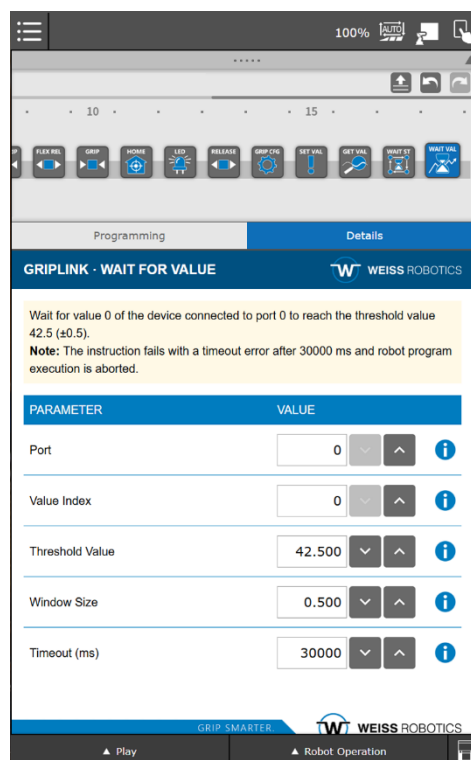
6.7.17 Control Force Retention – CLAMP

Using the CLAMP command, gripper modules with integrated gripping force retention can de-energize the drive in the HOLDING state to reduce the gripper module's power consumption. The workpiece continues to be held securely.

If CLAMP is activated, the drive switches off in the HOLDING state. If CLAMP is deactivated, gripper force control remains active in the HOLDING state, and the gripper module's controller continuously adjusts the gripper force.



The CLAMP command is not available on all gripping modules.



Parameter

Port:	Port index (0 to 31)
State:	Controls mechanical grip force maintenance ON: Gripping force maintenance is activated in the HOLDING state and the motor is shut off. OFF: The motor remains controlled in the HOLDING state.

Return values

None.

Script mode syntax

```
CALL IPL_WR_GRIPLINK_CLAMP(<Port>,<State>)
```

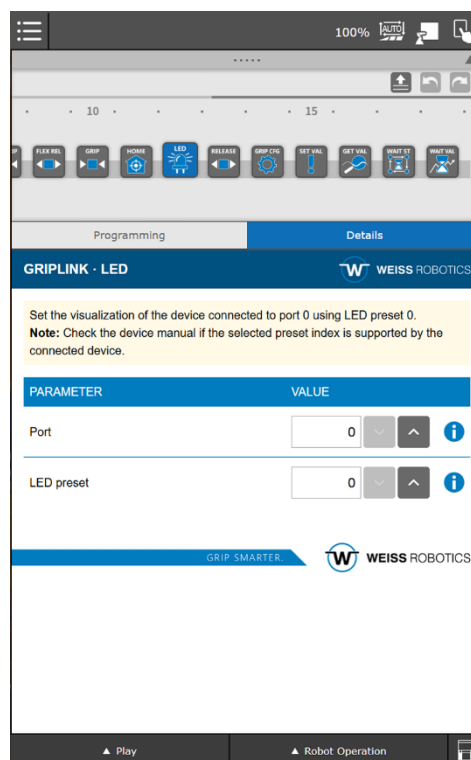
6.7.18 Controlling the Status LED – LED

The „LED“ instruction can be used to control the LED display of a connected device. This function allows you to activate predefined LED presets, for example, to visually indicate the current operating status of a device.

The instruction sends the selected LED preset to the device connected to the specified port. The device then displays its LED indicator according to the selected preset.



The availability of this function depends on the specific device connected. Not all devices support external control of the LED display. Information on whether and to what extent the LED display can be controlled can be found in the Operating Manual for the respective device.



Parameter

Port:	Port index (0 to 31)
LED preset:	Index for selecting the LED preset. The meaning of the preset depends on the device.

Return values

None.

Script mode syntax

```
CALL IPL_WR_GRIPLINK_LED(<Port>,<Preset Index>)
```

7 Troubleshooting

The GRIPLINK plugin outputs error messages during operation. Important messages and solutions are explained below.

7.1 TELNET Output

To analyze errors during operation, you can enable the TELNET output. Navigate to the corresponding setup menu via SETUP – Host Comm – 2 TELNET. Click in the „Password“ field in the „CONS“ line and enter a password (e.g., 1111).



Note for RoboGuide: For R-30iB version controllers, the password must be entered using the keypad of the emulated legacy teach pendant. Use the numeric keypad and the „ENTER“ keys to enter and confirm the password.

Username	Access	Password	Timer
TP	OUTPUT	1111	0
KCL	INPUT	1111	0
CONS	INPUT	1111	0

You can now connect to the robot controller via TELNET using a terminal program (e.g., PuTTY). This also works with RoboGuide.

7.2 Error Messages

7.2.1 Installation Error Message „MEMO-094 Parameters are different“

```
2 MEMO-094 Parameters are different
3 R E S E T
4 SYST-045 TP enabled in AUTO mode
5 R E S E T
```

Possible cause

Updating from an older to the latest plugin version

Solution

- Delete all program and variable files with the prefixes „GL_“ and „IPL_WR_GL_“, as well as the files named „GRIPLINK“, from the robot controller (see figures below)
- Reinstall the plugin

```
MD:\*. * 128/438
127 GIGELOG DG
128 GL_CMD PC
129 GL_COMM PC
130 GL_COMMON PC
131 GL_DEBUG PC
132 GRIPLINK PC
133 GRIPLINK VA
134 GRIPLINK VR
```

```
MD:\*. * 157/438
156 IPL_FANUC_SYS_ERPOST VR
157 IPL_WR_GRIPLINK_ASSERTDEV LS
158 IPL_WR_GRIPLINK_ASSERTDEV TP
159 IPL_WR_GRIPLINK_BYE LS
160 IPL_WR_GRIPLINK_BYE TP
161 IPL_WR_GRIPLINK_CFG PC
162 IPL_WR_GRIPLINK_CFG VA
163 IPL_WR_GRIPLINK_CFG VR
164 IPL_WR_GRIPLINK_CLAMP LS
165 IPL_WR_GRIPLINK_CLAMP TP
166 IPL_WR_GRIPLINK_CONNECT LS
167 IPL_WR_GRIPLINK_CONNECT TP
168 IPL_WR_GRIPLINK_CRXINIT LS
169 IPL_WR_GRIPLINK_CRXINIT TP
170 IPL_WR_GRIPLINK_DEVSTATE LS
171 IPL_WR_GRIPLINK_DEVSTATE TP
172 IPL_WR_GRIPLINK_DISABLE LS
173 IPL_WR_GRIPLINK_DISABLE TP
```

The figure does not show all programs!

7.2.2 System Error Message „EtherNet Adapter Error (1)“

Possible cause

By default, an Ethernet adapter is configured and must be deactivated.

Solution

- Navigate to the „I/O“ menu → „EtherNet/IP“
- In the first line (Connection 1), set the ENABLED field from „True“ to „False“
- Use the „Config“ softkey to switch to the configuration view

- Set the data size for „Input size“ and „Output size“ to 0
If necessary, you may need to activate the Legacy Pendant using the button in the lower right corner and enter the number using its numeric keypad
- Restart the robot controller

EtherNet/IP List(Rack 89)					
Description	TYP	Enable	Status	Slot	
Connection1	ADP	TRUE	ONLINE	1	

```

I/O EtherNet/IP

Adapter configuration :

Description :      Connection1
Input size (words) : 0
Output size(words) : 4
Alarm Severity : WARN

Scanner IP : *****
API O=>T :      0
API T=>O :      0

Old Value: 4

```

7.2.3 Message: „Could send command [command]: not connected“

The command [command] could not be sent.

Possible cause

A GRIPLINK command was executed without first calling GRIPLINK Connect, or the program pointer is not on the first instruction.

Solution

- Insert the GRIPLINK Connect program node (see sections 5.1 and 6.7.1)
- Move the program pointer to the first instruction

7.2.4 Message „GRIPLINK: Cmd [command] failed: E_NOT_OPEN“

The command [command] could not be sent because the interface is not open.

Possible cause

A GRIPLINK command was executed without first calling GRIPLINK Connect.

Solution

Install the GRIPLINK Connect program node (see sections 5.1 and 6.7.1)

7.2.5 Message „GRIPLINK Version not supported“

Possible cause

Outdated firmware version on the GRIPLINK Controller

Solution

Update all GRIPLINK Controllers in use to the firmware version listed in the system requirements (see section 1.3).

7.2.6 Message: „GL_COMM 190 : undefined built-in“

Possible cause

The KAREL option is not enabled

Solution

Check whether the KAREL option has been enabled by navigating to the HTTP settings in the „Host Comm“ section of the setup. Here, a „U“ must be entered in the left column of the „KAREL:*“ row. If not, press the „UNLOCK“ shortcut key.

HTTP Setup		
PROTECTED RESOURCES		
Name	Pwrd	Resource
U	*****	iPendant
U	*****	KAREL:*
U	*****	KCL:*
U	*****	VISION SETUP
A	*****	*****
A	*****	*****
A	*****	*****
A	*****	*****

The installation of the KAREL or User Socket Messaging option was not processed correctly. Make a complete backup of your robot. Contact FANUC Support to completely reset the robot's software, and install the latest software along with the necessary software options for GRIPLINK and your application.



Be sure to contact FANUC Support before resetting the robot and reinstalling the software!

7.2.7 Message: „GRIPLINK could not connect via SM slot Cx:“

Possible cause

The socket connection between the robot controller and the GRIPLINK Controller could not be established.

Solution

- Check the power supply to the GRIPLINK Controller
- In the „Host Comm“ setup, go to the settings for the selected client tag (Cx) (see chapter 4.5)
- Use the [ACTION] button to select the „UNDEFINE“ action
- Use the [ACTION] button to select the „DEFINE“ action
- Select the „START“ action using the [ACTION] button
- Restart the current program
- If necessary, disconnect the GRIPLINK from the power supply to reset it

7.2.8 Message „Listen failed:“

Possible cause

GRIPLINK Controller not active

Solution

Check the power supply to the GRIPLINK Controller.

7.2.9 Message „Device x not connected“

Possible cause

A GRIPLINK port has been selected to which no device is currently connected, or there is a hardware defect.

Solution

Check the ports and the connected devices

7.2.10 Message: „Stack Overflow“

Possible cause

The stack size for the current program is insufficient.

Solution

Increase the stack size in the program settings.

7.2.11 Message „GRIPLINK: Failed to listen on socket“

Possible cause

The socket connection between the robot controller and GRIPLINK has been interrupted.

Solution

- Check the network connection (network cable) between the robot controller and the GRIPLINK Controller. Also make sure it meets the requirements (Is the current limit on the power supply set correctly?)
- Restart the desired robot program. It must execute a „Connect“ node at the beginning!
- This message also appears if you intentionally want to disconnect the connection.

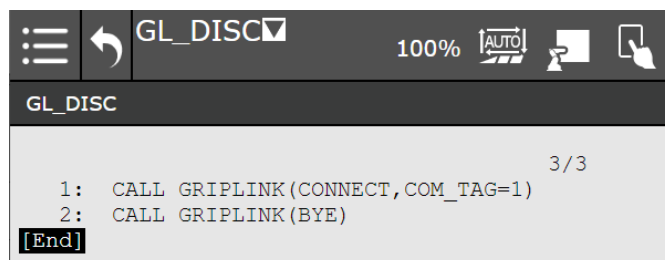
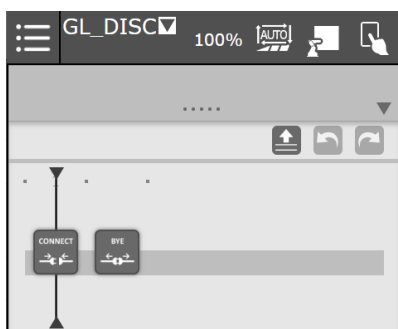
7.2.12 Connection between robot and GRIPLINK Controller cannot be disconnected

Possible cause

The socket connection between the robot controller and GRIPLINK is still active because the robot program was stopped midway.

Solution

- Briefly interrupt the power supply to the GRIPLINK Controller
- Create a separate robot program and insert only the CONNECT and BYE instructions. Select the program and run it. This will establish and then correctly terminate the connection to the GRIPLINK Controller.



Appendix A Device Status

Device status	Code	Description
NOT CONNECTED	0	Gripper module not connected
NOT INITIALIZED	1	Gripper module not initialized
IDLE	2	Ready for operation, not active
RELEASED	3	Workpiece released
NO PART	4	No workpiece found
HOLDING	5	Workpiece is held
OPERATING	6	Sensor ready
FAULT	7	Error condition

Appendix B Status Codes

If an error occurs during the execution of a GRIPLINK command, a corresponding status code is output. This describes the cause of the error and can be used for diagnostics in the robot program or during commissioning. The following table provides an overview of the possible status codes and their meanings.

Status code	Identifier	Description
0	E_SUCCESS	No error. Command successfully executed.
1	E_OVERRUN	Data overrun
2	E_RANGE_ERROR	Value out of range
3	E_NOT_AVAILABLE	Function or data not available
4	E_NOT_INITIALIZED	Device not initialized
5	E_TIMEOUT	Timeout
6	E_INSUFFICIENT_RESOURCES	Not enough memory available
7	E_CHECKSUM_ERROR	Checksum error
8	E_ACCESS_DENIED	Access denied
9	E_INVALID_HANDLE	Invalid handle
10	E_INVALID_PARAMETER	Invalid parameter
11	E_INDEX_OUT_OF_BOUNDS	Index out of bounds
12	E_IO_ERROR	Generic I/O error
13	E_READ_ERROR	Read error
14	E_WRITE_ERROR	Write error
15	E_NOT_FOUND	Resource not found
16	E_NOT_OPEN	File or device not open
17	E_EXISTS	Resource already exists
18	E_NO_COMM	Connection error
19	E_STATE_CONFLICT	Invalid state
20	E_NOT_SUPPORTED	Command or function not supported

21	E_INCONSISTENT_DATA	Data inconsistent
22	E_CMD_SYNTAX	Syntax error
23	E_CMD_UNKNOWN	Unknown command
24	E_CMD_ABORTED	Command aborted
25	E_CMD_FAILED	Command failed
26	E_AXIS_BLOCKED	Axis is blocked
27	E_PENDING	Action pending / Not yet finished

© 2026 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.

The technical data provided in this document is 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 in systems where a malfunction could result in personal injury.

