EWELLIX

EWELLIX Linear Module

SLIDEKIT 2.0 0S

User Manual

We pioneer motion

SCHAEFFLER

Contents

1 About the manual

1.1 Information in this user manual

This manual provides important information on how to work with the device safely and efficiently.

The manual is part of the device, must always be kept in the device's direct proximity and should be available for personnel to read at any time. All personnel working with the device must read and understand this manual before starting any work. Strict compliance with all specified safety notes and instructions is a basic requirement for safety at work.

Moreover, the accident prevention guidelines and general safety regulations applicable at the place of use of the device must also be complied with.

The EWELLIX SLIDEKIT 2.0 OS will be referred to as SLIDEKIT in the following.

1.2 Symbols

Safety precautions are identified by symbols and signal words as shown. The signal words indicate the severity of the hazard and the chance it could occur. Follow these safety precautions and act cautiously in order to avoid accidents, personal injury and damage to property.

The warning and hazard symbols are defined in accordance with ANSI Z535.6-2011.

1 Warning and hazard symbols

Signs and descriptions	
⚠ DANGER	In case of non-compliance, death or serious injury will occur.
⚠ WARNING	In case of non-compliance, death or serious injury may occur.
⚠ CAUTION	In case of non-compliance, minor or moderate injury may occur.
NOTICE	In case of non-compliance, damage or malfunctions in the product or the adjacent construction may occur.

1.3 Signs

The warning, prohibition, and mandatory signs are defined in accordance with DIN EN ISO 7010 or DIN 4844-2.

2 Warning, prohibition, and mandatory signs

Signs and descriptions	
	General warning
	Electrical voltage warning
	Hot surface warning
	Flammable materials warning
	Observe the manual

Signs and descriptions	
	Wear safety shoes
	Wear eye protection
	General mandatory sign

1.4 Legal notices

The information in this manual reflects the status at the time of publication.

Unauthorized modifications to or improper use of the product are not permitted. Schaeffler accepts no liability in these cases.

1.5 Limitation of liability

All information and notes in this manual were compiled with due consideration given to applicable standards and regulations, the present state of technology and our years of knowledge and experience.

The manufacturer is not liable for any damage resulting from:

- disregarding this manual
- unintended use
- employment of untrained personnel
- unauthorized conversions
- technical changes
- manipulation or removal of the screws on the drive
- use of unapproved spare parts.

Where the device has been customized, the actual product delivered may be different from what is described in this manual. In this case, ask Schaeffler for any additional instructions or safety precautions relevant to these devices.

We reserve the right to make technical modifications to the device to improve usability.

1.6 Availability

A current version of this manual is available at:
https://www.schaeffler.de/std/2231

Ensure that this manual is always complete and legible and is available to all persons engaged in transporting, fitting, dismantling, commissioning, operating, or maintaining the product.

Keep the manual in a safe place for immediate reference.

1.7 Images

The images in this manual may be schematic representations and may differ from the delivered device.

2 General safety regulations

This chapter contains generally applicable safety instructions that apply in addition to the safety instructions described in the chapters. Failure to comply with the guidelines and safety instructions contained in this manual may result in serious hazards that could cause death or serious injury to persons or damage to the device or equipment.

The safety instructions listed must always be taken into account before using the SLIDEKIT.

2.1 Intended use

SLIDEKIT is a horizontal linear axis for robots.

SLIDEKIT has been designed and built for the intended use described in the user manual of the linear module, with additional intended use defined as moving a robot in order to extend its operating range within an industrial environment.

Any use beyond the intended use or differing from the use described above is considered improper use.

Any claims resulting from damage due to improper use are excluded.

2.2 Non-intended Use

The device must not be operated in potentially explosive atmospheres.

Do not use the device to lift people.

2.3 Applicable documents

This user manual does not replace the user manuals of the supplied components but adds additional instructions relevant to the setup and operation of the SLIDEKIT system related to cobots.

Safe and proper operation is only possible if this user manual and the following information are heeded:

- user manual for the linear module and other components supplied
- regulations applicable at the place of use, according to the system into which the linear module is integrated and the prevailing environmental conditions
- regulations of the supervisory authorities (UVV accident prevention regulations)
- recognized technical rules for safe and professional work
- local laws and regulations
- environmental protection regulations
- other applicable regulations

Further information

Further information can be found in the following publication:

BA 129 | EWELLIX Linear Modules | CLSM |
https://www.schaeffler.de/std/2222

2.4 Qualified personnel

Operator duties:

- Ensure that only qualified and authorized personnel carry out the activities described in these instructions.
- Ensure that personal protective equipment is used.

Qualified personnel meet the following criteria:

- Product knowledge, e.g. by receiving training on how to use the product
- are fully familiar with the contents of this manual and, in particular, with all of the safety instructions
- are familiar with the relevant country-specific regulations

2.5 Safety regulations

The following safety regulations must be observed when working with the product. Additional information on hazards and specific behavioral instructions can be found, for example, in the chapters on Installation, Commissioning, Operation, and Maintenance.

2.5.1 Functional safety

SLIDEKIT is not a functional safety system that meets ISO 13489-1 and IEC 62061 standards. To integrate the SLIDEKIT into a functional safety chain, external safety devices must be added to the overall system

2.5.2 Safety equipment

- Pinching risk between the carriage and the end block of SLIDEKIT is minimized.
- Operation requires the SLIDEKIT controller to be connected to the safety I/O in the UR software.
- The SLIDEKIT controller checks the CANopen connection to the UR controller. If this connection is interrupted, the movement of the linear module is automatically stopped at the predefined position.
- If the UR software stops or fails, a stop signal should be sent to the SLIDEKIT controller. This stop signal must be programmed during the setup of the safety relay.

The following safety equipment has been integrated into SLIDEKIT to reduce the risk of personal injury or property damage:

2.5.2.1 Emergency stop

- Integration with an emergency-stop system is required for its intended use.
- Ensure that emergency stop functions are installed for the linear module and integrated in the safety chain of the entire system before using SLIDEKIT.
- The emergency stop function must be connected in such a way that a disruption of the power supply or the activation of the power supply after a power disruption cannot cause a hazardous situation for persons and objects.
- The emergency-stop systems must always be freely accessible.
- To integrate SLIDEKIT into a functional safety system with a STO (Safe Torque Off) safe condition, an external safety relay must be connected to the SLIDEKIT controller power supply, triggered by a functional safety function, such as a UR safety I/O.

2.5.2.2 Safety relay

Activation of the UR emergency stop will trigger a stop of the controller via 2 safety relays, certified ISO 13849-1. If the UR system is turned off, SLIDEKIT cannot be operated.

2.5.3 Potential risks

The following risks associated with operation of the SLIDEKIT must be taken into account as part of the application-specific risk assessment.

- The SLIDEKIT does not detect an impact automatically and will not stop movement in the event of an impact.
  This can lead to:
  - A person or object being crushed within the path of the linear module, which may result in serious injury.
  - Dynamic impact with a person or an object, which may cause serious injury.
- The SLIDEKITmovement does not stop at the desired position, and the UR-control software does not detect this.
  - Movement of the robot can occur at a different position than intended, potentially causing significant damage.

2.5.4 Commissioning

Only qualified personnel may start up the system.

Disconnect the power supply to the drive before performing installation or maintenance work.

Ensure that the drive is not under load or voltage before installation or maintenance work is performed.

Use suitable safety equipment when installing or maintaining the drive.

The power supply must comply with the technical specifications.

2.5.5 Maintenance and Repairs

Maintenance work and repairs may only be carried out by qualified personnel.

Disconnect the power supply to the drive before performing installation or maintenance work.

Ensure that the drive is not under load or voltage before installation or maintenance work is performed.

Use suitable safety equipment when installing or maintaining the drive.

3 Scope of delivery

The scope of delivery comprises:

- 1 linear module (with motor)
- 1 attachment plate compatible with LIFTKIT. The robot plate is available upon request.
- 1 SLIDEKIT control box
- 1 PiBox Ethernet TCP/IP module
- 2 main power cables (1.5 m)
- 1 motor power cable (3 m)
- 1 CANopen interface cable (3 m)
- 1 cable for proximity switches No. 1 and No. 2 (3 m, mounted on the linear axis)
- 1 safety-I/O cable (3 m, not shown in illustration)
- 1 Cobot interface cable (3 m, Ethernet cable, not shown in illustration)
- 1 digital E/A interface cable (3 m) (optional)
- 1 M12-DB9 adapter cable (15 cm, mounted on the linear axis)
- 1 cable holder (mounted on the linear axis)
- 2 limit switches with sockets

The robot plate is not included in the scope of delivery. It can be supplied as an accessory on request, matched to the customer's robot model.

Cables of other lengths are available on request.

1 Scope of delivery

1	Handheld programming device (not supplied)	2	CLSM linear module (with motor)
3	Fixation plate for UR3, UR5, UR10 and UR16 robots with LIFTKIT	4	SLIDEKIT control box
5	Cable for proximity switches 1 and 2 (3 m)	6	Motor power cable (3 m)
7	Safety-I/O cable (3 m)	8	Digital-I/O interface cable (3 m) (optional)
9	CobotInterface cable (3 m)	10	USBStick with URCaps software (not included with SLIDEKIT-00)

3.1 System requirements

- The SLIDEKIT 0S S00 is suitable for small and medium-sized cobots weighing up to 50 kg.
- The SLIDEKIT 0S S20 is suitable for large cobots weighing up to 75 kg.
- The URrobot controller is compatible with the Ethernet TCP/IP communication protocol and is equipped with an RJ45port.
- Power consumption (at rated load): AC 120 V to 230 V, 0.9 kVA

3.2 Check for transport damage

1. Check the product immediately upon delivery for any damage during transit.
2. Report any damage during transit promptly as a complaint to the carrier.

3.3 Check for defects

1. Check the product immediately upon delivery for any visible defects.
2. Report any defects promptly to the distributor of the product.
3. Do not put damaged products into operation.

4 Product description

SLIDEKIT is a horizontal linear axis for robots.

5 Montage

5.1 Tools required

- 5-mm wrench
- 6-mm wrench

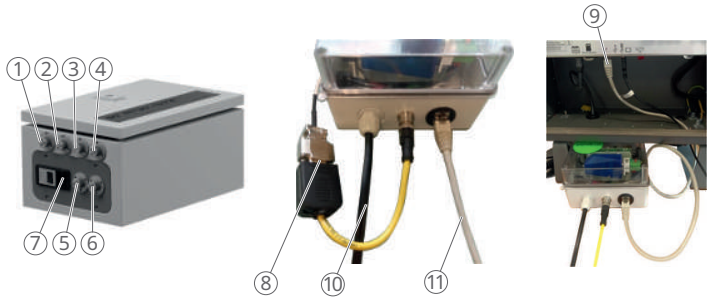
5.2 Robot installation on the linear module

1. Fasten the bottom plate to the SLIDEKIT base plate using M6 screws (not included in the scope of delivery).
2. Support the SLIDEKIT along its entire length or at least every 300 mm using a clamping unit or screw.
3. Mount the robot on a top plate specially manufactured for the customer's robot.
4. Align the robot or the SLIDEKIT using the alignment pins.
5. Secure the robot base using the 4 supplied screws, or the SLIDEKITbase using the 8 supplied screws.

A robot plate for attaching the robot to the SLIDEKIT can be ordered as an accessory upon request.

5.3 Connecting the power supply

2 Cable connections



001DC8FD

1	Cable for proximity switches No. 1 and No. 2	2	Safety-I/O cable
3	Communication interface to PiBox	4	Digital-I/O cable
5	CANopen interface cable of the servo-motor	6	Motor power cable
7	Main power cable	8	Cobot interface cable
9	RJ45 port on the robot controller	10	PiBox power cable
11	Ethernet cable		

1	Cable for proximity switches No. 1 and No. 2	2	Safety-I/O cable
3	Communication interface to PiBox	4	Digital-I/O cable
5	CANopeninterface cable of the servomotor	6	Motor power cable
7	Main power cable	8	Cobot interface cable
9	RJ45port on the robot controller	10	PiBox power cable
11	Ethernet cable		

1. Connect the main power cable to the connector (7) on the SLIDEKITcontroller.
2. Connect the two plugs of the motor power cable and the CANopen interface cable to the connectors for the motor power cable (5) and CANopeninterface cable (6).
3. Connect the digital I/O interface cable to the input connector (4) (SLIDEKIT-00 version only).
4. Connect the communication interface (3) via the DB9 connector to the PiBox Ethernet TCP/IP module (8).
5. Connect the safety-I/O cable to the connector (2).
6. Connect the cable for the proximity switch to the connector (1).
7. Connect the safety-I/O cable (2) to a safety DO on the robot controller.
8. Connect the PiBox Ethernet TCP/IP module (8) to the robot controller (9) using an RJ45cable.
9. Connect the main power cable (10) to the PiBox .

The safety I/O must be configured in the Safety menu of the robot controller or PLC ➤16|6.1.1.

During wiring, ensure that both the SLIDEKITcontrol box and the PiBox are not powered.

6 Software operation

The SLIDEKIT 2.0 0S contains a PiBox (Ethernet-TCP/IP module). This module enables control of the SLIDEKIT via Ethernet TCP/IP, port 50001, using script commands ➤17|6.2.

The IP address of the PiBox is 192.168.1.100 and can be changed ➤17|6.3.

6.1 SLIDEKIT Installation

Before commissioning, the SLIDEKIT requires the following settings:

- safety E/A: robot safety or SPSE/A
- communication
- SLIDEKIT setup via Ethernet TCP/IP script commands

Save the installation file so that the values are retained after a restart.

6.1.1 Setting up the safety I/O on the robot controller

To activate the safety I/O of the SLIDEKIT control box, the safety I/O connector (2) must be connected to the safety I/O of the robot or the safety I/O of the PLC . This output must provide DC 24 V in order to activate the safety relay that supplies power to the SLIDEKIT. If the safety I/O is not connected to and activated by the robot or PLC output, the SLIDEKIT cannot be operated, as the safety relay is normally open.

6.1.2 Communication

Communication between the SLIDEKIT and the robot or an external PLC takes place via TCP/IP communication through the RJ45 cable. This cable must be connected to the RJ45 port on the robot and the RJ45 port on the PiBox (Ethernet TCP/IP module).

The PiBox (Ethernet TCP/IP module) flashes green on the inside when the power supply is connected. If this is not the case, check the connection points.

6.1.3 Setting up the SLIDEKIT using Ethernet TCP/IP script commands

Always check the status of the SLIDEKIT first using the command get_status . If the result reads not ready, set up the SLIDEKIT using the following steps:

Step A:

1. Check the type using get_typesAvailable .
2. Select the type according to the SLIDEKIT in use using the script command set_type :

- SLIDEKIT 0S BE S00: SK 20x20 BG65S
- SLIDEKIT 0S PE S00: SK BELT 40 BG65X
- SLIDEKIT 0S PE S20: SK BELT 40 BG65S-C

Step B:

3. Perform homing using the command start_homing .

Step C:

4. Using the command set_virtualLimits,<lower limit>,<upper limit> set the virtual limits of the SLIDEKIT.

- The lower limit should generally be 0 .
- The lower limit should typically correspond to the value returned by the get_stroke command.

If the value returned by get_stroke deviates significantly from expectations, a problem may have occurred during homing, and the homing process should be repeated. Then check the status again using the script command get_status and make sure that the response reads ready .

6.2 Overview of PiBox Ethernet TCP-IP script commands

Command	Description
moveTo_absolutePosition	Moves the SLIDEKIT to the specified position.
get_typesAvailable	Returns all supported SLIDEKIT types.
get_type	Returns the current SLIDEKIT type.
set_type	Sets a new SLIDEKIT type.
get_velocity	Returns the velocity stored in the motion profile parameters.
set_velocity	Sets the velocity of the SLIDEKIT.
get_acceleration	Returns the acceleration stored in the motion profile parameters.
set_acceleration	Sets the acceleration of the SLIDEKIT.
get_deceleration	Returns the deceleration stored in the motion profile parameters.
set_deceleration	Sets the deceleration of the SLIDEKIT.
get_position	Returns the current position of the SLIDEKIT.
get_stroke	Returns the maximum stroke in mm.
get_status	Returns the current status of the SLIDEKIT.
stop_moving	Stops the movement of the SLIDEKIT.
get_motionProfileParameters	Returns the motion profile parameters.
set_motionProfileParameters	Sets the motion profile parameters.
start_homing	Starts homing.
stop_homing	Stops homing.
set_virtualLimits	Sets the virtual limits of the SLIDEKIT.
get_virtualLimits	Returns the virtual limits of the SLIDEKIT.

The SLIDEKIT 2.0 is not a functional safety system as defined in ISO 13489-1 or IEC 62061. To integrate the SLIDEKIT 2.0 into a functional safety chain, external safety devices must be incorporated into the overall system.

6.3 Set IP address for PiBox

The PiBox uses a static IP address. The default address is 192.168.1.100.

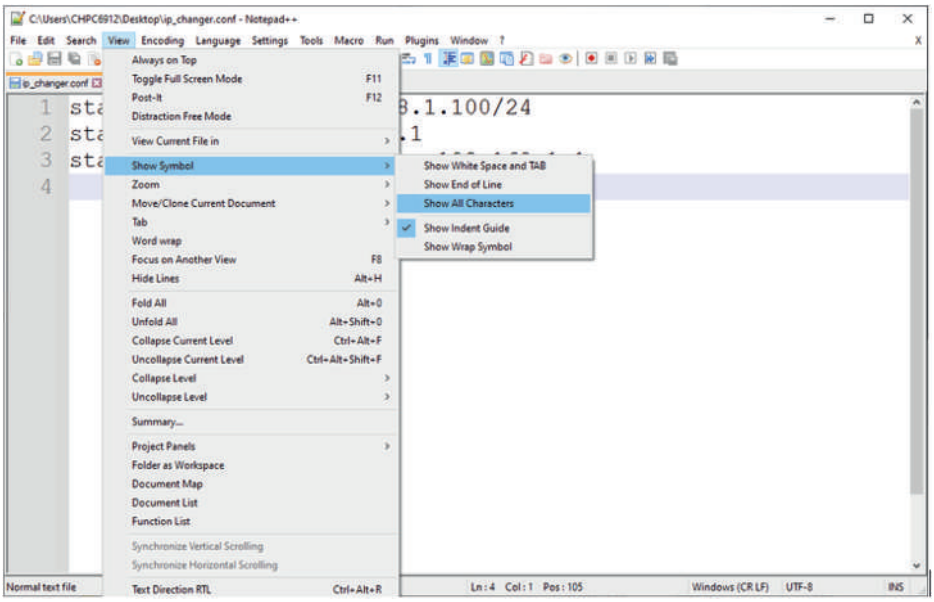
If a different IP address needs to be set, proceed as follows:

1. On the PC, create a file with the name ip_changer.conf .

A suitable program for this is Freeware Notepad++ or other similar program.

2. Insert the following content:
   - static ip_address=192.168.1.100/24
   - static routers=192.168.1.1
   - static domain_name_servers=192.168.1.1
3. Adjust the addresses. Ensure that /24 remains at the end of the static IP address.
4. Make all characters visible.

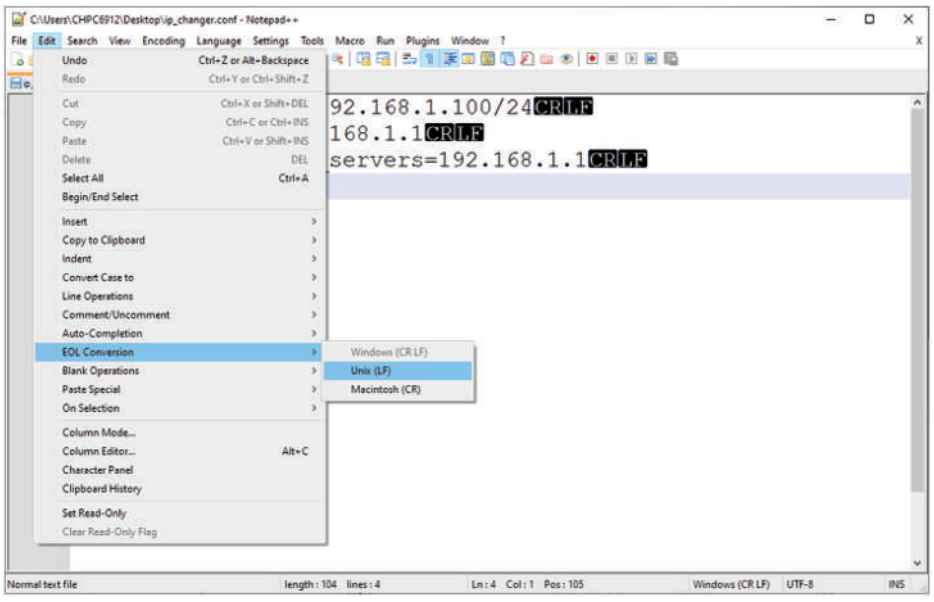
3 Show all characters



001CEEC3

5. Convert the line ending to Unix (LF).

4 EOL conversion to UNIX (LF) format



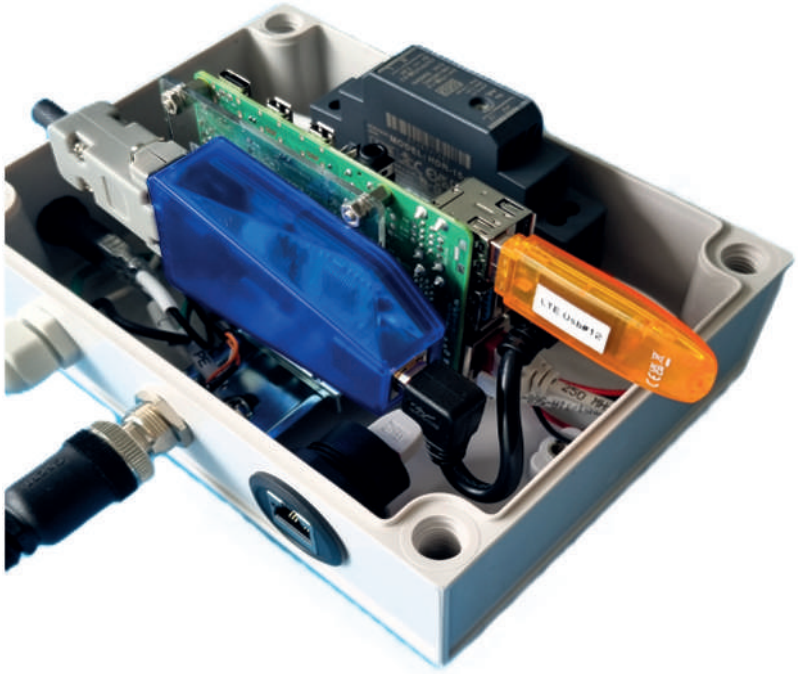
001CEEC2

6. Save the file to a USB stick previously formatted in FAT32.
7. Switch off the PiBox using the power switch.
8. Disconnect all connected cables.
9. Open the PiBox using a screwdriver.
10. Insert the USB stick into any available USB port on the PiBox.
11. Plug the power cable into the PiBox.
12. Switch on the PiBox using the power switch.
13. Wait 5 min .
14. Switch off the PiBox.
15. Disconnect all cables from the PiBox.

16. Remove the USB stick.
17. Reinstall the PiBox cover.
18. Secure the cover.

» The following empty file is created on the USB stick to confirm that the IP address change was successful: update_ip_address_success-full_from_“Name of USB stick”.

5 Inside view of the PiBox



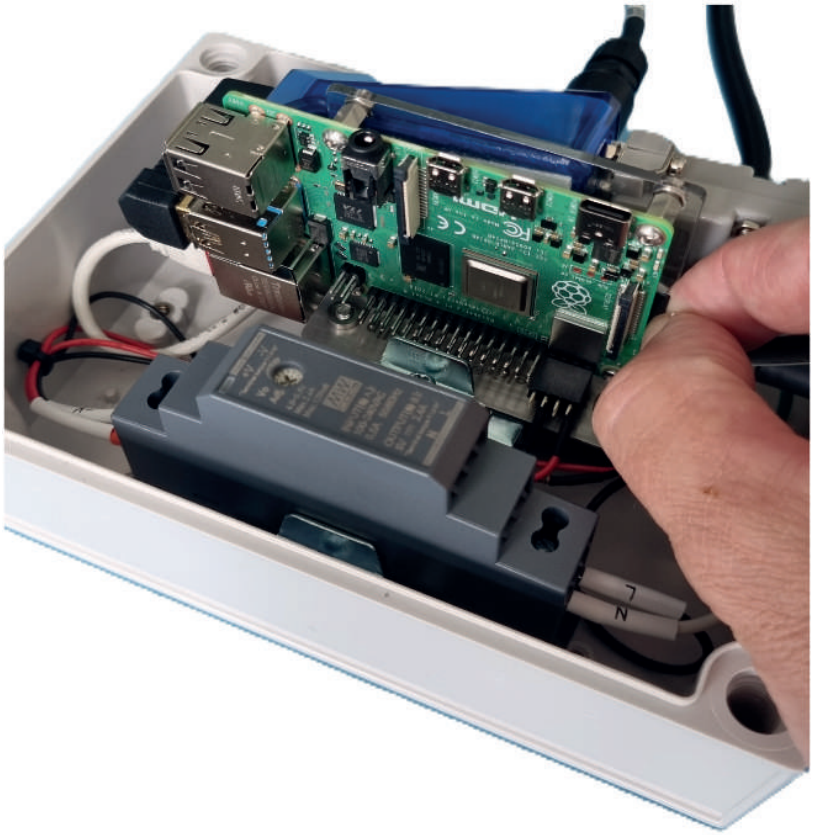
001CEEC4

6.4 Perform software update for PiBox

Software updates can be carried out by flashing a new image onto the controller’s SD card.

1. Switch off the PiBox using the power switch.
2. Remove all connection cables.
3. Open the PiBox.
4. Remove the micro SD card.

6 Removing the micro SD card



5. Download and install one of the following tools: Raspberry Imager or balenaEtcher
6. Copy the image file to the SD card.
7. Insert the SD card into the PC.

! Do not format the SD card.

8. Start the Raspberry Imager or balenaEtcher software.
9. Select image file.
10. Select SD card.
11. Start the writing process.
12. Reinsert the SD card into the PiBox.
13. Reinstall the PiBox cover.

Further information

- Raspberry Imager: https://www.raspberrypi.org/ downloads/
- balenaEtcher: https://www.balena.io/etcher/

6.5 PiBox API programming manual

Based on the current state of the SLIDEKIT, status information describing its current condition can be retrieved using Stargate commands. The status can be checked, for example, by sending the command get_status ➤34|6.7.14.

Some status messages include additional information that provides a more detailed explanation of the SLIDEKIT state.

6.5.1 Initialized

Stargate has been started and initialized. The following information indicates why the SLIDEKIT is not yet ready for movement.

3 Additional information

System message	Description	Status query
NOT_ CONNECTED_ TO_SLIDEKIT	The SLIDEKIT is not connected to Stargate.	get_status,OK,INITIALIZED, NOT_CONNECTED_TO_SLIDEKIT
CONNECTING	Connection is being established. This information is typically temporary and very brief, except in the event of a power failure of the SLIDEKIT, in which case it remains until an event occurs that causes a change in the status or in the additional information.	get_status,OK, INITIALIZED,CONNECTING
KEEP_ALIVE_ FAILED_OR_ POWER_CUT	Temporary condition during the state transition from CONNECTED to INITIALIZED.	get_status,OK,INITIALIZED, - KEEP_ALIVE_FAILED_OR _POWER_CUT

6.5.2 Connected

The connection between Stargate and the SLIDEKIT has been established.

Additional information:

Message	Description	Command to retrieve status response
TYPE_NOT_SET	The SLIDEKIT type has not been selected.	get_status,OK,CONNECTED, TYPE_NOT_SET
HOMING_ INTERRUPTED_ OR_NOT_STARTED	SLIDEKIT homing has not been started.	get_status,OK,CONNECTED, HOMING_INTERRUPTED_ OR_NOT_STARTED
HOMING_ INTERRUPTED_ BY_REQUEST	Homing was initiated but not completed successfully because it was stopped by the client using the command “stop_homing”.	get_status,OK,CONNECTED, HOMING_INTERRUPTED_ BY_REQUEST

Homing was started but not completed successfully.

Additional information:

Message	Error	Command to retrieve status response
HOMING_ ATTAINED_TR_ NOT_REACHED	-	get_status,OK, CONNECTED,HOMING_ATTAINED_ TR_NOT_REACHED
HOMING_ERROR_ VELOCITY_ IS_NOT_ZERO	The velocity during homing was not zero when it should have stopped.	get_status,OK, CONNECTED,HOMING_ERROR_ VELOCITY_IS_NOT_ZERO
HOMING_ERROR_ VELOCITY_ IS_ZERO	The velocity during homing was zero even though the SLIDEKIT should have been moving.	get_status,OK, CONNECTED,HOMING_ERROR_ VELOCITY_IS _ZERO
HOMING_ERROR_ NO_RESPONSE_ FROM_ MOTOR_NODE	No response from the motor.	get_status,OK, CONNECTED,HOMING_ERROR_NO_ RESPONSE_FROM_MOTOR_NODE
HOMING_ERROR_ FAILED_ TO_UPDATE_ PERSISTENCE	Homing values could not be written to the motor.	get_status,OK, CONNECTED,HOMING_ERROR_ FAILED_TO_UPDATE_PERSISTENCE

6.5.3 Homing active ready

Temporary condition that occurs when the SLIDEKIT is performing homing but is not actively moving.

Additional information: none

6.5.4 Homing active in motion

The SLIDEKIT is performing homing and is currently moving.

Additional information: none

6.5.5 Ready

The SLIDEKIT is ready to execute a movement. Additional information is provided about the result of the last movement:

Message	Description	Command to retrieve status response:
M2R_NOT_STARTED	No movement has been executed since homing.	get_status,OK,READY, M2R_NOT_STARTED
POSITION_IS_REACHED	The previously required position has been reached.	get_status,OK,READY, POSITION_IS_REACHED
PROGRAM_STOPPED	Movement stopped by the customer.	get_status,OK,READY, PROGRAM_STOPPED

If a problem occurs during movement and the position is not reached, the following additional information may be displayed:

Message	Description	Command to retrieve status response
SLIDEKIT_RUN_INTO_THE_FAULT_STATE	The SLIDEKIT has entered a FAULT STATE for some reason. The error may be a secondary fault that occurs when the motor cannot keep up with the commands it receives.	get_status,OK,READY, SLIDEKIT_RUN_INTO_THE_FAULT_STATE
M2R_NO_SLIDEKIT_CONNECTION_ERROR	A communication error has occurred with the SLIDEKIT. For example, the communication cable between the PiBox and the SLIDEKIT may have been disconnected.	get_status,OK,READY,M2R_NO_SLIDEKIT_CONNECTION_ERROR
M2R_NO_RESPONSE_FROM_MOTOR	The motor did not send a response as expected (e. g., motor malfunction).	get_status,OK,READY,M2R_NO_SLIDEKIT_CONNECTION_ERROR
POSITION_IS_NOT_REACHED	The commanded position was not reached (e. g., because the SLIDEKIT may have been obstructed).	get_status,OK,READY, POSITION_IS_NOT_REACHED

If the program is stopped in the READY state by sending the “stop_moving” command. In this case, the additional information is a combination of the client command (stop_moving) and the result of the previous movement:

Message	Description	Command to retrieve status response
M2R_PROGRAM_STOPPED__M2R_POSITION_REACHED	The program was stopped, but the last commanded position was reached (e. g., it was stopped during execution of the get command).	get_status,OK,READY,M2R_PROGRAM_STOPPED__M2R_POSITION_REACHED
M2R_PROGRAM_STOPPED__M2R_POSITION_NOT_REACHED_ERROR	The program was stopped during movement, and the commanded position had not yet been reached.	get_status,OK,READY,M2R_PROGRAM_STOPPED__M2R_POSITION_NOT_REACHED_ERROR
M2R_PROGRAM_STOPPED__M2R_MOTOR_RUN_INTO_FAULT_STATE	The program was stopped, but the system also entered the fault state “FAULT” (e. g., due to a “secondary error”)	get_status,OK,READY,M2R_PROGRAM_STOPPED__M2R_MOTOR_RUN_INTO_FAULT_STATE

6.5.6 Movement

The SLIDEKIT is executing a movement.

Additional information: none

6.6 Confirmation commands

Each command returns a confirmation as part of its response. The confirmation indicates whether the command was executed successfully. Depending on the confirmation, the response may contain different numbers of return values.

4 Command confirmation

Message	Description	Response format
OK	The command was successfully triggered or successfully executed.	• If the command returns a value: <command name>, OK, <value 1>,<value 2>...<value n> – Value 1, value 2, value n: values returned by a command Commands can return different numbers of values • If the command does not return a value: <command name>, OK
OOR	Out of range The command cannot be executed because at least one parameter is out of range.	<command name>, OOR, <index of received parameter>, <minimum value >, <maximum value> • Index of received parameter: index of received parameter that is out of range • minimum value: the minimum value for this parameter • maximum value: the maximum value for this parameter
NF	Not found. The specified command does not exist.	unspecified_command,OK
NA	Not allowed. The command cannot be executed in the current state.	<command_name>

Message	Description	Response format
WNP	Wrong number of parameters used in the command.	<command name>, WNP, <minimum value>, <maximum value> • minimum value: minimum number of required parameters • maximum value: maximal permissible number of parameters
VE	Value error. Incorrect data type for the specified client parameter.	<command name>, VE, <index of received parameter>, <data type> • Index of received parameter: index of a parameter with the incorrect type • Data type: Type required in the previously specified index. Possible types: integer, floating point (with one decimal place), string
EF	Execution failed. Command and parameters are valid, but execution of the command failed.	<command name>, EF, <error cause 1>, <error cause 2>, <error cause n> Error cause: Additional information describing why execution of the command failed.

6.7 Stargate commands

6.7.1 Stop homing

5 Stop homing

Command	Description	Status change
stop_homing	The command stops the homing process. If the homing process is stopped by the client command “stop_homing”, the status is always set to “CONNECTED,HOMING_INTERRUPTED_BY_REQUEST”.	When the command is executed successfully, the SLIDEKIT status changes from “HOMING_ACTIVE_READY” or “HOMING_ACTIVE_MOVING” to “CONNECTED,HOMING_INTERRUPTED_BY_REQUEST” Client format: stop_homing Parameters: none

6 Permitted states: stop_homing

Command	Description
HOMING_ACTIVE_READY	In home position
HOMING_ACTIVE_MOVING	Moving to home position

7 Possible confirmations: stop_homing

Message	Description	Response format
OK	Command executed successfully. Homing stopped successfully.	stop_homing,OK
WNP	Wrong number of parameters specified. The command should not contain any parameters.	stop_homing,WNP,0,0
NA	Not allowed. Stop homing cannot be executed in the current state.	stop_homing,NA

6.7.2 Move to position

8 Move to absolute position

Command	Description	State change
moveTo_ absolutePosition	The command moves the SLIDEKIT to the specified position. Optionally, the command can include additional parameters to set the velocity, acceleration, and deceleration. The values for velocity, acceleration, and deceleration are used for the current command only and are not stored as motion profile parameters. If only 1, 2. or 3 parameters are specified, the missing values are taken from the motion profile parameters. If no motion profile parameters have been set using the following commands: • set_motionProfileParameters • set_velocity • set_acceleration • set_deceleration and no optional parameters have been specified, the command returns an error. When the command “moveTo_absolute Position” is triggered, the SLIDEKIT is locked for manual movement. Even after the SLIDEKIT has reached the required position, it remains locked.	A successfully executed command changes the SLIDEKIT state from “READY,<*>” to “MOVING”. Client formats: • moveTo_absolutePosition,<position> • moveTo_absolutePosition,<position>,<velocity> • moveTo_absolutePosition,<position>,<velocity>,<acceleration> • moveTo_absolutePosition,<position>,<velocity>,<acceleration>,<deceleration> Parameters: • Position, FLOAT with one decimal place: position in mm (mandatory) • Velocity, INTEGER: acceleration in mm/s (optional) • Acceleration. INTEGER: acceleration in mm/s² (optional) • Deceleration, INTEGER: deceleration in mm/s² (optional)

9 Permitted states: moveTo_absolutePosition

Command	Description
READY	Ready

10 Possible confirmations: moveTo_absolutePosition

Message	Description	Response format
OK	Movement triggered successfully.	moveTo_absolutePosition,OK
OOR	One of the parameters is out of range.	moveTo_absolutePosition,OOR,<index of received parameter>,<minimum value>,<maximum value> • Index of received parameter: index of the first received parameter that is out of range • minimum value: the minimum value for this parameter • maximum value: the maximum value for this parameter
WNP	Wrong number of parameters specified. The command should have at least 1 and a maximum of 4 parameters.	set_motionProfileParameters, WNP,1,4

Message	Description	Response format
NA	Not allowed. Movement is not possible in the current state.	moveTo_absolutePosition,NA
VE	Value error. One of the parameters has an invalid type.	moveTo_absolutePosition,VE,<index of received parameter>,<data type> • Index of received parameter: index of the first parameter with the incorrect type • Data type: type required in the previously specified index Examples: • moveTo_absolute Position,VE,1,FLOAT: the required position value is not of type FLOAT • moveTo_absolute Position,VE,2,INTEGER: the velocity value is not of type Integer • moveTo_absolute Position,VE,3,INTEGER: the acceleration value is not of type Integer • moveTo_absolute Position,VE,4,INTEGER: the deceleration value is not of type Integer
EF	Velocity, acceleration, and deceleration are not stored as motion profile parameters • set_motionProfileParameters • set_velocity • set_acceleration • set_deceleration	moveTo_absolutePosition,EF, Requested data do not exist:<motion parameter>

11 Examples for EF, moveTo_absolutePosition

Client format	Description	Response format
moveTo_ absolutePosition, <position>	Velocity, acceleration, and deceleration are not stored as motion profile parameters • set_motionProfileParameters • set_velocity • set_acceleration • set_deceleration	moveTo_absolutePosition,EF, Requested data do not exist: velocity, Requested data do not exist: acceleration, Requested data do not exist: deceleration
moveTo_ absolutePosition, <position>,<velocity>	Acceleration and deceleration not stored as motion profile parameters • set_motionProfileParameters • set_acceleration • set_deceleration	set_motionProfileParameters,EF, Requested data do not exist: acceleration, Requested data do not exist: deceleration
moveTo_ absolutePosition, <position>,<velocity>,<acceleration>	Acceleration and deceleration not stored as motion profile parameters • set_motionProfileParameters • set_deceleration	moveTo_absolutePosition,EF, Requested data do not exist: deceleration

6.7.3 Retrieve types

12 Retrieve types

Command	Description	State change
get_types Available	The command returns all supported SLIDEKIT types. The types are separated by ‘,’ (comma).	Client format: get_typesAvailable Parameters: none

13 Permitted states: get_typesAvailable

Command	Description
INITIALIZED	Initialized
CONNECTED	Connected
READY	Ready
MOVING	Movement
HOMING_ACTIVE_READY	In home position
HOMING_ACTIVE_MOVING	Moving to home position

14 Possible confirmations: get_typesAvailable

Message	Description	Response format
OK	Command executed successfully. Types successfully returned as a response.	get_typesAvailable,OK,<SLIDEKIT type 1>,<SLIDEKIT type 2>...<SLIDEKIT type n>
WNP	Wrong number of parameters specified. The client command should not contain any parameters.	get_typesAvailable,WNP,0,0

6.7.4 Retrieve type

15 Retrieve type

Command	Description	State change
get_type	The command returns the current SLIDEKIT type.	Client format: get_type Parameters: none

16 Permitted states: get_type

Command	Description
INITIALIZED	Initialized
CONNECTED	Connected
READY	Ready
MOVING	Movement
HOMING_ACTIVE_READY	In home position
HOMING_ACTIVE_MOVING	Moving to home position

17 Possible confirmations: get_type

Message	Description	Response format
OK	Command executed successfully. SLIDEKIT type is returned.	get_type,OK,<SLIDEKIT type>
WNP	Wrong number of parameters specified. The command should not contain any parameters.	get_type,WNP,0,0
EF	Execution failed. • Example: The SLIDEKIT type cannot be retrieved because no type has been set.	get_type,EF,Requested data do not exist:config_type Retrieve status: get_status,OK,CONNECTED, - TYPE IS NOT SET

6.7.5 Set type

18 Set type

Command	Description	State change
set_type	The command sets a new SLIDEKIT type.	The SLIDEKIT type must be one of the types returned by the command get_typesAvailable. State change: Upon successful execution, the command changes the SLIDEKIT state to “HOMING_INTERRUPTED_OR_NOT_STARTED” Client format: set_type,<type> Parameters: Type: STRING, required type (mandatory)

19 Permitted states: set_type

Command	Description
INITIALIZED	Initialized
CONNECTED	Connected
READY	Ready

20 Possible confirmations: set_type

Message	Description	Response format
OK	Command executed successfully. Type set successfully.	set_type,OK
WNP	Wrong number of parameters specified. The command should contain exactly one parameter.	set_type,WNP,1,1
OOR	The selected type is out of range.	set_type,OOR,ConfigurationHolder: no configuration exists with the name <type>
NA	Not allowed. The type cannot be executed in the current state.	set_type,NA
VE	Value error (e. g., empty string specified as the type).	set_type,VE,1,STRING
EF	Execution failed (e. g., attempt to set the SLIDEKIT type that is already selected).	set_type,EF,model with the name <type> already selected

6.7.6 Retrieve velocity

21 Retrieve velocity

Command	Description	State change
get_velocity	The command returns the velocity stored in the motion profile parameters. The unit of measurement for velocity is mm/s.	Client format: get_velocity Parameters: none

22 Permitted states: get_velocity

Command	Description
CONNECTED	Connected
READY	Ready
MOVING	Movement

23 Possible confirmations:get_velocity

Message	Description	Response format
OK	Command executed successfully. The velocity is returned as a response.	get_velocity,OK,<velocity>
WNP	Wrong number of parameters specified. The command should contain exactly one parameter.	get_velocity,WNP,0,0
NA	Not allowed. The velocity cannot be determined in the current state.	get_ velocity,NA
EF	Execution failed. • Example: The velocity cannot be retrieved because it has not been set	get_velocity,EF,Requested data do not exist:velocity

6.7.7 Set velocity

24 Set velocity

Command	Description	State change
set_velocity	The command sets the velocity. The velocity is stored as a motion profile parameter. If the SLIDEKIT is executing a movement, the command “set_velocity” has no effect on the current movement.	Client format: set_velocity,<velocity> Parameters: Velocity: INTEGER, required velocity in mm/s (mandatory).

25 Permitted states: set_velocity

Command	Description
CONNECTED	Connected
READY	Ready
MOVING	Movement “set_velocity” is not applied in the current movement. It can be applied in subsequent commands.

26 Possible confirmations: set_velocity

Message	Description	Response format
OK	Command executed successfully. Velocity set successfully.	set_velocity,OK
WNP	Wrong number of parameters specified. The command should contain exactly one parameter.	set_velocity,WNP,1,1
OOR	Out of range. The specified velocity is not within the defined velocity limits.	set_velocity,OOR,1,<minimum value>,<maximum value> • minimum value: minimum supported velocity • maximum value: maximum supported velocity
NA	Not allowed. The velocity cannot be set in the current state.	set_velocity,NA
VE	Value error. The velocity must be an integer.	set_velocity,VE,1,INTEGER
EF	Execution failed • Velocity cannot be set because the SLIDEKIT type has not been set.	set_velocity,EF, Requested data do not exist:min_velocity, Requested data do not exist:max_velocity Retrieve status response: get_status,OK,CONNECTED,TYPE_NOT_SET

6.7.8 Retrieve acceleration

27 Retrieve acceleration

Command	Description	State change
get_acceleration	The command returns the acceleration. The acceleration is stored as a motion profile parameter. The unit of measurement for acceleration is mm/s².	Client format: get_acceleration Parameters: none

28 Permitted states: get_acceleration

Command	Description
CONNECTED	Connected
READY	Ready
MOVING	Movement

29 Possible confirmations: get_acceleration

Message	Description	Response format
OK	Command executed successfully. The acceleration is returned as a response.	get_acceleration,OK,<acceleration>
WNP	Wrong number of parameters specified. The command should not contain any parameters.	get_acceleration,WNP,0,0
NA	Not allowed. Acceleration cannot be determined in the current state.	get_acceleration,NA
EF	Execution failed. • E. g.: The acceleration cannot be determined because it has not been set.	get_acceleration,EF,Requested data do not exist: acceleration

6.7.9 Set acceleration

30 Set acceleration

Command	Description	State change
set_acceleration	The command sets the acceleration. The acceleration is stored as a motion profile parameter.	Client format: set_acceleration,<acceleration> Parameters: Acceleration: INTEGER, required acceleration in mm/s² (mandatory)

31 Permitted states: set_acceleration

Command	Description
CONNECTED	Connected
READY	Ready
MOVING	Movement “set_acceleration” is not applied in the current movement. It can be applied in subsequent commands.

32 Possible confirmations: set_acceleration

Message	Description	Response format
OK	Command executed successfully. Acceleration set successfully.	set_acceleration,OK
WNP	Wrong number of parameters specified. The command should contain exactly one parameter.	set_ acceleration,WNP,1,1
OOR	Out of range. The specified acceleration is not within the defined acceleration limits.	set_ acceleration,OOR,1,<minimum value>,<maximum value> • minimum value: minimum supported acceleration • maximum value: maximum support acceleration
NA	Not allowed. The set acceleration cannot be executed in the current state.	set_ acceleration,NA
VE	Value error. The acceleration must be an integer.	set_ acceleration,VE,1,INTEGER
EF	Execution failed • For example, the acceleration cannot be set because the SLIDEKIT type has not been set	set_acceleration,EF, Requested data do not exist: min_acceleration, Request data do not exist: max_acceleration Retrieve status response: get_status,OK,CONNECTED, TYPE_NOT_SET

6.7.10 Retrieve deceleration

33 Retrieve deceleration

Command	Description	State change
get_ deceleration	The command returns the deceleration. The deceleration is stored as a motion profile parameter. The unit of measurement for deceleration is mm/s^2.	Client format: get_deceleration Parameters: none

34 Permitted states: get_ deceleration

Command	Description
CONNECTED	Connected
READY	Ready
MOVING	Movement

35 Possible confirmations: get_ deceleration

Message	Description	Response format
OK	Command executed successfully. The deceleration is returned as a response.	get_deceleration,OK,<deceleration>
WNP	Wrong number of parameters specified. The command should not contain any parameters.	get_deceleration,WNP,0,0
NA	Not allowed. Deceleration cannot be retrieved in the current state.	get_deceleration,NA
EF	Execution failed. • Example: The deceleration cannot be retrieved because it has not been set.	get_deceleration,EF,Requested data do not exist: deceleration

6.7.11 Set deceleration

36 Set deceleration

Command	Description	State change
set_deceleration	The command sets the deceleration. The deceleration is stored as a motion profile parameter.	Client format: set_deceleration,<deceleration> Parameters: Deceleration: INTEGER, required deceleration in mm/s² (mandatory)

37 Permitted states: set_deceleration

Command	Description
CONNECTED	Connected
READY	Ready
MOVING	Movement "set_deceleration" is not applied in the current movement. It can be applied in subsequent commands.

38 Possible confirmations: set_deceleration

Message	Description	Response format
OK	Command executed successfully. Deceleration set successfully.	set_deceleration,0
WNP	Wrong number of parameters specified. The command should contain exactly one parameter.	set_ deceleration,WNP,1,1
OOR	Out of range. The specified deceleration is not within the defined deceleration limits.	set_ deceleration,OOR,1,<minimum value >,<maximum value> • minimum value: minimum supported deceleration. • maximum value: maximum supported deceleration
NA	Not allowed. The set deceleration cannot be executed in the current state.	set_ deceleration,NA
VE	Value error. The deceleration must be an integer.	set_deceleration,VE,1,INTEGER
EF	Execution failed. • Deceleration cannot be set because the SLIDEKIT type has not been set.	set_deceleration,EF, Requested data do not exist:min_deceleration, Request data do not exist:max_deceleration Status query: get_status,OK,CONNECTED, TYPE_NOT_SET

6.7.12 Retrieve position

39 Retrieve position

Command	Description	Status change
get_position	The command returns the current position in mm.	Client format: get_position Parameters: none

40 Permitted states: get_position

Command	Description
CONNECTED	Connected
READY	Ready
MOVING	Movement
HOMING_ACTIVE_READY	In home position
HOMING_ACTIVE_MOVING	Moving to home position

41 Possible confirmations: get_position

Message	Description	Response format
OK	Command executed successfully. Position is returned in the response.	get_position,OK,<position> Confirmation with OK is expected in the following states: • CONNECTED,VIRTUAL_LIMITS_NOT_SET • READY,<additional information> • MOVING
WNP	Wrong number of parameters specified. The command should not contain any parameters.	get_ position,WNP,0,0
NA	Not allowed. The position cannot be determined in the current state.	get_ position,NA
EF	Execution failed • Position cannot be determined because homing has not been carried out.	get_position,EF,Homing is not done EF is expected in the following states: • HOMING_ACTIVE_READY • HOMING_ACTIVE_MOVING • CONNECTED,TYPE_NOT_SET • CONNECTED,HOMING_ INTERRUPTED_OR_ NOT_STARTED • CONNECTED,HOMING_ INTERRUPTED_ BY_REQUEST • CONNECTED,<homing error>

6.7.13 Retrieve stroke

42 Retrieve stroke

Command	Description	State change
get_stroke	The command returns the maximum stroke in mm. The maximum stroke is calculated during homing.	Client format: get_stroke Parameters: none

43 Permitted states:get_stroke

Command	Description
CONNECTED	Connected
READY	Ready
MOVING	Movement

44 Possible confirmations: get_stroke

Message	Description	Response format
OK	Command executed successfully. The stroke is returned as a response.	get_stroke,OK,<stroke>
WNP	Wrong number of parameters specified. The command should not contain any parameters.	get_stroke,WNP,0,0
NA	Not allowed. The stroke cannot be retrieved in the current state.	get_ position,NA
EF	Execution failed (e. g., the maximum stroke cannot be retrieved because homing has not been carried out).	get_position,EF, Requested data do not exist:max_stroke

6.7.14 Retrieve status

45 Retrieve status

Command	Description	State change
get_status	The command returns the current status, or the current status and additional information ➤34|6.7.14.	Status change: none Client format: get_status Parameters: none

46 Permitted states: get_status

Command	Description
CONNECTED	Connected
READY	Ready
MOVING	Movement
HOMING_ACTIVE_READY	In home position
HOMING_ACTIVE_MOVING	Moving to home position

47 Possible confirmations: get_status

Message	Description	Response format
OK	Command executed successfully. Status and, where applicable, additional information are returned as a response. • if the returned status includes additional information: get_status,OK,<status>,<additional information> • if the returned status does not include additional information	get_status,OK,<status>
WNP	Wrong number of parameters specified. The command should not contain any parameters.	get_ status,WNP,0,0

6.7.15 Stop movement

48 Stop movement

Command	Description	State change
stop_moving	The command stops the movement of the SLIDEKIT. When the command is executed, the SLIDEKIT is unlocked and can be moved manually.	See the overview below

When the command is executed successfully, the SLIDEKIT status changes:

49 Status change

Previous status	New status
MOVING	READY,PROGRAM_STOPPED
READY,POSITION_ IS_REACHED	READY,M2R_ PROGRAM_STOPPED_ M2R_POSITION_ REACHED
READY,SLIDEKIT_ RUN_INTO_THE_FAULT_STATE	READY,M2R_PROGRAM_STOPPED__ M2R_ MOTOR_RUN_INTO_FAULT_STATE
READY,POSITION_ IS_NOT_REACHED	READY,M2R_ PROGRAM_STOPPED__ M2R_POSITION_NOT_ REACHED_ERROR
	Client format:
	stop_moving

50 Permitted states: stop_moving

Command	Description
READY	Ready
MOVING	Movement

51 Possible confirmations: stop_moving

Message	Description	Response format
OK	Command executed successfully. SLIDEKIT stopped successfully.	stop_moving,OK
WNP	Wrong number of parameters specified. The command should not contain any parameters.	stop_moving,WNP,0,0
NA	Not allowed. Movement stop cannot be executed in the current state.	stop_moving,NA

6.7.16 Retrieve motion profile parameters

52 Retrieve motion profile parameters

Command	Description	State change
get_motion ProfileParameters	The command returns the motion profile parameters. Motion profile parameters consist of velocity, acceleration, and deceleration. Each motion profile parameter can be retrieved using “get_velocity”, “get_acceleration”, and “get_deceleration”.	Client format: get_motionProfileParameters Parameters: none

53 Permitted states: get_motionProfileParameters

Command	Description
CONNECTED	Connected
READY	Ready
MOVING	Movement

54 Possible confirmations: get_motionProfileParameters

Message	Description	Response format
OK	Command executed successfully. All profile parameters (velocity, acceleration, deceleration) are returned as a response.	get_motionProfileParameters, OK,<velocity>,<acceleration>,<deceleration>
WNP	Wrong number of parameters specified. The command should not contain any parameters.	get_motionProfileParameters,WNP,0,0
NA	Not allowed. Profile parameters cannot be retrieved in the current state.	get_motionProfileParameters,NA
EF	Execution failed. • For example, because one or more profile parameters have not been set.	Depending on how many profile parameters have not been set, the command may return multiple error messages. See the overview below.

55 Error description

Error	
If all profile parameters have not been set	get_motionProfileParameters,EF, Requested data do not exist:velocity, Requested data do not exist:acceleration, Requested data do not exist:deceleration
If the velocity has not been set	get_motionProfileParameters,EF, Requested data do not exist:velocity
If the acceleration has not been set	get_motionProfileParameters,EF, Requested data do not exist: acceleration
If the deceleration has not been set	get_motionProfileParameters,EF, Requested data do not exist: deceleration
If velocity and acceleration have not been set	get_motionProfileParameters,EF, Requested data do not exist: velocity, Requested data do not exist: acceleration
If velocity and deceleration have not been set	get_motionProfileParameters,EF, Requested data do not exist: velocity, Requested data do not exist: deceleration
If acceleration and deceleration have not been set	get_motionProfileParameters,EF, Requested data do not exist: acceleration, Requested data do not exist: deceleration

6.7.17 Set motion parameters

56 Set motion parameters

Command	Description	State change
set_motionProfileParameters	The command sets the motion profile parameters. The values stored in the profile parameters (velocity, acceleration, deceleration) are used as default values when executing a movement to an absolute position. Each motion parameter can also be set individually using the commands: • set_velocity • set_acceleration • set_deceleration	Client format: • set_motionProfileParameters, <velocity> • set_motionProfileParameters, <velocity>, <acceleration> • set_motionProfileParameters, <velocity>, <acceleration>,<deceleration> Parameters: • Velocity: INTEGER, required velocity mm/s (mandatory) • Acceleration, INTEGER, required acceleration mm/s^2 (optional) • Deceleration, INTEGER, required deceleration mm/s^2 (optional)

57 Permitted states: set_motionProfileParameters

Command	Description
CONNECTED	Connected
READY	Ready
MOVING	Movement “set_motionProfileParameters” is not applied in the current movement. It can be applied in subsequent commands

58 Possible confirmations: set_motionProfileParameters

Message	Description	Response format
OK	Command executed successfully. Profile parameters set successfully.	set_motionProfileParameters,OK
WNP	Wrong number of parameters specified. The command should have at least 1 and at most 3 parameters.	set_motionProfileParameters, WNP,1,3
OOR	Out of range. One of the specified parameters is not within the defined limits.	set_motionProfileParameters,OOR, <index of received parameter>,<minimum value>,<maximum value> • Index of received parameter: index of the first received parameter that is out of range • minimum value: the minimum value for this parameter • maximum value: the maximum value for this parameter
NA	Not allowed. Profile parameters cannot be set in the current state.	set_motionProfileParameters,NA
VE	Value error. All profile parameters must be of type INTEGER.	set_motionProfileParameters,VE, <index of received parameter>,INTEGER
EF	Execution failed. For example, because one or more profile parameters were specified, but the SLIDEKIT type has not been set. Depending on how many profile parameters were specified, the command may return multiple error messages.	–
	The velocity is specified in the client command (set_motionProfileParameters,<velocity>).	set_motionProfileParameters,EF, Requested data do not exist: min_velocity, Requested data do not exist: max_velocity
	Velocity and acceleration are specified in the client command (set_motionProfileParameters,<velocity>,<acceleration>).	set_motionProfileParameters,EF, Requested data do not exist: min_velocity, Requested data do not exist: max_velocity, Requested data do not exist: min_acceleration, Requested data do not exist: max_acceleration
	Velocity, acceleration, and deceleration are specified in the client command (set_motionProfileParameters,<velocity>,<acceleration>,<deceleration>)	set_motionProfileParameters,EF, Requested data do not exist: min_velocity, Requested data do not exist: max_velocity, Requested data do not exist: min_acceleration, Requested data do not exist: max_acceleration, Requested data do not exist: min_deceleration, Requested data do not exist: max_deceleration

6.7.18 Start Homing

59 Start homing

Command	Description	State change
start_homing	The command starts the homing process. While performing homing, the SLIDEKIT moves back to its minimum position and then extends to the maximum position. Once the homing process is completed, the SLIDEKIT stores the information for the maximum stroke.	A successfully triggered command changes the state of the SLIDEKIT. See below.

60 Status change

Previous status	New status
READY<*>	Temporarily to “HOMING_ACTIVE_READY”, then to “HOMING_ACTIVE_MOVING”.
CONNECTED,<*>	Temporarily to “HOMING_ACTIVE_READY”, then to “HOMING_ACTIVE_MOVING”.
If an error occurs during homing	CONNECTED,<homing error> Homing error ➤21|6.5.2 Client format: start_homing Parameters: none

61 Permitted states: start_homing

Command	Description
CONNECTED	Connected
READY	Ready

62 Possible confirmations: start_homing

Message	Description	Response format
OK	Homing triggered successfully.	start_homing,OK
WNP	Wrong number of parameters specified. The command should not contain any parameters.	start_homing,WNP,0,0
NA	Not allowed. Homing cannot be started in the current state.	start_homing,NA
EF	Execution failed. • Example: Homing cannot be started because the type has not been set.	start_homing,EF, Requested data do not exist:config_type Status query: get_status,OK,CONNECTED, - TYPE_NOT_SET

6.7.19 Set virtual limits

63 Set virtual limits

Command	Description	State change
set_virtualLimits	The command sets the virtual limits of the SLIDEKIT.	When the command is executed successfully, the SLIDEKIT state changes: See the overview below. Client-Format: set_virtualLimits,<lower limit>,<upper limit > Parameters: • Lower limit: FLOAT with one decimal place, minimum position to which the SLIDEKIT can retract in mm (mandatory) • Upper limit: FLOAT with one decimal place, maximum position to which the SLIDEKIT can extend in mm (mandatory)

64 Change in state

Previous status	New status
CONNECTED,VIRTUAL_ LIMITS_NOT_SET	READY,M2R_NOT_STARTED

65 Permitted states: set_virtualLimits

Command	Description
CONNECTED	Connected
READY	Ready

66 Possible confirmations: set_virtualLimits

Message	Description	Response format
OK	Command executed successfully. Limits set successfully.	set_virtualLimits,OK
WNP	Wrong number of parameters specified. The command accepts exactly 2 parameters.	set_virtualLimits,WNP,2,2
OOR	Out of range. One of the specified parameters is not within the limits. The lower limit value must not be less than 0 and the upper limit value must not be greater than the maximum stroke.	set_virtualLimits,OOR,<index of received parameter >,<minimum value>,<maximum value> • Index of received parameter: index of the first received parameter that is out of range • minimum value: the minimum value for this parameter • maximum value: the maximum value for this parameter
NA	Not allowed. The set limit values cannot be executed in the current state.	set_virtualLimits, NA
VE	Value error. Limit values should be of type FLOAT.	virtualLimits,VE,<index of received parameter >,FLOAT Index of received parameter: index of the first parameter with the incorrect type.
EF	Execution failed. Example: The limits cannot be set because homing has not been carried out.	set_virtualLimits,EF,Requested data do not exist:max_stroke Limits cannot be set if the maximum stroke is not known. The maximum stroke is calculated during homing.

6.7.20 Retrieve virtual limits

67 Retrieve virtual limits

Command	Description	State change
get_virtualLimits	The command returns the virtual limit values. The unit of measurement for virtual limit values is mm.	Client format: get_virtualLimits Parameters: none

68 Permitted states: get_virtualLimits

Command	Description
CONNECTED	Connected
READY	Ready
MOVING	Movement

69 Possible confirmations: get_virtualLimits

Message	Description	Response format
OK	Command executed successfully. Virtual limits are returned as a response.	get_virtualLimits,OK, <lower limit >,<upper limit> • Lower limit: virtual minimum to which the SLIDEKIT can retract • Upper limit: virtual maximum to which the SLIDEKIT can extend.
WNP	Wrong number of parameters specified. The command should not contain any parameters.	get_virtualLimits,WNP,0,0
NA	Not allowed. Virtual limits cannot be retrieved in the current state.	get_virtualLimits,NA
EF	Execution failed. • E. g.: The limits cannot be retrieved because they have not been set.	get_virtualLimits,EF, Requested data do not exist: virtual_minimum, Requested data do not exist: virtual_maximum” Retrieve status response: get_status,OK,CONNECTED, VIRTUAL_LIMITS_NOT_SET

7 Maintenance

Maintenance work and repairs may only be carried out by qualified personnel.

1. Disconnect the power supply before performing maintenance and repairs.
2. Ensure that the linear module is not under load or under tension.
3. Use suitable safety equipment.

DANGER

Risk of fatal injury from unauthorized reconnection of the power supply

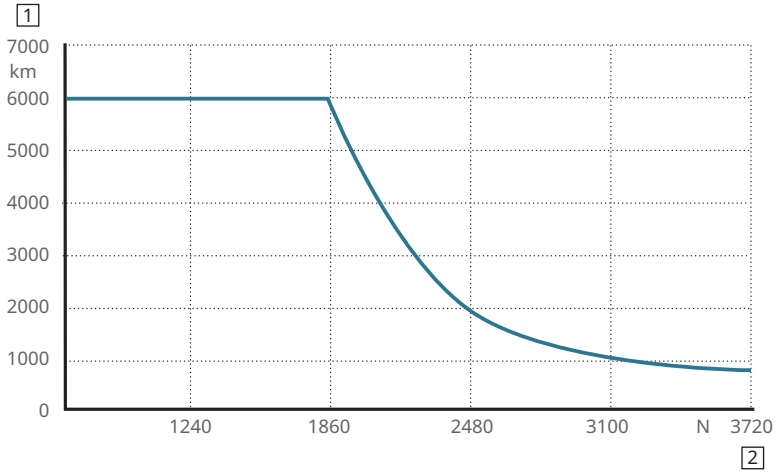
Risk of fatal injury to persons in the hazard area due to moving parts or electric shock if the power supply is switched on without authorization during work on the system and causes the system to restart.

- Before starting work, switch off the system and secure it against reconnection.

7.1 Lubrication intervals

Under normal operating conditions (v ≤ 1 m/s, travel below Fm ≤ 0.3 c.) 4.08 cm^3, inject grease as per the conditions in the diagram below.

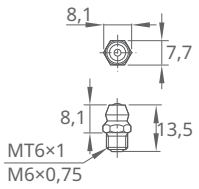
7 Lubrication intervals



1	Lubrication interval	2	Load

Lubrication can be carried out via the special lubricating nipple.

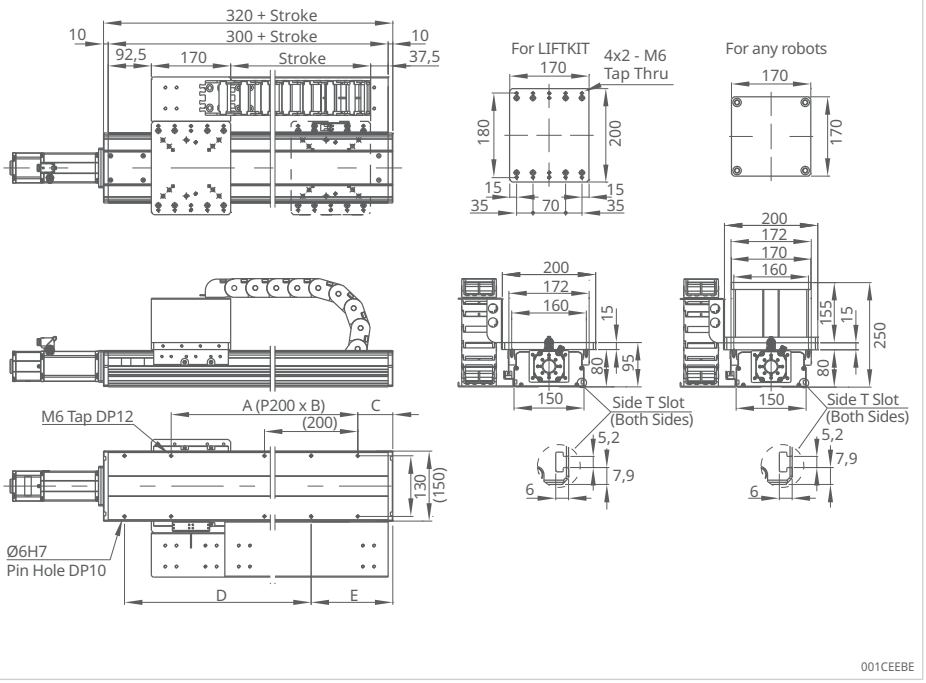
8 Lubricating nipple



8 Technical information

8.1 Version with ball screw

9 Version with ball screw



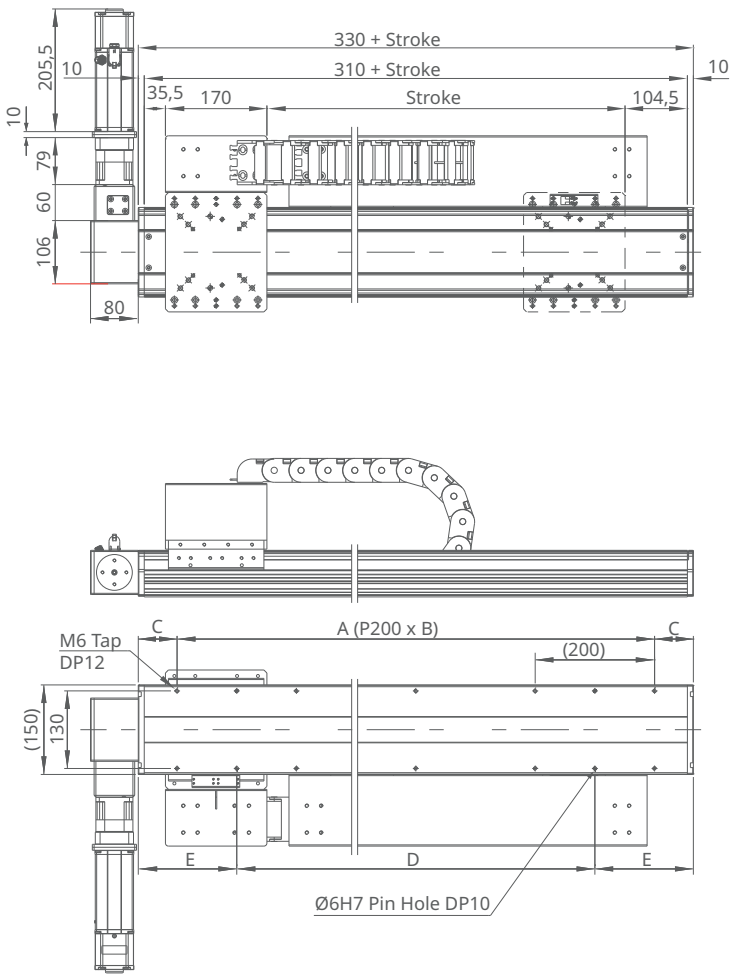
70 Technical information for versions with a ball screw

Point	Stroke	A	B	C	D	E
	mm					
1	100	200	1	75	200	175
2	200	400	2	25		125
3	300	400	2	75	400	175
4	400	600	3	25		125
5	500	600	3	75	600	175
6	600	800	4	25		125
7	700	800	4	75	800	175
8	800	1000	5	25		125
9	900	1000	5	75	1000	175
10 [1)]	1000 [1)]	1200 [1)]	6 [1)]	25 [1)]		125 [1)]
11	1100	1200	6	75	1200	175
12	1200	1400	7	25		125
13	1300	1400	7	75	1400	175
14	1400	1600	8	25		125
15	1500	1600	8	75	1600	175
16	1600	1800	9	25		125
17	1700	1800	9	75	1800	175
18 [1)]	1800 [1)]	2000 [1)]	10 [1)]	25 [1)]		125 [1)]

[1)] Standard stroke

8.2 Version with belt drive

10 Version with belt drive



A	for LIFTKIT	B	for other robots
1	4×2 M6, Tap Thru	2	Side T-slot (on both sides)

71 Technical information for versions with a belt drive

Item	Stroke	A	B	C	D	E
	mm					
10	1000	1200	6	65	1000	165
11	1100	1200	6	115	1000	215
12	1200	1400	7	65	1200	165
13	1300	1400	7	115	1200	215
14	1400	1600	8	65	1400	165
15	1500	1600	8	115	1400	215
16	1600	1800	9	65	1600	165
17	1700	1800	9	115	1600	215
18	1800	2000	10	65	1800	165
19	1900	2000	10	115	1800	215
20	2000	2200	11	65	2000	165
21	2100	2200	11	115	2000	215
22	2200	2400	12	65	2200	165
23	2300	2400	12	115	2200	215
24	2400	2600	13	65	2400	165

Item	Stroke	A	B	C	D	E
	mm					
25 [1]	2500 [1]	2600 [1]	13 [1]	115 [1]	2400 [1]	215 [1]
26	2600	2800	14	65	2600	165
27	2700	2800	14	115	2600	215
28	2800	3000	15	65	2800	165
29	2900	3000	15	115	2800	215
30 [1]	3000 [1]	3200 [1]	16 [1]	65 [1]	3000 [1]	165 [1]

[1] Standard stroke

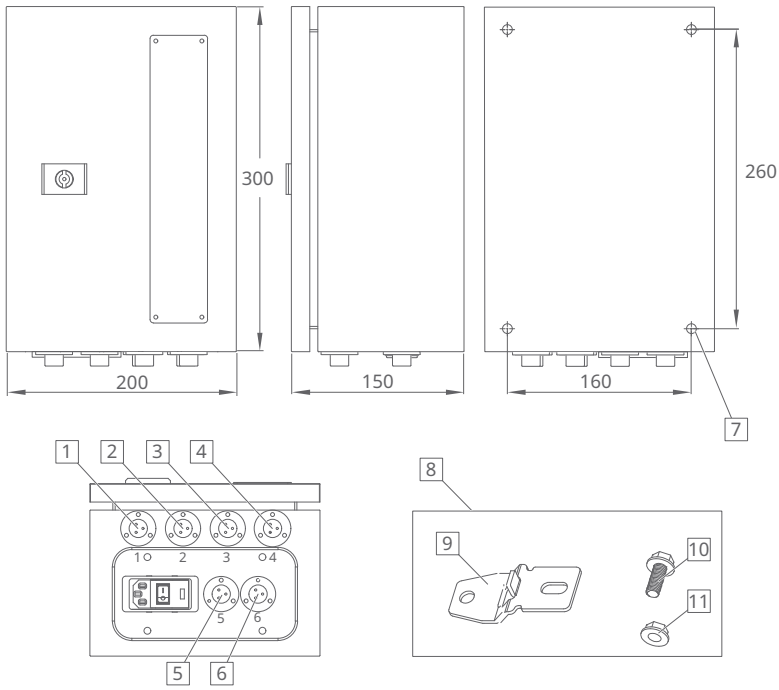
8.3 Compatibility

72 Compatibility with robots

SLIDEKIT	UR	Fanuc	Yaskawa	TM	Doosan	ABB
Attachment plate	On request	On request	On request	On request	On request	On request
S00	UR3	CRX5	HC10	TM5	A0509/s	CRB 15000-5 15000-5
	UR5	CRX10	-	TM12	M0609	4 CRB 15000-10
	UR10	CRX20	-	TM14	M1509	4 CRB 15000-12
	UR16	-	-	TM20	A0912/s M1013 M0617	-

8.4 Controller

11 Controller

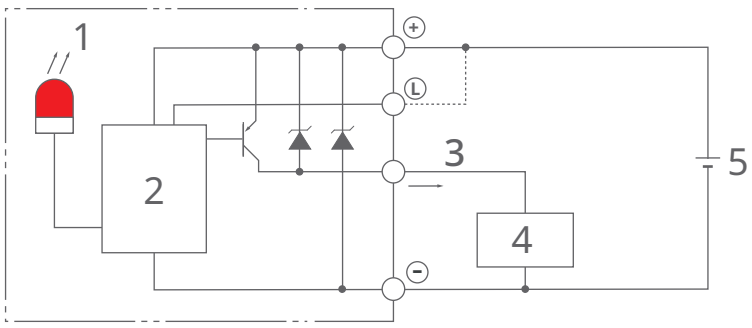


001CDE4B

1	Connection for proximity switch 1 (Home) Connection for proximity switch 2 (+Limits)	2	Connection for safety-I/O cable
3	Connection for cobot interface cable	4	Connection for digital-I/O interface cable
5	CANopen interface connection	6	Motor power cable connection
7	Ø 4–Ø 8.5 (connection opening to SLIDEKIT control box)	8	Wall bracket (optional)
9	Clip: 4EA	10	Screw: 4EA
11	Nut: 4EA		

8.5 Limit switches

12 Limit switches



001CDE30

1	Indicator light (red)	2	Main circuit
3	OUT IC	4	Load
5	DC 5V to 25 V		

73 EE-SX574P-WR 1M

Model	Configuration of the output	Times			End conditions
EE-SX67P EE-SX67P-WR	Lights-ON		Incident		Short circuit between (L) terminal and positive (+) terminal
			Interrupted		
		Indicator light (red)	ON		
			OFF		
	Dark-ON	Output transistor	ON		
			OFF		
		Load (relay)	Operates		
			Releases		
	Lights-ON		Incident		Open between (L) terminal and positive (+) terminal *1 *2
			Interrupted		
		Indicator light (red)	ON		
			OFF		
	Dark-ON	Output transistor	ON		
			OFF		
		Load (relay)	Operates		
			Releases		

1	+	V_{CC}
2	L	L [1)]
3	OUT	OUTPUT
4	–	GND (0 V)

1) Pin 2 is not required for the EE-SX474.

Schaeffler Technologies AG & Co. KG
Georg-Schäfer-Straße 30
97421 Schweinfurt
Germany
www.schaeffler.de/en
info.de@schaeffler.com

In Germany:
Phone 0180 5003872
From other countries:
Phone +49 9721 91-0

BA 143 / 01 / en-US / 2026-03