SCHUNK

Software manual

2D Grasping Kit

SCHUNK software module FANUC

Translation of original software manual

Hand in hand for tomorrow

Imprint

Technical changes:
We reserve the right to make alterations for the purpose of technical improvement.

Document number: 1621291

Version: 02.00 | 02/12/2025 | en

Dear Customer,

Thank you for trusting our products and our family-owned company, the leading technology supplier of robots and production machines.

Our team is always available to answer any questions on this product and other solutions. Ask us questions and challenge us. We will find a solution!

Best regards,

Your SCHUNK team

Customer Management
Tel. +49-7133-103-2503
Fax +49-7133-103-2189
cmg@de.schunk.com

Please read the operating manual in full and keep it close to the product.

Table of Contents

1 General

1.1 About this manual

This manual contains information about the SCHUNK software module for FANUC robots and how to use them.

The software is used for simple integration and control of the following products in FANUC robot applications:

- 2D Grasping Kit

Definition of terms "Product"

The term "product" replaces the product names listed above in this manual.

This manual describes the software environment for a FANUC robot.

NOTE: The illustrations in this manual are intended to provide a basic understanding and may deviate from the actual version.

In addition to these instructions, the documents listed under ▸ 1.7 [🗋 6] are applicable.

Validity

These instructions apply for the following software versions:

SVC Firmware Version	FANUC version
>=3.0.0	V9.40P/56 Option: R648 User Socket Msg

1.2 Target group

This manual is intended for robot integrators who have basic mechanical and electrical training skills and who are also familiar with elementary programming concepts.

Commissioning and troubleshooting may only be performed by qualified personnel with appropriate training.

The following knowledge is required:

- Basic knowledge of robotics
- Knowledge in handling FANUC robots

Electrical installations may only be carried out by a suitably trained electrician.

1.3 Symbol definition

The following symbols are used in this manual:

■ Prerequisite for an action

1. Action 1
2. Action 2

    ⇨ Intermediate results

⇨ Final results

▸ 1.3 [🗋 5]: chapter number and [page number] in hyperlinks

1.4 Abbreviations

SVC	SCHUNK Vision Controller (Industrial PC)
AI	Artificial Intelligence
TCP	Tool Center Point
CMOS	Control Memory Operating System (memory)

1.5 Brands

- FANUC is a registered trademark of FANUC CORPORATION (Japan).

1.6 Presentation of Warning Labels

To make risks clear, the following signal words and symbols are used for safety notes.

⚠ DANGER

Dangers for persons!

Non-observance will inevitably cause irreversible injury or death.

⚠ WARNING

Dangers for persons!

Non-observance can lead to irreversible injury and even death.

⚠ CAUTION

Dangers for persons!

Non-observance can cause minor injuries.

NOTICE

Material damage!

Information about avoiding material damage.

1.7 Applicable documents

- Commissioning instructions for 2D Grasping Kit *
- Operating manual for FANUC robots

The documents labeled with an asterisk (*) can be downloaded from **schunk.com/downloads**.

2 Functional description

The software module facilitates operation and creation of applications for SCHUNK products on a FANUC robot.

All necessary controls are installed via the software module. After the installation is complete, the programming elements are deployed within the graphical user interface (GUI). The GUI supports the entire configuration and parameterization of SCHUNK products as well as the necessary control and programming options.

The following functions are available in the software module and can be used in a robot program:

- **SVC_INIT**: Initializes the SVC and starts all necessary processes to enable all further commands to be used.
- **SVC_CONNECT**: Establishes a connection to the SVC.
- **SVC_ROBPREPOS**: Saves the robot configuration.
- **SVC_REPLY**: Configures the register in which the reply from the SVC is saved.
- **SVC_GRASP**: Starts object detection and receives the grasping pose for an object.
- **SVC_FEEDBACK**: Sends feedback to SVC with regard to the gripping action performed.
- **SVC_SETPROJECT**: Requests a project change based on the project ID.
- **SVC_SETTOOL**: Requests a tool change based on the tool ID.
- **SVC_GETOBJECTCOUNT**: Analyzes the scene; the number of all objects and the number of instances of a specified object class are determined.
- **SVC_GETSTATE**: Queries the status of the SVC.
- **SVC_SENDROBOTPOSE**: Transmits the current robot position to the SVC.
- **SVC_TIMEOUTSTATE**: Displays the status of the command timeout.
- **SVC_CONSTATE**: Sends feedback on the connection status.
- **SCV_DISCONNECT**: Ends the connection to the SVC.
- **View**: Starts the output of the image evaluated from the last gripping process.

Further information on the functions, ▶ 6 [🗋 15].

3 Connecting the product to the robot control system

Before connecting or commissioning the product, read the operating manual of the robot and observe the instructions in this manual!

⚠ WARNING

Risk of injury due to unexpected movements!

If the power supply is switched on or residual energy remains in the system, components can move unexpectedly and cause serious injuries.

- Before starting any work on the product: Switch off the power supply and secure against restarting.
- Make sure, that no residual energy remains in the system.

⚠ CAUTION

Risk of injury from electric shock due to contact with live parts!

- Follow the operating manual for the robot.
- Before starting any work on the product: Switch off the energy supply and secure against re-connection.

NOTE

Safety-relevant signals (e.g. emergency stop) must be wired externally, e.g. via safety relays, thus completely disconnecting the product from the power supply.

- Perform a risk assessment for the entire robotic application based on legal requirements to evaluate all safety-related aspects of the application.

- SVC (industrial PC) is connected to the robot via a network cable.
- IP address of the SVC is known (factory settings 192.168.1.76)
- IP address of the robot is in the same subnet (example 192.168.1.100)

- For information on connecting the product, see the commissioning instructions.

4 Installing the software module

NOTICE

Possible damage to product!

The product or the robot may get damaged if electrical cables are connected or disconnected during operation.

- Connect or disconnect electrical connections only when the device is switched off.

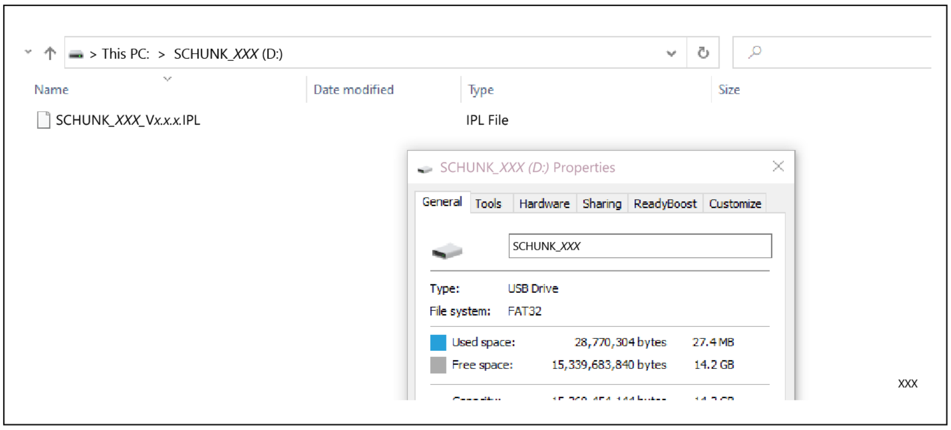
Preparing the installation

NOTE

A USB data carrier (type A) is required for installation.

The USB data carrier must comply with the following requirements:

- Formatted in FAT32 format



> This PC: > SCHUNK_XXX (D:)

Name	Date modified	Type	Size
SCHUNK_XXX_Vx.x.x.IPL		IPL File	

SCHUNK_XXX (D:) Properties

SCHUNK_XXX

Type: USB Drive

File system: FAT32

Used space:	28,770,304 bytes	27.4 MB
Free space:	15,339,683,840 bytes	14.2 GB

Installing

NOTE

To avoid malfunctions, SCHUNK recommends installing the current version of the software module.

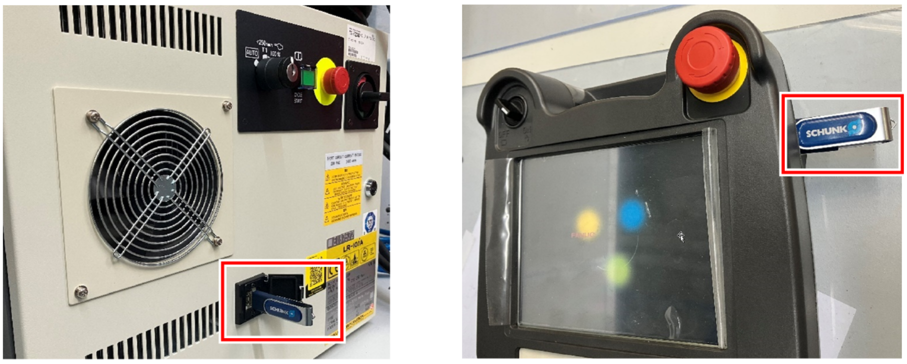
■ Product is mounted and connected to the robot control system.

1. Download the current version of the software module at **schunk.com/downloads** and copy it to the USB stick.

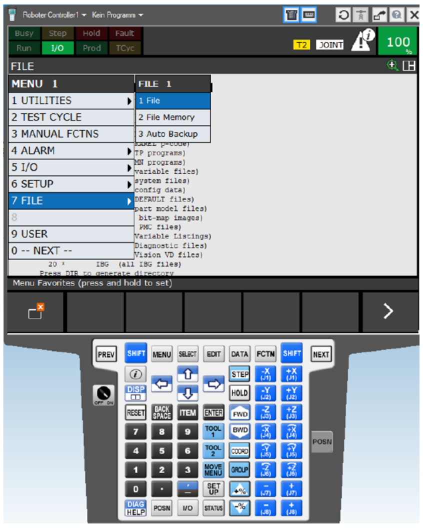
   ⇨ The FANUC Controller control software must be compatible with the version of the software module. Information on this can be found in the download area.

2. Extract the zip file directly to the USB stick.

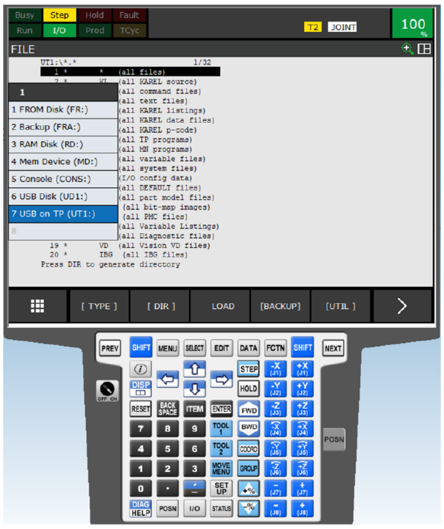
3. Connect the USB stick to the robot controller or the teach pendant.



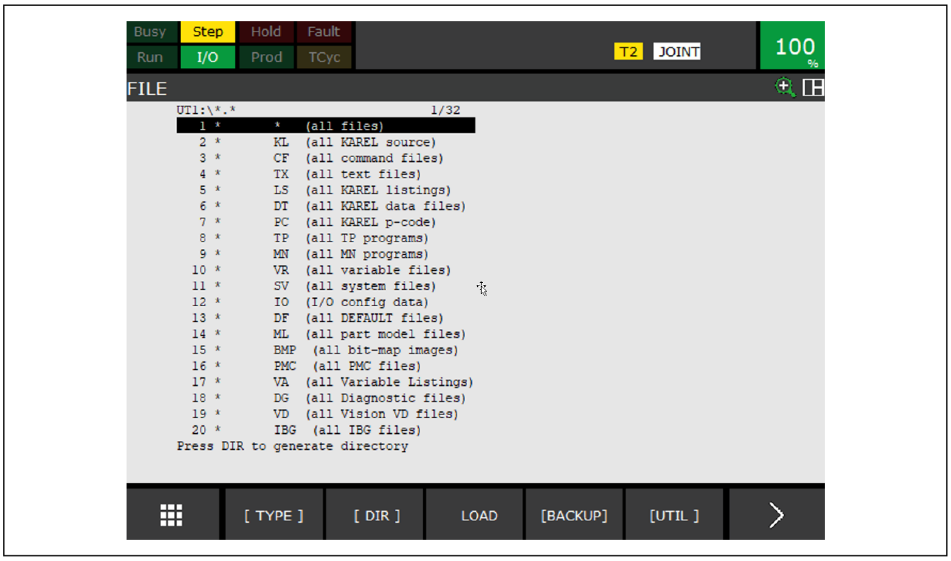
4. Select "Menu" > "File".



5. Select the appropriate USB drive via "UTIL" > "Set Device".

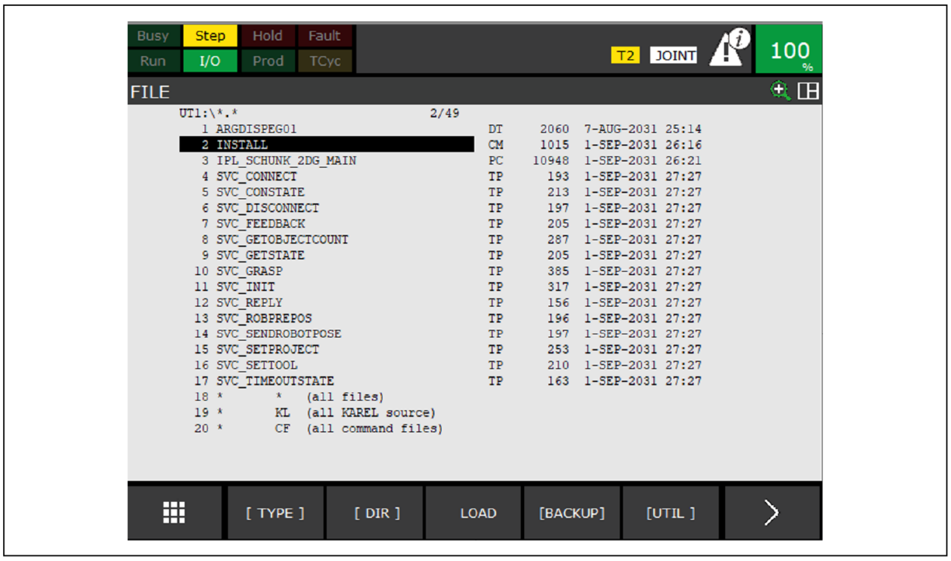


6. Select "All files".



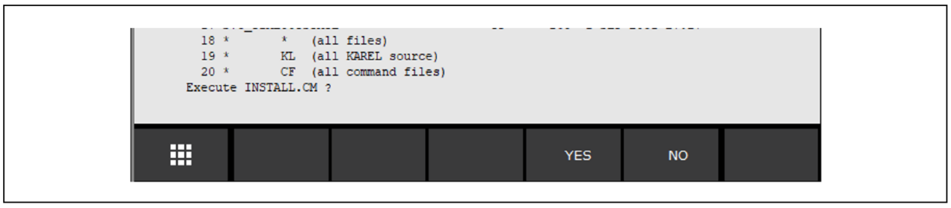
```
UT1:\*.*                        1/32
   1 *        *   (all files)
   2 *        KL  (all KAREL source)
   3 *        CF  (all command files)
   4 *        TX  (all text files)
   5 *        LS  (all KAREL listings)
   6 *        DT  (all KAREL data files)
   7 *        PC  (all KAREL p-code)
   8 *        TP  (all TP programs)
   9 *        MN  (all MN programs)
  10 *        VR  (all variable files)
  11 *        SV  (all system files)
  12 *        IO  (I/O config data)
  13 *        DF  (all DEFAULT files)
  14 *        ML  (all part model files)
  15 *        BMP  (all bit-map images)
  16 *        PMC  (all PMC files)
  17 *        VA  (all Variable Listings)
  18 *        DG  (all Diagnostic files)
  19 *        VD  (all Vision VD files)
  20 *        IBG  (all IBG files)
Press DIR to generate directory
```

7. Select "Install" and confirm by pressing Enter.



```
UT1:\*.*                             2/49
   1 ARGDISPEG01                            DT    2060  7-AUG-2031 25:14
   2 INSTALL                                CM    1015  1-SEP-2031 26:16
   3 IPL_SCHUNK_2DG_MAIN                    PC   10948  1-SEP-2031 26:21
   4 SVC_CONNECT                            TP     193  1-SEP-2031 27:27
   5 SVC_CONSTATE                           TP     213  1-SEP-2031 27:27
   6 SVC_DISCONNECT                         TP     197  1-SEP-2031 27:27
   7 SVC_FEEDBACK                           TP     205  1-SEP-2031 27:27
   8 SVC_GETOBJECTCOUNT                     TP     287  1-SEP-2031 27:27
   9 SVC_GETSTATE                           TP     205  1-SEP-2031 27:27
  10 SVC_GRASP                              TP     385  1-SEP-2031 27:27
  11 SVC_INIT                               TP     317  1-SEP-2031 27:27
  12 SVC_REPLY                              TP     156  1-SEP-2031 27:27
  13 SVC_ROBPREPOS                          TP     196  1-SEP-2031 27:27
  14 SVC_SENDROBOTPOSE                      TP     197  1-SEP-2031 27:27
  15 SVC_SETPROJECT                         TP     253  1-SEP-2031 27:27
  16 SVC_SETTOOL                            TP     210  1-SEP-2031 27:27
  17 SVC_TIMEOUTSTATE                       TP     163  1-SEP-2031 27:27
  18 *        *   (all files)
  19 *        KL  (all KAREL source)
  20 *        CF  (all command files)
```

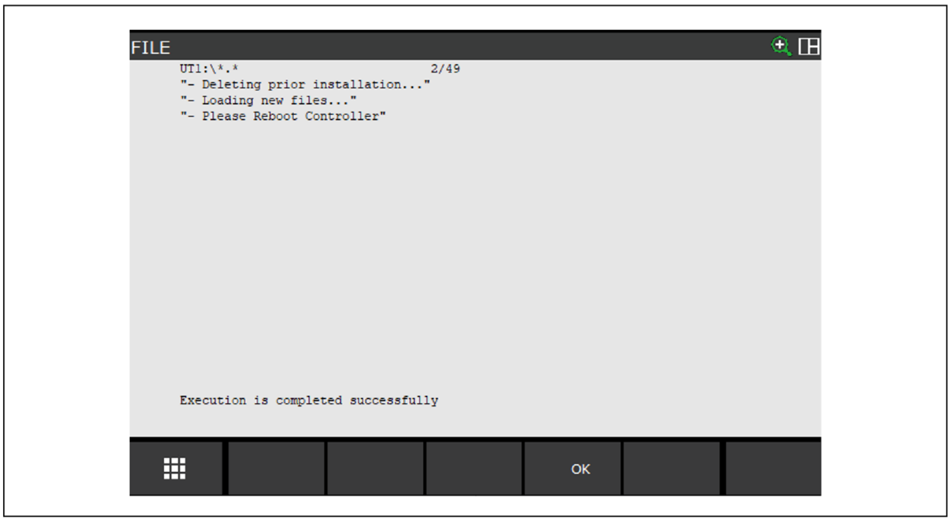
8. Confirm installation by pressing "YES".



```
  18 *        *   (all files)
  19 *        KL  (all KAREL source)
  20 *        CF  (all command files)
Execute INSTALL.CM ?
```

⇨ The installation is executed.

9. Restart the robot after successful installation.



```
UT1:\*.*                          2/49
"- Deleting prior installation..."
"- Loading new files..."
"- Please Reboot Controller"


Execution is completed successfully
```

5 Configuring the software module

⚠ WARNING

Risk of injury due to sudden movements!

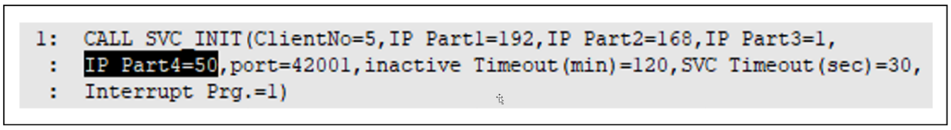
Components could move unexpectedly and result in serious injuries.

- During commissioning, observe all warnings displayed on the software interface.
- Keep a safe distance and wear suitable protective equipment.

■ Robot and SVC are switched on and connected with a network cable.

■ Robot and SVC are in the same network.

- Call up the "SVC_INIT" command in the robot program.



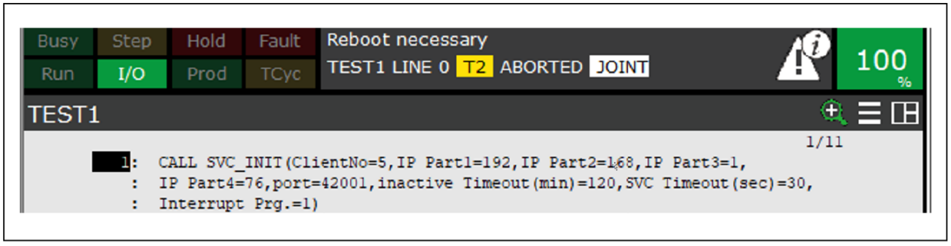
```
CALL SVC_INIT(ClientNo=5,IP Part1=192,IP Part2=168,IP Part3=1,
IP Part4=50,port=42001,inactive Timeout(min)=120,SVC Timeout(sec)=30,
Interrupt Prg.=1)
```

The following settings are possible in the menu item:

- Client no. of the FANUC controller
  Factory setting: 5
- Select IP address part 1
  Factory setting: 192
- Select IP address part 2
  Factory setting: 168
- Select IP address part 3
  Factory setting: 1
- Select IP address part 4
  Factory setting: 76
  IP = "192.168.1.76"
- Selection of the port SVC
  Factory setting: 42001
- Inactive timeout
  Factory setting: 120 [min]
- Timeout: Exact limit for a timeout (in seconds) between the robot request and the SVC response.
- Program interruption: By activating this option, the main program is interrupted as soon as a function of this PlugIn detects an error.

NOTE

Always call up the "SVC_INIT" command before using the 2D Grasping Kit for the first time. If changes are made to the communication settings, the robot controller needs to be restarted.



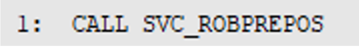
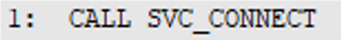
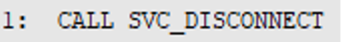
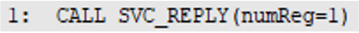
Reboot necessary

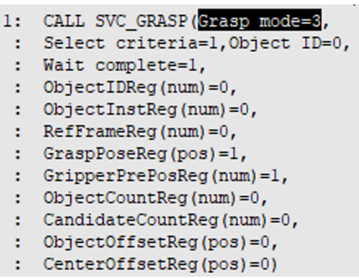
TEST1 LINE 0 T2 ABORTED JOINT

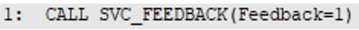
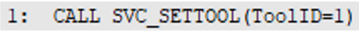
```
CALL SVC_INIT(ClientNo=5,IP Part1=192,IP Part2=168,IP Part3=1,
IP Part4=76,port=42001,inactive Timeout(min)=120,SVC Timeout(sec)=30,
Interrupt Prg.=1)
```

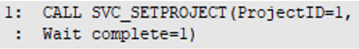
6 Creating robot program

After installing the software module, the following functions can be inserted into a robot program.

Function	Description
SVC_ROBPREPOS  `CALL SVC_ROBPREPOS`	The robot configuration is saved (CMOS). This configuration (e.g. F,U,T,0,0,0) is added to all positions reported by the SVC. The command can be executed at any time. No connection to the SVC is required.
SVC_CONNECT  `CALL SVC_CONNECT`	A connection to the SVC is established. The connection parameters are taken from the "Settings" (▸ 5 [🗋 13]). If no connection can be established, an error message appears.
SVC_DISCONNECT  `CALL SVC_DISCONNECT`	The connection to the SVC is disconnected.
SVC_REPLY ▸ 6.1 [🗋 17]  `CALL SVC_REPLY(numReg=1)`	Configuration of the register where the response of the SVC commands is saved.

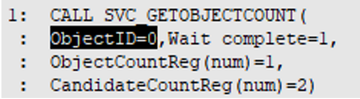
SVC_GRASP
▸ 6.2 [🗋 18]

`CALL SVC_GRASP(Grasp mode=3,`
`Select criteria=1,Object ID=0,`
`Wait complete=1,`
`ObjectIDReg(num)=0,`
`ObjectInstReg(num)=0,`
`RefFrameReg(num)=0,`
`GraspPoseReg(pos)=1,`
`GripperPrePosReg(num)=1,`
`ObjectCountReg(num)=0,`
`CandidateCountReg(num)=0,`
`ObjectOffsetReg(pos)=0,`
`CenterOffsetReg(pos)=0)`
	Starts object detection and receives the grasping pose for an object. All necessary information is provided to handle a grasping task. The function is parameterized by specifying the grasping mode and object ID.

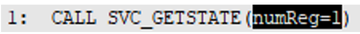
SVC_FEEDBACK ▸ 6.3 [🗋 19]  `CALL SVC_FEEDBACK(Feedback=1)`	Sends a feedback signal to the SVC in relation to the grasping action performed.
SVC_SETTOOL ▸ 6.5 [🗋 19]  `CALL SVC_SETTOOL(ToolID=1)`	The SVC is informed that a new tool is being used (observe SVC instructions, Chapter "Tool").

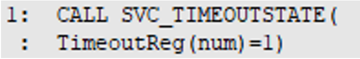
SVC_SETPROJECT
▸ 6.4 [🗋 19]

`CALL SVC_SETPROJECT(ProjectID=1,`
`Wait complete=1)`
	Requests a project change based on the project ID.

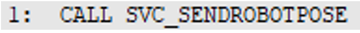
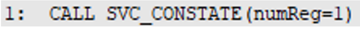
SVC_GETOBJECTCOUNT ▸ 6.7 [🗋 20]	This function makes it possible to analyze the scene. The number of all objects and the number of instances of a specified object class are determined.

Function	Description


`CALL SVC_GETOBJECTCOUNT(`
`ObjectID=0,Wait complete=1,`
`ObjectCountReg(num)=1,`
`CandidateCountReg(num)=2)`	

SVC_GETSTATE ▸ 6.8 [🗋 20]  `CALL SVC_GETSTATE(numReg=1)`	The function queries the SVC status and writes it to the configured numeric register.

SVC_TIMEOUTSTATE
▸ 6.9 [🗋 21]

`CALL SVC_TIMEOUTSTATE(`
`TimeoutReg(num)=1)`
	The command monitoring status is determined. The time required for a request is written to the configured register. A negative value indicates that a timeout has occurred. (Timeout error)

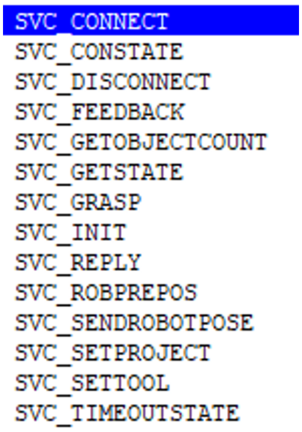
SVC_SENDROBOTSTATE  `CALL SVC_SENDROBOTPOSE`	The current robot position (X,Y,Z,W,P,R) is transmitted to the SVC. The robot configuration is not taken into account here. For further information, see the SVC commissioning instructions.
SVC_CONSTATE  `CALL SVC_CONSTATE(numReg=1)`	The connection status is queried and the result is written to a numerical register.

Further information on programming with the editor can be found in the FANUC manuals.

Add function to robot program

1. Start the program editor and select the "CALL program" command.

   ⇨ All available functions are displayed in the menu.

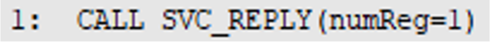


2. Select the desired SVC command from the list.
3. Adjust parameters if needed. To do this, select and change parameters in the command. Constant parameters can also be transferred via registers.

   ⇨ For further information on the functions, see the following sections.

6.1 SVC_REPLY

Configuration of the register in which the response of the SVC commands is saved.



```
1:  CALL SVC_REPLY(numReg=1)
```

Possible return values

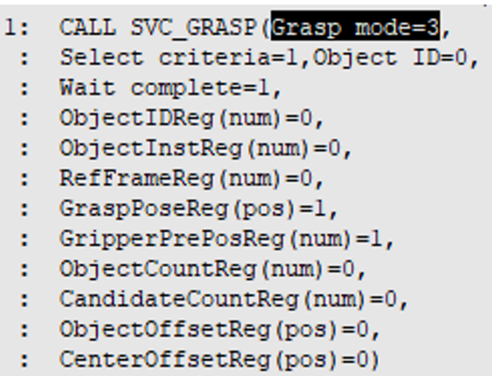
Return value	Description
1	Successful
2	Error
3	No object found (only for *SVC_GRASP*)
4	No grasp found (only for *SVC_GRASP*)
5	Invalid object number (only for *SVC_GRASP* and *SVC_GETOBJECTCOUNT*)
6	Camera not connected (only for *SVC_GRASP* and *SVC_GETOBJECTCOUNT*)
7	Camera not calibrated (only for *SVC_GRASP* and *SVC_GETOBJECTCOUNT*)
8	Robot not calibrated
9	Workspace not calibrated (only for *SVC_GRASP* and *SVC_GETOBJECTCOUNT*)
10	No active project (only for *SVC_GRASP* and *SVC_GETOBJECTCOUNT*)
11	No active tool (only for *SVC_GRASP*)

The numeric register is always written to when a command is sent.

Note: This command only needs to be called once or whenever the response register is to be changed.

6.2 SVC_GRASP

This command is used to provide all the information required to handle a grasping task.

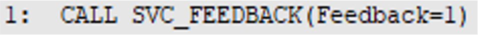


```
CALL SVC_GRASP(Grasp mode=3,
Select criteria=1,Object ID=0,
Wait complete=1,
ObjectIDReg(num)=0,
ObjectInstReg(num)=0,
RefFrameReg(num)=0,
GraspPoseReg(pos)=1,
GripperPrePosReg(num)=1,
ObjectCountReg(num)=0,
CandidateCountReg(num)=0,
ObjectOffsetReg(pos)=0,
CenterOffsetReg(pos)=0)
```

- Select gripper mode, object preselection and object class
  - As a fixed value (constant)
  - Dynamic via register
- Wait until the command has been completely processed.
  - **Wait complete**: The program sequence is stopped until the command has been processed and a response is available. The response is stored in a register that was defined with the "SVC_REPLY" command.
  - **No wait**: Further actions can be carried out after the command has been sent. The user is responsible for correct processing. (Monitor corresponding status information)
- Select the register in which the answer is written
  - Gripping position and center offset are written to position registers. The configuration of the robot (e.g. B. F,U,T,0,0,0,0) is taken from the position saved with "SVC_ROBPREPOS".
  - If register "0" is selected, this value is not reported back.
  - The gripping position and gripper position must always be read.

6.3 SVC_FEEDBACK

Sends a feedback signal in relation to the grasping action performed to the SVC.

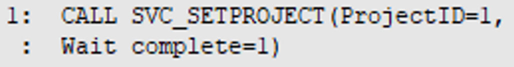


```
CALL SVC_FEEDBACK(Feedback=1)
```

Depending on the quality of the grasping action performed select ("1" = OK or "2" = Poor) as feedback (constant). The return value can also be written via a register.

6.4 SVC_SETPROJECT

This command switches the active project. It takes some time (up to 40 seconds) for this command to be fully processed.



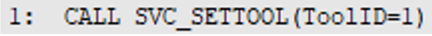
```
CALL SVC_SETPROJECT(ProjectID=1,
Wait complete=1)
```

- Variable for the project ID with the ID of the project to be activated.

  The project selection can also be made via a numerical register.
- Wait until the command has been completely processed.
  - **Wait complete**: The program sequence is stopped until the command has been processed and a response is available. The response is stored in a register that was defined with the "SVC_REPLY" command.
  - **No wait**: Further actions can be carried out after the command has been sent. The user is responsible for correct processing. (Monitor corresponding status information)

6.5 SVC_SETTOOL

This command switches the active tool.

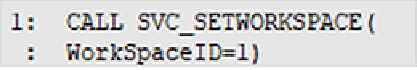


```
CALL SVC_SETTOOL(ToolID=1)
```

- The SVC is informed that a new tool is being used (observe SVC manual Chapter "Tool").

6.6 SVC_SETWORKSPACE

This command switches the workspace.

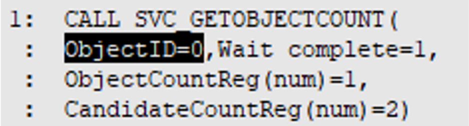


```
CALL SVC_SETWORKSPACE(
WorkSpaceID=1)
```

- The SVC is informed that a new workspace is being used (See SVC manual, "Workspace" chapter).

6.7 SVC_GETOBJECTCOUNT

This command determines the number of all objects and the number of instances of a specified object class.



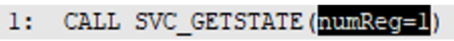
```
CALL SVC_GETOBJECTCOUNT(
ObjectID=0,Wait complete=1,
ObjectCountReg(num)=1,
CandidateCountReg(num)=2)
```

- Write the variable for the object class ID using the ID of the target object or select "0" for all objects.

  The result is written to the configured numerical register. If the register is set to "0", the result is ignored.
- Object selection can also be done via a numerical register.
- Wait until the command has been completely processed.
  - **Wait complete**: The program sequence is stopped until the command has been processed and a response is available. The response is stored in a register that was defined with the "SVC_REPLY" command.
  - **No wait**: Further actions can be carried out after the command has been sent. The user is responsible for correct processing. (Monitor corresponding status information)

6.8 SVC_GETSTATE

The command queries the SVC status and writes the result to the configured numeric register.

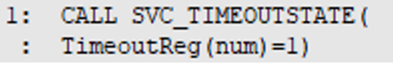


```
CALL SVC_GETSTATE(numReg=1)
```

Possible valid return values:
1 = initialization
2 = ready for operation
3 = stopped
4 = error

6.9 SVC_TIMEOUTSTATE

The "command monitoring" status (setting timeout) is written to a numerical register.



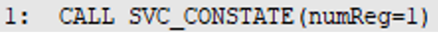
```
CALL SVC_TIMEOUTSTATE(
   :  TimeoutReg(num)=1)
```

If the time is exceeded, the time that has elapsed until the error occurs is displayed in negative. Otherwise, you can query the time (positive in sec.) that has elapsed since a command was sent.

In the event of an error, an alarm is always triggered (warning, error depending on configuration, ▸ 5 [🗋 13])

6.10 SVC_CONSTATE

The connection status is queried and the result is written to a numerical register.



```
CALL SVC_CONSTATE(numReg=1)
```

Possible valid return values:

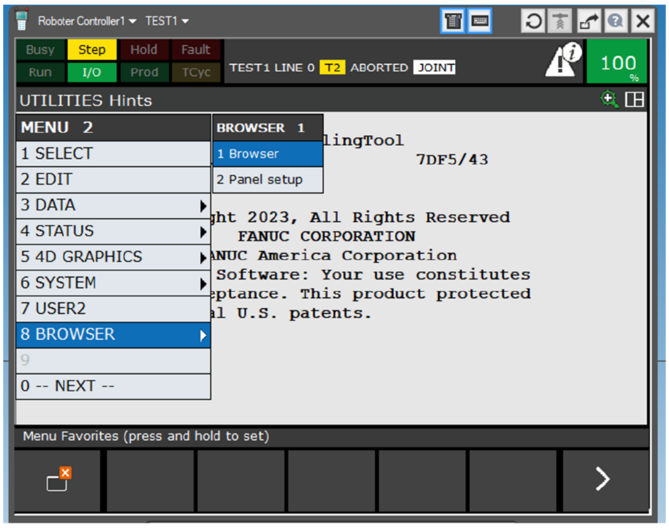
0 = not connected

1 = connected

6.11 View

An image with additional information is displayed. A new screen is calculated after each "SVC_GRASP" or "SVC_GETOBJECTCOUNT" command. The image can be called up in the pedant via the browser.

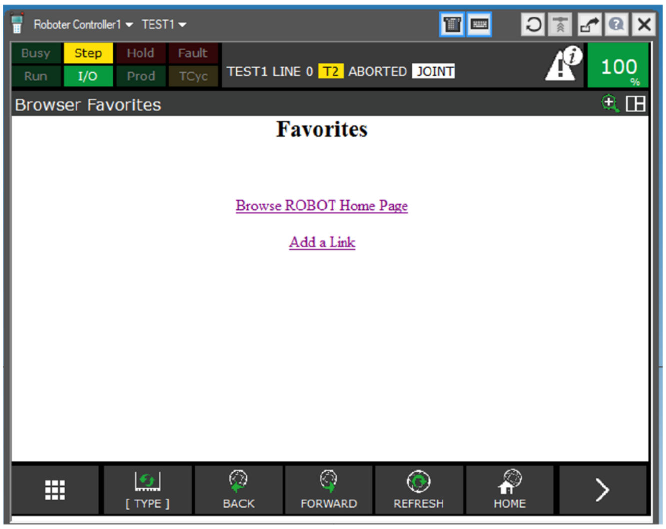
1. *Menu > Next > Browser > Open browser.*



```
lingTool
          7DF5/43

ght 2023, All Rights Reserved
   FANUC CORPORATION
NUC America Corporation
 Software: Your use constitutes
ptance. This product protected
l U.S. patents.
```

2. Select "Add a Link".

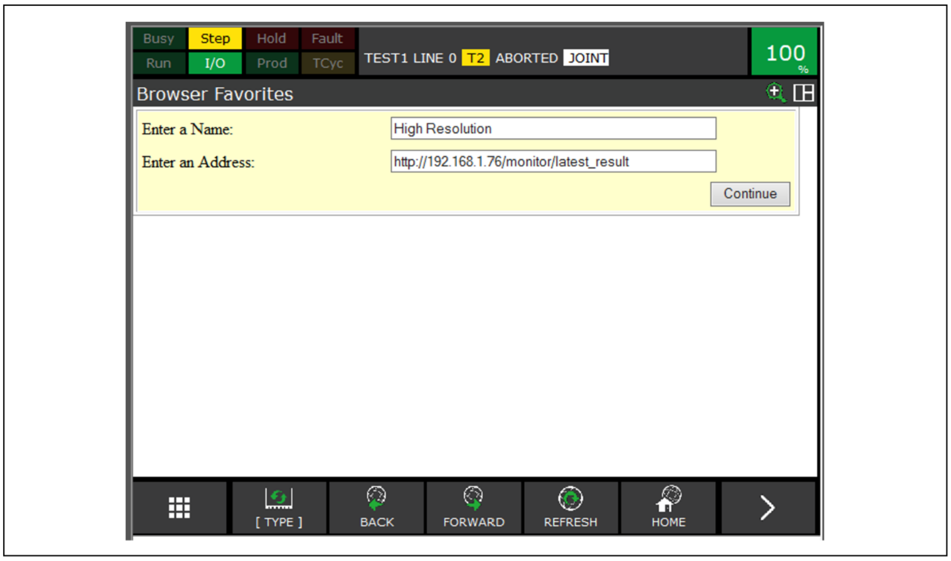


Favorites

Browse ROBOT Home Page

Add a Link

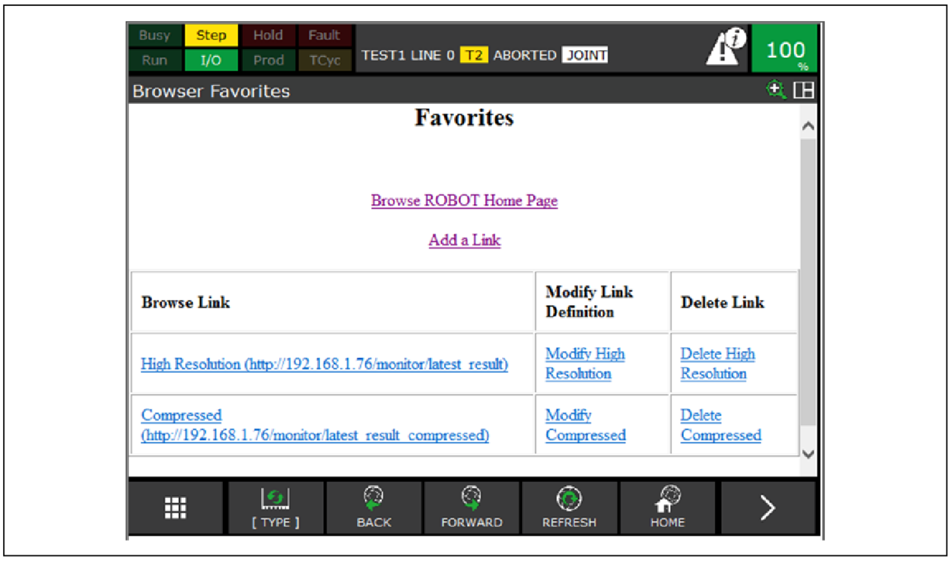
3. Enter the address depending on the resolution.
   - ⇨ High resolution:
     http://*<ip SVC box>*/monitor/latest_result
   - ⇨ Low resolution:
     http://*<ip SVCbox>*/monitor/latest_result_compressed



Enter a Name: High Resolution

Enter an Address: http://192.168.1.76/monitor/latest_result

4. Select the desired link under "BrowseLink".
   - ⇨ Image is displayed in the corresponding resolution.
5. To refresh, select "Refresh" (after command "SVC_GRASP" or "SVC_GETOBJECTCOUNT").



Favorites

Browse ROBOT Home Page

Add a Link

Browse Link	Modify Link Definition	Delete Link
High Resolution (http://192.168.1.76/monitor/latest_result)	Modify High Resolution	Delete High Resolution
Compressed (http://192.168.1.76/monitor/latest_result_compressed)	Modify Compressed	Delete Compressed

7 Example of a robot program

This "Pick&Place" example program can be used as a guide for creating individual applications.

Prerequisites

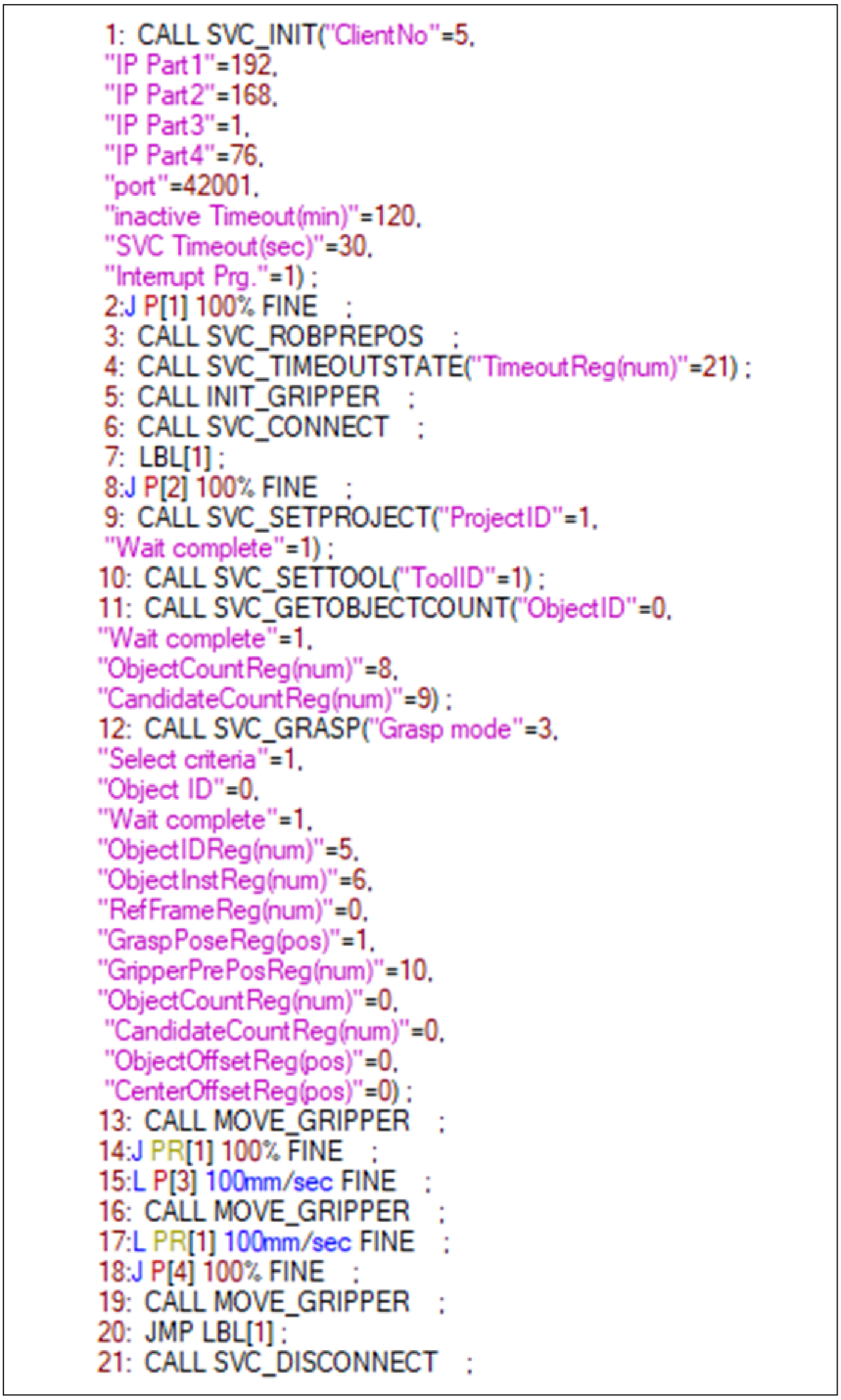
The following preconditions must be met:

- Option package *User Socket Msg (R648)* is installed.
- Gripper is installed.
- Tool is set up.
- SVC extension is installed.
- SVC is configured and ready for operation.
- Gripper is configured and ready for operation.

NOTE

If the gripping position is approached via a joint command, the option package *MROT (R640)* is recommended.

TP Code of a sample program



```
  1:  CALL SVC_INIT("ClientNo"=5,
  "IP Part1"=192,
  "IP Part2"=168,
  "IP Part3"=1,
  "IP Part4"=76,
  "port"=42001,
  "inactive Timeout(min)"=120,
  "SVC Timeout(sec)"=30,
  "Interrupt Prg."=1) ;
  2:J P[1] 100% FINE    :
  3:  CALL SVC_ROBPREPOS    ;
  4:  CALL SVC_TIMEOUTSTATE("TimeoutReg(num)"=21) ;
  5:  CALL INIT_GRIPPER    ;
  6:  CALL SVC_CONNECT    ;
  7:  LBL[1] ;
  8:J P[2] 100% FINE    ;
  9:  CALL SVC_SETPROJECT("ProjectID"=1,
  "Wait complete"=1) ;
 10:  CALL SVC_SETTOOL("ToolID"=1) ;
 11:  CALL SVC_GETOBJECTCOUNT("ObjectID"=0,
 "Wait complete"=1,
 "ObjectCountReg(num)"=8,
 "CandidateCountReg(num)"=9) ;
 12:  CALL SVC_GRASP("Grasp mode"=3,
 "Select criteria"=1,
 "Object ID"=0,
 "Wait complete"=1,
 "ObjectIDReg(num)"=5,
 "ObjectInstReg(num)"=6,
 "RefFrameReg(num)"=0,
 "GraspPoseReg(pos)"=1,
 "GripperPrePosReg(num)"=10,
 "ObjectCountReg(num)"=0,
  "CandidateCountReg(num)"=0,
  "ObjectOffsetReg(pos)"=0,
  "CenterOffsetReg(pos)"=0) ;
 13:  CALL MOVE_GRIPPER    ;
 14:J PR[1] 100% FINE    ;
 15:L P[3] 100mm/sec FINE    ;
 16:  CALL MOVE_GRIPPER    ;
 17:L PR[1] 100mm/sec FINE    ;
 18:J P[4] 100% FINE    ;
 19:  CALL MOVE_GRIPPER    ;
 20:  JMP LBL[1] ;
 21:  CALL SVC_DISCONNECT    ;
```

Line number	Description
1	Initialize the SVC.
2	Drive the robot close to the grasping task.

Line number	Description
3	Adopt the current robot configuration for the grasping task.
4	Use register 21 for timeout monitoring.
5	Initialize the gripper. Further information can be found in the commissioning instructions for the gripper. NOTE: This command is not part of this software and varies depending on the gripper manufacturer.
6	Establish a connection to the SCHUNK Vision Controller (SVC).
7	Set jump mark.
8	Move robot to "observation position". The camera image must not be obscured.
9	Select the project (optional).
10	Select tool (optional).
11	• Show found objects (optional). • Show all objects (0). • Wait until the command has been completely processed (1). • Write the result in registers 8 and 9: – Register 8 contains all objects found. – Register 9 contains all objects found that belong to the selected object ID. In this example, this corresponds to all objects, as the object ID has been set to "0".
12	Perform a calculation for a grasp: • Gripper mode: "Automatic" (3). • Object selection: "Random" (1). • Object ID to be gripped: "Any object" (0). • Wait until this command has been completed (1). • Write the object ID of the planned grasping object in register 5. • Write the object instance of the planned grasping object in register 6. • Write the calculated gripping position in position register 1.

Line number	Description
	• Write the calculated gripper finger position in position register 10. • Ignore the information about the number of objects found (0). • Ignore the information about the number of objects found that belong to the selected object ID (0). • Ignore the "Object Offset" (0). • Ignore the "Center Offset" (0).
13	Move the gripper finger to the calculated position. Further information can be found in the commissioning instructions for the gripper. NOTE: This command is not part of this software and varies depending on the gripper manufacturer.
14	Move to the calculated gripping position.
15	Move to gripping position (Z offset).
16	Grasp the object. Further information can be found in the commissioning instructions for the gripper. NOTE: This command is not part of this software and varies depending on the gripper manufacturer.
17	Move back to the calculated gripping position.
18	Drive to the storage position.
19	Put the object down Further information can be found in the commissioning instructions for the gripper. NOTE: This command is not part of this software and varies depending on the gripper manufacturer.
20	Jump to jump mark 1. The next object is gripped.
21	Terminates the connection to the SVC.

Tab.: Command Explanations

8 Troubleshooting

Message	Action
No connection to vision controller	• Check connection cable. • Check the network settings of the robot. • Check network settings on SVC (*Settings > Network > Robot*).
(IPL_SCHUNK_2DG-MAIN, xxx) Undefined build in	• Install option package *User Socket Msg (R648)*.
Wrong protocol version	• SVC Update. Request new version from SCHUNK.
Command timeout	• Check the timeout setting: • Check the connection to the SVC
Invalid object class	• Correct object class.
Camera not connected	• Check the connection to the camera • Check the connection of the camera to the SVC (correct port?).
Camera not calibrated	• Calibrate camera via SVC interface.
Robot not calibrated	• Calibrate robot via SVC interface.
Workspace not calibrated	• Calibrate workspace via SVC interface.
No active project	• Activate project via SVC interface. • Select project via "SVC_SETPROJECT" command.
No active tool	• Activate gripper via SVC interface. • Select gripper via "SVC_SETTOOL" command.
No grasp found	• Check SVC project (gripper settings). • Check the position of the gripper objects. • Check configuration of the robot.
No object found	• Check SCV project. • Check the camera's field of view. • Are objects available?
Error processing request	• Check last SCV command.
Unitialized data is used (after grasp command)	• Check robot configuration. • Use "SVC_SETROBOTPOSE".

SCHUNK SE & Co. KG
Spanntechnik | Greiftechnik | Automatisierungstechnik

Bahnhofstr. 106 – 134
D-74348 Lauffen/Neckar
Tel. +49-7133-103-0
info@de.schunk.com
schunk.com

Folgen Sie uns I *Follow us*

Wir drucken nachhaltig I *We print sustainable*