UNIVERSAL ROBOTS

Software Handbook

PolyScope 5

Original instructions (en)

SW 5.25

The information contained herein is the property of Universal Robots A/S and shall not be reproduced in whole or in part without prior written approval of Universal Robots A/S. The information herein is subject to change without notice and should not be construed as a commitment by Universal Robots A/S. This document is periodically reviewed and revised.

Universal Robots A/S assumes no responsibility for any errors or omissions in this document.

Copyright © 2009-2025 by Universal Robots A/S.

The Universal Robots logo is a registered trademark of Universal Robots A/S.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Contents

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

1. Introduction

Description

This software manual provides the essential information you need to start using your Universal Robots robot.

NOTICE

Before powering on the robot for the first time, please refer to the following sections:

- Read the safety information in the Hardware Description that you can find in the box.
- Set the safety configuration parameters defined by the risk assessment (see 2 Software Safety Configuration on page 32).

All functions for using PolyScope can be found in this manual.
Please refer to the 3rd party provider of any URcaps for their propriety documentation.
The software manual may be used together with the Script Manual, if you are going to create scripts for your robot programs

1.1. Applied Standards

Description

This section describes relevant standards applied to the development and manufacturing of the UR robot, including the robot arm, Control Box and Teach Pendant .
A standard is not a law, but a document developed by stakeholders within a given industry. Standards contain requirements and guidance for a product or product group.
The abbreviations in this manual and their meaning are listed in the table below:

Abbreviations in this document	
ISO	International Organization for Standardization
IEC	International Electrotechnical Commission
EN	European Norm
TS	Technical Specification
TR	Technical Report
ANSI	American National Standards Institute
RIA	Robotic Industries Association (now known as “A3”)
CSA	Canadian Standards Association

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Applied Standards

Maintaining robot compliance with the following standards requires adhering to the assembly instructions, safety instructions and guidance in this manual. For the safety of the robot application, the integrator is required to comply with ISO 10218-2. Unauthorized modifications invalidate the Declaration of Incorporation (DOI), certifications and conformity of the robot.
UR robots comply with the relevant requirements in the applied standards. Applicable test reports and certifications included in this manual and the standards are listed in the Declaration of Incorporation.
The standards applicable to this manual are listed in the table below:

ISO 13849-1
ISO 13849-2

Clause	Description
Safety of machinery - Safety related parts of control systems Part 1: General principles for design Part 2: Validation	The safety control system is designed according to the requirements of these standards. The Safety Functions are certified to these functional safety standards.

ISO 13850

Clause	Description
Safety of machinery - Emergency stop - Principles for design	---

ISO 12100

Clause	Description
Safety of machinery - General principles for design - Risk assessment and risk reduction	---

ISO 10218-1

Clause	Description
Robots and robotic devices - Safety requirements for industrial robots Part 1: Robots	This standard is intended for the robot manufacturer, not the integrator. ISO 10218-2 has the safety requirements associated with the robot system, application and cell. It deals with the design and integration of the robot application.

ANSI/RIA R15.06

Clause	Description
Industrial Robots and Robot Systems - Safety Requirements	This American national standard is a national adoption without deviation of both ISO 10218-1 and ISO 10218-2, combined into one document. The language is changed from British International English to American English, but the technical contents are the same. Part 2 of this standard is intended for the integrator of the robot system / robot application, and not Universal Robots.

CAN/CSA-Z434

Clause	Description
Industrial Robots and Robot Systems - General Safety Requirements	This Canadian national standard is a national adoption of both ISO 10218-1 and ISO 10218-2 combined into one document. CSA added User to clauses within Part 2. Part 2 of this standard is intended for the integrator of the robot system/ robot application, and not Universal Robots.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

CAN/CSA-Z434

Clause	Description
Industrial Robots and Robot Systems - General Safety Requirements	This Canadian national standard is a national adoption of both ISO 10218-1 and ISO 10218-2 combined into one document. CSA added User to clauses within Part 2. Part 2 of this standard is intended for the integrator of the robot system/ robot application, and not Universal Robots.

IEC 61000-6-2 IEC 61000-6-4

Clause	Description
Electromagnetic compatibility (EMC) Part 6-2: Generic standards - Immunity for industrial environments Part 6-4: Generic standards - Emission standard for industrial environments	These standards define requirements for the electrical and electromagnetic disturbances. Conforming to these standards ensures that the UR robots perform well in industrial environments and that they do not disturb other equipment.

IEC 61326-3-1

Clause	Description
Electrical equipment for measurement, control and laboratory use - EMC requirements Part 3-1: Immunity requirements for safety-related systems and for equipment intended to perform safety-related functions (functional safety) - General industrial applications	This standard defines extended EMC immunity requirements for safety-related functions. Conforming to this standard ensures that the safety functions perform even if other equipment exceeds the EMC emission limits defined in the IEC 61000 standards.

IEC 61131-2

Clause	Description
Programmable controllers Part 2: Equipment requirements and tests	Both standard and safety-rated 24V I/Os comply with the requirements of this standard to ensure reliable communication with other PLC systems.

IEC 14118

Clause	Description
Safety of machinery - Prevention of unexpected startup	Safety requirements to prevent an unexpected start and re-start, as a result of power loss or interruption of power.

IEC 60204-1

Clause	Description
Safety of machinery - Electrical equipment of machines Part 1: General requirements	The emergency stop function is designed as a Stop Category 1 according to this standard. Stop Category 1 is a controlled stop with power to the motors to achieve the stop and then removal of power when the stop is achieved.

IEC 60947-5-5

Clause	Description
Low-voltage switchgear and controlgear Part 5-5: Control circuit devices and switching elements - Electrical emergency stop device with mechanical latching function	---

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

IEC 60529

Clause	Description
Degrees of protection provided by enclosures (IP Code)	This standard defines enclosure ratings regarding protection against dust and water.

IEC 60320-1

Clause	Description
Appliance couplers for household and similar general purposes Part 1: General requirements	The mains input cable complies with this standard.

ISO 9409-1

Clause	Description
Manipulating industrial robots - Mechanical interfaces Part 1: Plates	The tool flange on UR robots conforms to a type according to this standard. Robot tools (end-effectors) should also be constructed according to the same type to ensure proper fitting to the mechanical interface of the specific UR robot.

ISO 13732-1

Clause	Description
Ergonomics of the thermal environment - Methods for the assessment of human responses to contact with surfaces Part 1: Hot surfaces	---

IEC 61140

Clause	Description
Protection against electric shock - Common aspects for installation and equipment	A protective earth/ground connection is mandatory, as defined in the Part I Hardware Installation Manual.

IEC 60068-2-1IEC 60068-2-2IEC 60068-2-27IEC 60068-2-64

Clause	Description
Environmental testing Part 2-1: Tests - Test A: Cold Part 2-2: Tests - Test B: Dry heat Part 2-27: Tests - Test Ea and guidance: Shock Part 2-64: Tests - Test Fh: Vibration, broadband random and guidance	---

IEC - 61784-3

Clause	Description
Industrial communication networks - Profiles Part 3: Functional safety fieldbuses - General rules and profile definitions	---

IEC 61784-3

Clause	Description
Safety of machinery - Electrical equipment of machines Part 1: General requirements	---

IEC 60664-1
IEC 60664-5

Clause	Description
Insulation coordination for equipment within low-voltage systems Part 1: Principles, requirements and tests Part 5: Comprehensive method for determining clearances and creepage distances equal to or less than 2 mm	---

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

EUROMAP 67:2015, V1.11

Clause	Description
Electrical Interface between Injection Molding Machine and Handling Device / Robot	The E67 accessory module, that interfaces with injection molding machines, complies with this standard.

1.2. Joint Description

Description

The Universal Robots robot arm is composed of tubes and joints. You use the PolyScope to coordinate the motion of these joints to move the robot arm. You attach tools to end of the robot arm, or Tool Flange . Moving the robot arm positions the tool. You cannot position the tool directly above, or directly below the Base.

- **Base**: where the robot is mounted.
- **Shoulder** and **Elbow**: make larger movements.
- **Wrist 1** and **Wrist 2**: make finer movements.
- **Wrist 3**: where the tool is attached to the Tool Flange.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

1.3. Robot Arm Installation

Description Install and power on the robot arm and Control Box to start using PolyScope.

Assemble the robot You have to assemble the robot arm, Control Box and Teach Pendant to be able to continue.

1. Unpack the robot arm and the Control Box.
2. Mount the robot arm on a sturdy, vibration-free surface.
3. Place the Control Box on its Foot.
4. Connect the robot cable to the robot arm and the Control Box.
5. Plug in the mains, or main power cable, of the Control Box.

WARNING

Failure to secure the robot arm to a sturdy surface can lead to injury caused by the robot falling.

- Ensure the robot arm is secured to a sturdy surface

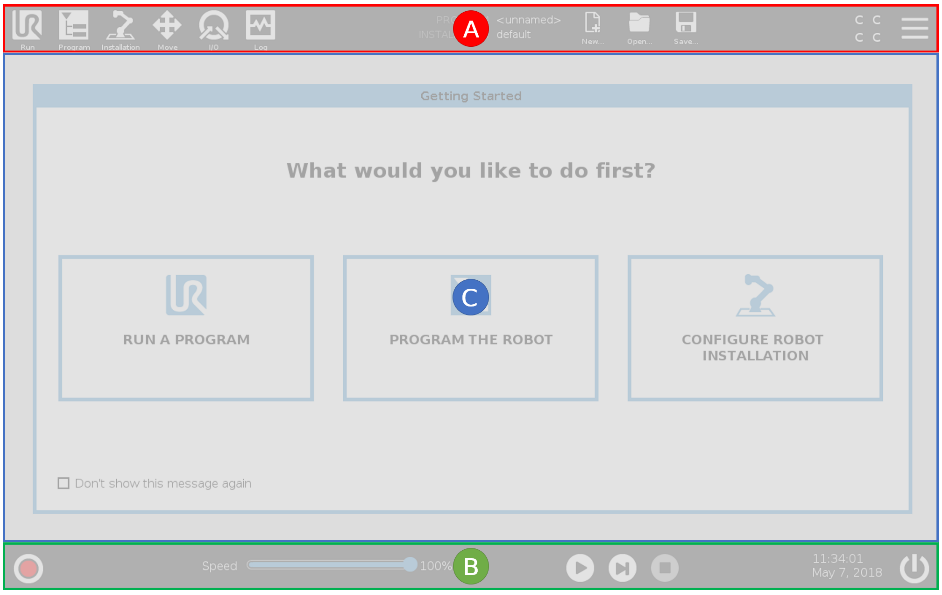
Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

1.4. PolyScope Overview

Description

PolyScope is the Graphical User Interface (GUI) on the **Teach Pendant** that operates the robot arm via a touch screen. You create, load and execute programs for the robot in PolyScope. The PolyScope interface is divided as shown in the following illustration:

- A: **Header** with icons/tabs that make interactive screens available to you.
- B: **Footer** with buttons that control your loaded program/s.
- C: **Screen** with fields and options to manage and monitor robot actions.



Using the Touch Screen

The touch sensitivity is designed to avoid false selections on PolyScope, and to prevent unexpected motion of the robot.

The Teach Pendant touch screen is optimized for use in industrial environments. Unlike consumer electronics, Teach Pendant touch screen sensitivity is, by design, more resistant to environmental factors such as:

- water droplets and/or machine coolant droplets
- radio wave emissions
- other conducted noise from the operating environment.

For best results, use the tip of your finger to make a selection on the screen.
In this manual, this is referred to as a "tap".
A commercially available stylus may be used to make selections on the screen if desired.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

1.4.1. Icons/Tabs On PolyScope

Description

The following section lists and defines the icons/tabs and buttons in the PolyScope interface.

Header Icons / Functions

Run is a simple means of operating the robot using pre-written programs.

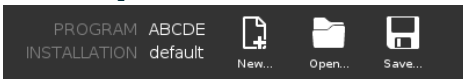
Program creates and/or modifies robot programs.

Installation configures robot arm settings and external equipment e.g. mounting and safety.

Move controls and/or regulates robot movement.

I/O monitors and sets live Input/Output signals to and from robot control box.

Log indicates robot health as well as any warning or error messages.



Program and Installation Manager selects and displays active program and installation. The Program and Installation Manager includes: File Path, New, Open and Save.

New... creates a new Program or Installation.

Open... opens a previously created and saved Program or Installation.

Save... saves a Program, Installation or both at the same time.

Operational modes

Automatic indicates the operational mode of the robot is set to Automatic. Tap it to switch to the Manual operational mode.

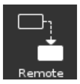
Manual indicates the operational mode of the robot is set to Manual. Tap it to switch to the Automatic operational mode.

Remote Control

The Local mode and Remote mode icons only become accessible if you enable Remote Control.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Local indicates the robot can be controlled locally. Tap it to switch to Remote control.



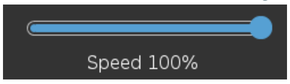
Remote indicates the robot can be controlled from a remote location. Tap it to switch to Local control.

Safety Checksum displays the active safety configuration.

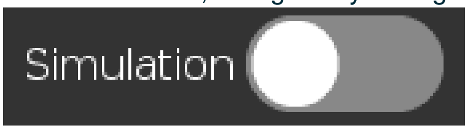

Hamburger Menu accesses PolyScope Help, About and Settings.

Footer Icons / Functions


Initialize manages robot state. When RED, press it to make the robot operational.

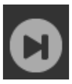


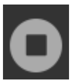
Speed Slider shows in real time the relative speed at which the robot arm moves, taking safety settings into account.



Simulation button toggles a program execution between Simulation Mode and the Real Robot. When running in Simulation Mode, the Robot Arm does not move. Therefore, the robot cannot damage itself or nearby equipment in a collision. If you are unsure what the Robot Arm will do, use Simulation Mode to test programs.


Play starts current loaded robot Program.


Step allows a Program to be run single-stepped.


Stop halts current loaded robot Program.

High Speed Manual Mode

The High Speed Manual hold-to-run function is only available in manual mode when a Three-Position Enabling Device is configured.

High Speed Manual Modeallows both tool speed and elbow speed to temporarily exceed 250mm/s.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

1.5. Freedrive

Description

Freedrive allows the robot arm to be manually pulled into desired positions
For most robot sizes, the most typical way to enable Freedrive is to press the Freedrive button on the Teach Pendant. More ways to enable and use Freedrive are described in the following sections.
In Freedrive, the robot arm joints move with little resistance because the brakes are released. Resistance increases as the robot arm in Freedrive approaches a predefined limit or plane. This makes pulling the robot into position feel heavy.

WARNING
Injury to personnel can occur due to unexpected motion.

- Verify the configured payload is the payload being used.
- Verify the correct payload is securely attached to the tool flange.

Enabling Freedrive

You can enable Freedrive in the following ways:

- Use the 3PE Teach Pendant.
- Use the Freedrive on robot.
- Use I/O Actions.

NOTICE
Enabling Freedrive while you are moving the robot arm, can cause it to drift leading to faults.

- Do not enable Freedrive while you are pushing or touching the robot.

3PE Teach Pendant

To use the 3PE TP button to freedrive the robot arm:

1. Rapidly light-press, release, light-press again and keep holding the 3PE button in this position.

Now you can pull the robot arm into a desired position, while the light-press is maintained.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Freedrive on robot

To use Freedrive on robot to freedrive the robot arm:

1. Press-and-hold the button of switch configured for **Freedrive on robot**.
2. When the Freedrive panel appears in PolyScope, select the desired movement type for the robot arm's joints. Or use the list of axes to customize the movement type.
3. You can define the type of feature if required, by selecting an option from the Feature dropdown list.

   The robot arm can stop moving if it approaches a singularity scenario. Tap **All axes are free** in the Freedrive panel to resume movement.
4. Move the robot arm as desired.

Backdrive

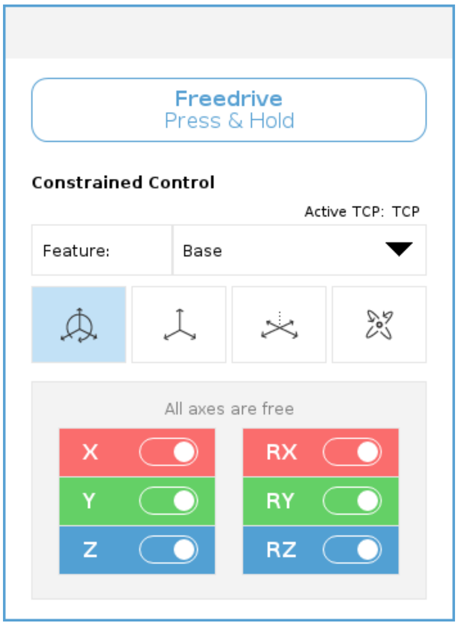
During initialization of the robot arm, minor vibrations may be observed when the robot brakes are released. In some situations, such as when the robot is close to collision, these vibrations are undesirable. Use Backdrive to force specific joints to a desired position without releasing all brakes in the robot arm.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

1.5.1. Freedrive Panel

Description

When the robot arm is in Freedrive, a panel appears on PolyScope, as illustrated below.



To access the Freedrive Panel

1. In the Header, tap the Move tab.
2. At the bottom of the screen, tap Freedrive.

   The Freedrive Panel opens.
3. Press and hold the Freedrive button inside the Panel.

   You can the robot arm manually, similar to pressing the Freedrive button located on the Teach Pendant.

A LED indicates when the robot arm approaches a singularity position. The LED is detailed in the following section.

LED in Freedrive panel

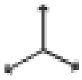
The LED on the status bar of the Freedrive panel indicates:

- When one or more joints are approaching their joint limits.
- When the robot arm's positioning is approaching singularity. Resistance increases as the robot approaches singularity, making it feel heavy to position.

Freedrive Panel icons

You can lock one or more of the axes allowing the TCP to move in a particular direction, as defined in the table below.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

 All axes are free	Movement is allowed through all axes.
 Plane	Movement is only allowed through the X-axis and Y-axis.
 Translation	Movement is allowed through all axes, without rotation.
 Rotation	Movement is allowed through all axes, in a spherical motion, around the TCP.

CAUTION

Moving the robot arm in some axes when a tool is attached, can present a pinch point.

- Use caution when moving the robot arm in any axis.

1.6. Backdrive

Description

Backdrive is a Manual Mode used to force specific joints to a desired position without releasing all brakes in the robot arm.
This is sometimes necessary if the robot arm is close to collision and the vibrations that accompany a full restart are not desired.
The robot joints feel heavy to move, while Backdrive is in use.

You can use any of the following sequences to enable Backdrive:

- 3PE Teach Pendant
- 3PE device/switch
- Freedrive on robot

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

3PE Teach Pendant

To use the 3PE TP button to backdrive the robot arm.

1. On the Initialize screen, tap **ON** to start the power up sequence.
2. When the robot state is **Teach Pendant 3PE Stop**, light-press, then light-press-and-hold, the 3PE TP button.
   The robot state changes to **Backdrive**.
3. Now you can apply significant pressure to release the brake in a desired joint to move the robot arm.
   As long as light-press is maintained on the 3PE button, Backdrive is enabled, allowing the arm to move.

3PE device/switch

To use a 3PE device/switch to backdrive the robot arm.

1. On the Initialize screen, tap **ON** to start the power up sequence.
2. When the robot state is **Teach Pendant 3PE Stop**, light-press, then light-press-and-hold, the 3PE TP button.
   The robot state changes to **System 3PE Stop**.
3. Press and hold the 3PE device/switch.
   The robot state changes to **Backdrive**.
4. Now you can apply significant pressure to release the brake in a desired joint to move the robot arm.
   As long as the hold is maintained on both the 3PE device/switch and the 3PE TP button, Backdrive is enabled, allowing the arm to move.

Freedrive on robot

To use Freedrive on robot to backdrive the robot arm.

1. On the Initialize screen, tap **ON** to start the power up sequence.
2. When the robot state is **Teach Pendant 3PE Stop**, press and hold the **Freedrive on robot**.
   The robot state changes to **Backdrive**.
3. Now you can apply significant pressure to release the brake in a desired joint to move the robot arm.
   As long as the hold is maintained on the Freedrive on robot, Backdrive is enabled, allowing the arm to move.

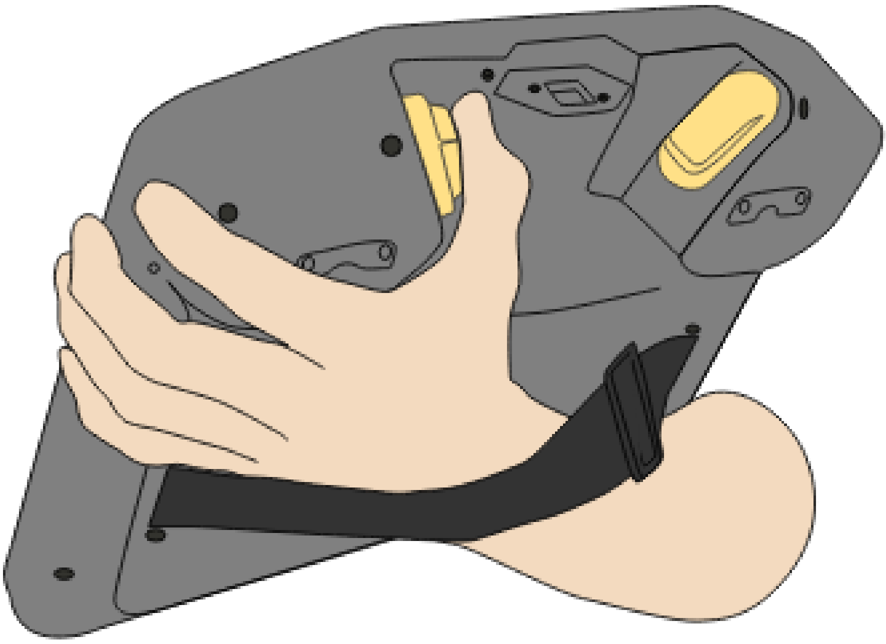
Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

1.6.1. Backdrive Inspection

Description

If the robot is close to colliding with something, you can use Backdrive to move the robot arm to a safe position before initializing.

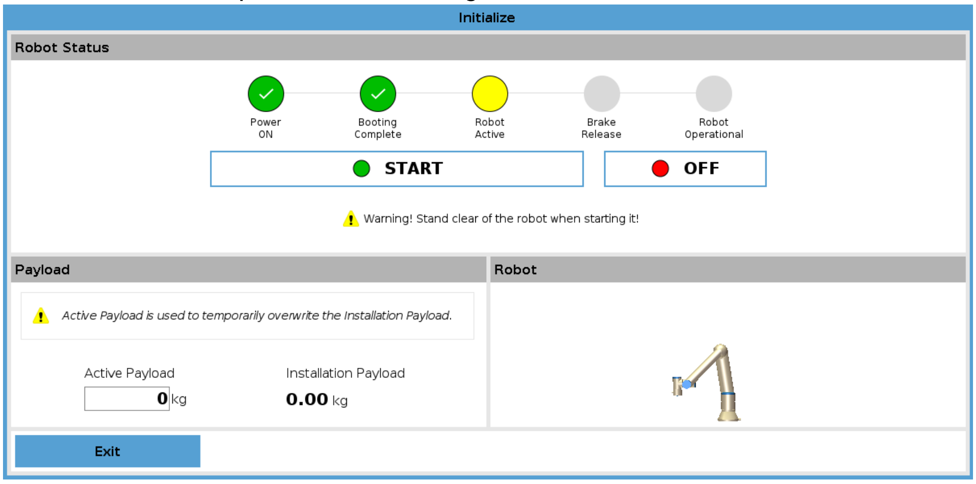
3PE Teach Pendant



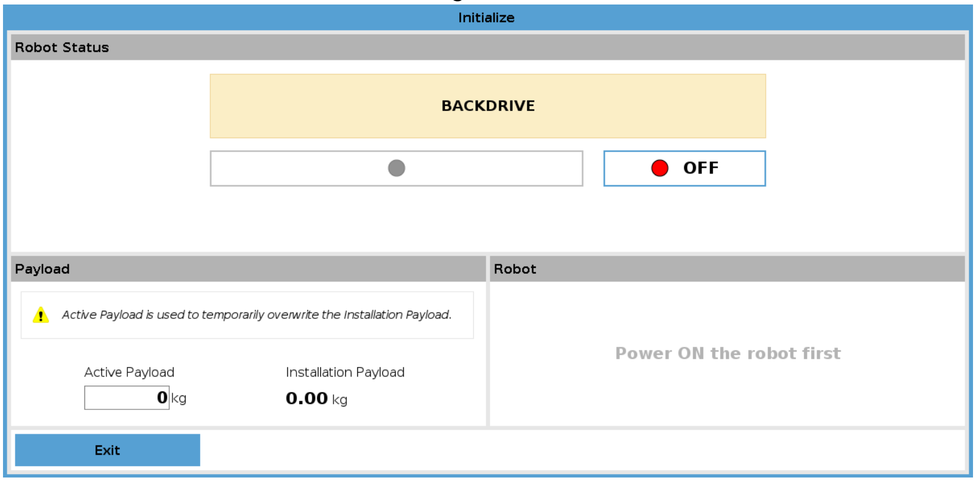
Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Enable Backdrive

1. Press ON to enable power. Status changes to *Robot Active*



2. Press and hold Freedrive. Status changes to *Backdrive*



3. Move robot as in Freedrive mode. Joint brakes are released where needed once the Freedrive button is activated.

NOTICE

In Backdrive Mode the robot is “heavy” to move around.

MANDATORY ACTION

You must test Backdrive mode on all joints.

Safety settings

Verify the robot safety settings comply with the robot installation risk assessment.

Additional safety inputs and outputs are still functioning

Check which safety inputs and outputs are active and that they can be triggered via PolyScope or external devices.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

1.7. Quick System Start-up

Quick System Start

> **MANDATORY ACTION**
>
> Before using the PolyScope, verify that the robot arm and Control Box are correctly installed.

This is how you quickly start up the robot.

1. On the **Teach Pendant**, press the emergency stop button.
2. On the Teach Pendant, press the power button and allow the system to start, displaying text on the **PolyScope**.
3. A popup appears on the touch screen indicating that the system is ready and that the robot must be initialized.
4. In the popup dialog, tap **Go to Initialize Screen** to access the Initialize screen.
5. Unlock the emergency stop button to change the robot state from **Emergency Stopped** to **Power off**.
6. Step outside the reach (workspace) of the robot.
7. On the **Initialize Robot** screen, tap the **ON** button and allow the robot state to change to **Idle**.
8. In the **Payload** field, in **Active Payload**, verify the payload mass. You can also verify that the mounting position is correct, in the **Robot** field.
9. Tap the **Start** button, for the robot to release its brake system. The robot vibrates and makes clicking sounds, indicating it is ready to be programmed.

> **NOTICE**
>
> Learn to program your Universal Robots robot on www.universal-robots.com/academy/

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

1.8. The First Program

Description

A program is a list of commands telling the robot what to do. For most tasks, programming is done entirely using the PolyScope software. PolyScope allows you to teach the robot arm how to move using a series of waypoints to set up a path for the robot arm to follow.

Use the Move tab to move the Robot Arm to a desired position, or teach the position by pulling the Robot Arm into place while holding down the Freedrive button at the top of the Teach Pendant.

You can create a program can to send I/O signals to other machines at certain points in the robot's path, and perform commands like **if…then** and **loop**, based on variables and I/O signals.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

To create a simple program

This is a simple example program, to show how easy it is to use a UR robot. It assumes a harmless environment and a very careful user. Do not increase the speed or acceleration above the default values. Always conduct a risk assessment before placing the robot into operation.

1. On PolyScope, in the Header **File Path**, tap **New...** and select **Program**.
2. Under Basic, tap **Waypoint** to add a waypoint to the program tree. A default MoveJ is also added to the program tree.
3. Select the new waypoint and in the Command tab, tap **Waypoint**.
4. On the Move Tool screen, move the robot arm by pressing the move arrows.
   You can also move the robot arm by holding down the Freedrive button and pulling the Robot Arm into desired positions.
5. Once the robot arm is in position, press **OK** and the new waypoint displays as Waypoint_1.
6. Follow steps 2 to 5 to create Waypoint_2.
7. Select Waypoint_2 and press the Move Up arrow until it is above Waypoint_1 to change the order of the movements.
8. Stand clear, hold on to the emergency stop button and in the PolyScope Footer, press **Play** button for the Robot Arm to move between Waypoint_1 and Waypoint_2.
   Congratulations! You have now produced your first robot program that moves the Robot Arm between the two given waypoints.

NOTICE

A singularity position can prevent the robot arm from moving into many poses/orientations and can block robot arm movement altogether.

- Avoid placing the robot arm into a singularity position

You can find more detailed information in the section on Singularity.

NOTICE

Do not drive the robot into itself or anything else as this may cause damage to the robot.

WARNING

Keep your head and torso outside the reach (workspace) of the robot. Do not place fingers where they can be caught.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

1.9. Cybersecurity Threat Assessment

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

1.9.1. General Cybersecurity

Description

Connecting a Universal Robots robot to a network can introduce cybersecurity risks. These risks can be mitigated by using qualified personnel and implementing specific measures for protecting the robot's cybersecurity.
Implementing cybersecurity measures requires conducting a cybersecurity threat assessment.
The purpose is to:

- Identify threats
- Define trust zones and conduits
- Specify the requirements of each component in the application

WARNING
Failure to conduct a cybersecurity risk assessment can place the robot at risk.

- The integrator or competent, qualified personnel shall conduct a cybersecurity risk assessment.

NOTICE
Only competent, qualified personnel shall be responsible for determining the need for specific cybersecurity measures and for providing the required cybersecurity measures.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

1.9.2. Cybersecurity Requirements

Description

Configuring your network and securing your robot requires you to implement the threat measures for cybersecurity.
Follow all the requirements before you start configure your network, then verify the robot setup is secure.

Cybersecurity

- Operating personnel must have a thorough understanding of general cybersecurity principles and advanced technologies as used in the UR robot.
- Physical security measures must be implemented to allow only authorized personnel physical access to the robot.
- There must be adequate control of all access points. For example: locks on doors, badge systems, physical access control in general.

WARNING

Connecting the robot to a network that is not properly secured, can introduce security and safety risks.

- Only connect your robot to a trusted and properly secured network.

Network configuration requirements

- Only trusted devices are to be connected to the local network.
- There must be no inbound connections from adjacent networks to the robot.
- Outgoing connections from the robot are to be restricted to allow the smallest relevant set of specific ports, protocols and addresses.
- Only URCaps and magic scripts from trusted partners can be used, and only after verifying their authenticity and integrity

Robot setup security requirements

- Change the default password to a new, strong password.
- Disable the "Magic Files" when not actively used (PolyScope 5).
- Disable SSH access when not needed. Prefer key-based authentication over password-based authentication
- Set the robot firewall to the most restrictive usable settings and disable all unused interfaces and services, close ports and restrict IP addresses
-

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

1.9.3. Cybersecurity Hardening Guidelines

Description

Although PolyScope includes many features for keeping the network connection secure, you can harden security by observing to following guidelines:

- Before connecting your robot to any network, always change the default password to a strong password.

> **NOTICE**
> You cannot retrieve or reset a forgotten or lost password.
> - Store all passwords securely.

- Use the built-in settings to restrict the network access to the robot as much as possible.
- Some communication interfaces have no method of authenticating and encrypting communication. This is a security risk. Consider appropriate mitigating measures, based on your cybersecurity threat assessment.
- SSH tunneling (Local port forwarding) must be used to access robot interfaces from other devices if the connection crosses the trust zone boundary.
- Remove sensitive data from the robot before it is decommissioned. Pay particular attention to the URCaps and data in the program folder.
  - To ensure secure removal of highly sensitive data, securely wipe or destroy the SD card.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

1.10. Modes

Description

You access and activate different modes using Teach Pendant or the Dashboard Server. If an external mode selector is integrated, it control the modes - not PolyScope or the Dashboard Server.

Automatic Mode Once activated, the robot can only execute a program of pre-defined tasks. You cannot modify or save programs and installations.
Manual Mode Once activated, you can program the robot. You can modify and save programs and installations.

The speeds used in Manual Mode must be limited to prevent injury. When the robot is operating in Manual Mode, a person could be positioned within reach of the robot. The speed must be limited to the value that is appropriate for the application risk assessment.

WARNING
Injury can occur if the speed used, while the robot is operating in Manual Mode, is too high.

High Speed Manual Mode can be used. It allows both tool speed and elbow speed to temporarily exceed 250 mm/s, while a hold-to-run is used.
Hold-to-run is performed by continuous contact with the Speed Slider.

The robot performs a Safeguard Stop in Manual mode, if a Three-Position Enabling Device is configured, and either released (not pressed) or it is fully compressed.

Switching between Automatic mode to Manual mode requires the Three-Position Enabling Device to be fully released and pressed again to allow the robot to move.
When using High Speed Manual Mode, use safety joint limits or safety planes to restrict the robot's moving space.

Mode switching

Operational mode	Manual	Automatic
Freedrive	x	*
Move robot with arrows on Move Tab	x	*
Edit & save program & installation	x	
Execute Programs	Reduced speed**	*
Start program from selected node	x	
*Only when no Three-Position Enabling Device is configured. ** If a Three-Position Enabling Device is configured, the robot operates at Manual Reduced Speed unless High Speed Manual Mode is activated.		

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

> **WARNING**
> - Any suspended safeguards must be returned to full functionality before selecting Automatic Mode.
> - Wherever possible, Manual Mode shall only be used with all persons located outside the safeguarded space.
> - If an external mode selector is used, it must be placed outside the safeguarded space.
> - No-one is to enter, or be within, the safeguarded space in Automatic Mode, unless safeguarding is used or the collaborative application is validated for power and force limiting (PFL).

Three-Position Enabling Device

When a Three-Position Enabling Device is used and the robot is in Manual Mode, movement requires pressing the Three-Position Enabling Device to the center-on position. The Three-Position Enabling Device has no effect in Automatic Mode.

> **NOTICE**
> - Some UR robot sizes might not be equipped with a Three-Position Enabling Device. If the risk assessment requires the enabling device, a 3PE Teach Pendant must be used.

A 3PE Teach Pendant (3PE TP) is recommended for programming. If another person can be within the safeguarded space when in Manual Mode, an additional device can be integrated and configured for the additional person's use.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

2. Software Safety Configuration

Description

This section covers how to access the robot safety settings. It is made up of items that help you set up the robot Safety Configuration.

WARNING

Before you configure your robot safety settings, your integrator must conduct a risk assessment to guarantee the safety of personnel and equipment around the robot. A risk assessment is an evaluation of all work procedures throughout the robot lifetime, conducted in order to apply correct safety configuration settings. You must set the following in accordance with the risk assessment.

1. The integrator must prevent unauthorized persons from changing the safety configuration e.g. installing password protection.
2. Use and configuration of the safety-related functions and interfaces for a specific robot application.
3. Safety configuration settings for set-up and teaching before the robot arm is powered on for the first time.
4. All safety configuration settings accessible on this screen and sub-tabs.
5. The integrator must ensure that all changes to the safety configuration settings comply with the risk assessment.

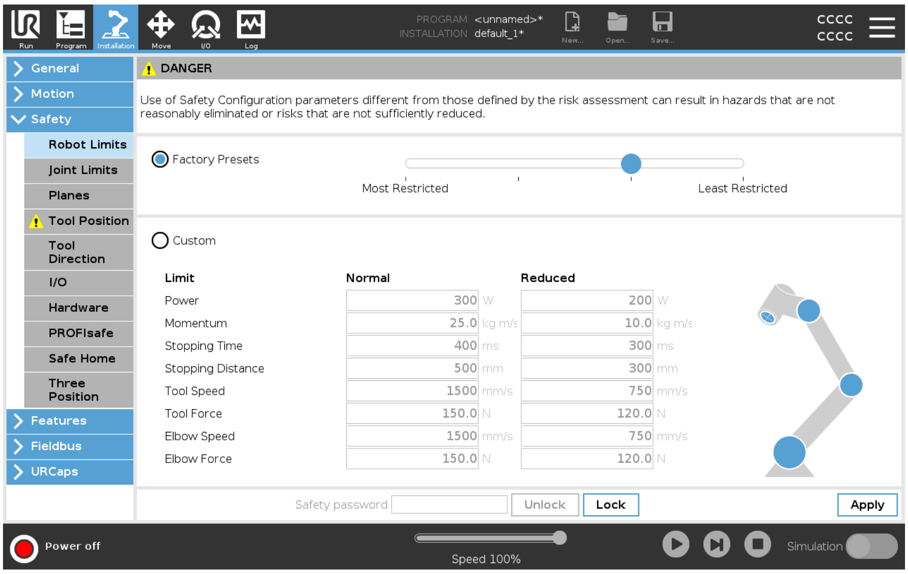
Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Accessing Software Safety Settings

Safety Settings are password protected and can only be configured once a password is set and subsequently used.

To access the software safety settings

1. In your PolyScope header, tap the **Installation** icon.
2. In the Side Menu on the left of the screen, tap **Safety**.
3. Observe that the **Robot Limits** screen displays, but settings are inaccessible.
4. If a **Safety password** was previously set, enter the password and press **Unlock** to make settings accessible. Note: Once Safety settings are unlocked, all settings are now active.
5. Press **Lock** tab or navigate away from the Safety menu to lock all Safety item settings again.



Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

2.1. Setting a Software Safety Password

Description

You must set a password to Unlock all safety settings that make up your Safety Configuration. If no safety password is applied, you are prompted to set it up.

To set a Software Safety password

You can tap the **Lock** tab to lock all Safety settings again or simply navigate to a screen outside of the Safety menu.

1. In your PolyScope header right corner, press the **Hamburger** menu and select **Settings**.
2. On the left of the screen, in the blue menu, press **Password** and select **Safety**.
3. In **New password**, type a password.
4. Now, in **Confirm new password**, type the same password and press **Apply**.
5. In the bottom left of the blue menu, press **Exit** to return to previous screen.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

2.2. Changing the Software Safety Configuration

Description	Changes to the Safety Configuration settings must comply with the risk assessment conducted by the integrator.
Recommended procedure for the integrator:	To change the safety configuration 1. Verify that changes comply with the risk assessment conducted by the integrator. 2. Adjust safety settings to the appropriate level defined by the risk assessment conducted by the integrator. 3. Verify that the settings are applied. 4. Place following text in the operators' manuals: Before working near the robot, make sure that the safety configuration is as expected. This can be verified e.g. by inspecting the Safety Checksum in the top right corner of PolyScope for any changes.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

2.3. Applying a New Software Safety Configuration

Description

The robot is powered off while you make changes to the configuration.
Your changes only take effect after you tap the **Apply** button.
The robot cannot be powered on again until you select **Apply and Restart** to visually inspect your robot Safety Configuration which, for safety reasons, is displayed in SI Units in a popup.
You can select **Revert Changes** to return to the previous configuration. When your visual inspection is complete you can select **Confirm Safety Configuration** and the changes are automatically saved as part of the current robot installation.

2.3.1. Safety Checksum

Description

The **Safety Checksum** icon displays your applied robot safety configuration.

It could be four or eight digits.
A four-digit Checksum should be read from top to bottom and left to right, while an eight-digit Checksum is read left to right, top row first. Different text and/or colors indicate changes to the applied safety configuration.

The **Safety Checksum** changes if you change the **Safety Functions** settings, because the **Safety Checksum** is only generated by the safety settings.
You must apply your changes to the **Safety Configuration** for the **Safety Checksum** to reflect your changes.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

2.4. Safety Configuration without Teach Pendant

Description

You can use the robot without attaching the Teach Pendant. Removing the Teach Pendant requires defining another Emergency Stop source. You must specify if the Teach Pendant is attached to avoid triggering a safety violation.

> **CAUTION**
> If the Teach Pendant is detached or disconnected from the robot, the Emergency Stop button is no longer active. You must remove the Teach Pendant from the vicinity of the robot.

To safely remove the Teach Pendant

The robot can be used without PolyScope as the programming interface.
To configure the robot without a Teach Pendant

1. In the Header tap **Installation**.
2. In the Side Menu on left tap **Safety** and select **Hardware**.
3. Input Safety password and **Unlock** the screen.
4. Deselect **Teach Pendant** to use robot without PolyScope interface.
5. Press **Save and restart** to implement changes.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

2.5. Software Safety Modes

Description

Under normal conditions, i.e. when no protective stop is in effect, the safety system operates in a Safety Mode associated with a set of safety limits.

- **Normal** is the safety configuration that is active by default
- **Reduced** is the safety configuration that is active when the robot **Tool Center Point** (TCP) is positioned beyond a Trigger Reduced plane, or when triggered using a configurable input.
- **Recovery mode** activates when a safety limit from the active limit set is violated, the robot arm performs a Stop Category 0.

  If an active safety limit, such as a joint position limit or a safety boundary, is in violation when the robot arm is powered on, the robot arm starts up in recovery mode. This makes it possible to move the robot arm back within the safety limits.

  In Recovery mode, the movement of the robot arm is restricted by a fixed limit that you cannot customize.

WARNING

Limits for **joint position**, **tool position** and **tool orientation** are disabled in Recovery mode, so take caution when moving the robot arm back within the limits.

The menu of the Safety Configuration screen enables the user to define separate sets of safety limits for both configurations: Normal and Reduced. For the tool and joints,reduced limits for speed and momentum are required to be more restrictive than their Normal mode counterparts.

To Switch Modes: PolyScope

1. In the Header, select the profile icon.
   - **Automatic** indicates the operational mode of the robot is set to Automatic.
   - **Manual** indicates the operational mode of the robot is set to Manual.

Using the Dashboard Server

1. Connect to the Dashboard server.
2. Use the **Set Operational Mode** commands.
   - Set Operational Mode Automatic
   - Set Operational Mode Manual
   - Clear Operational Mode

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

2.6. Software Safety Limits

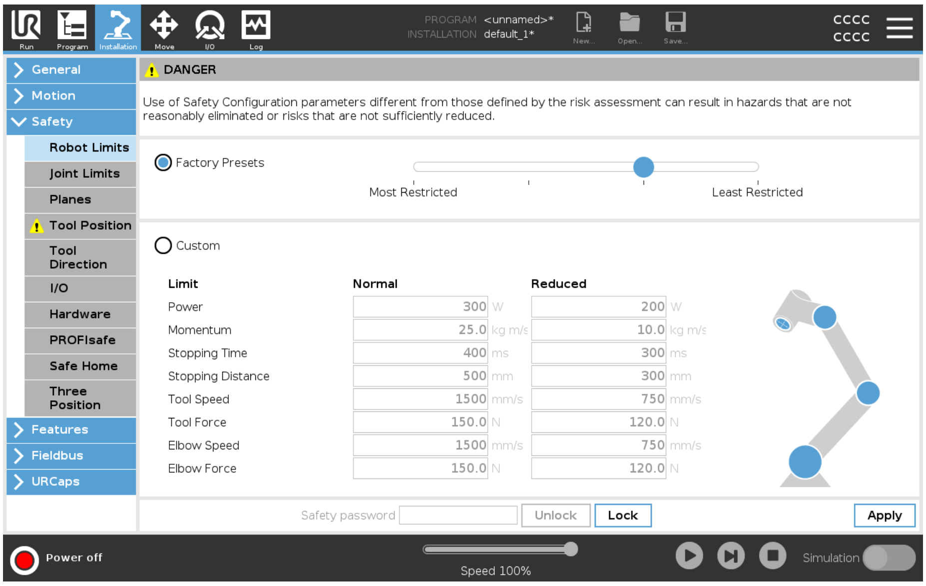
Description The safety system limits are defined in the Safety Configuration . The safety system receives values from the input fields and detects any violation if any the values are exceeded. The robot controller prevents violations by making a robot stop or by reducing the speed.

2.6.1. Robot Limits

Description Robot Limits restrict general robot movements. The Robot Limits screen has two configuration options: **Factory Presets** and **Custom**.

Factory Presets Factory Presets is where you can use the slider to select a predefined safety setting . The values in the table are updated to reflect the preset values ranging from **Most Restricted** to **Least Restricted**

> **NOTICE**
> Slider values are only suggestions and do not substitute a proper risk assessment.



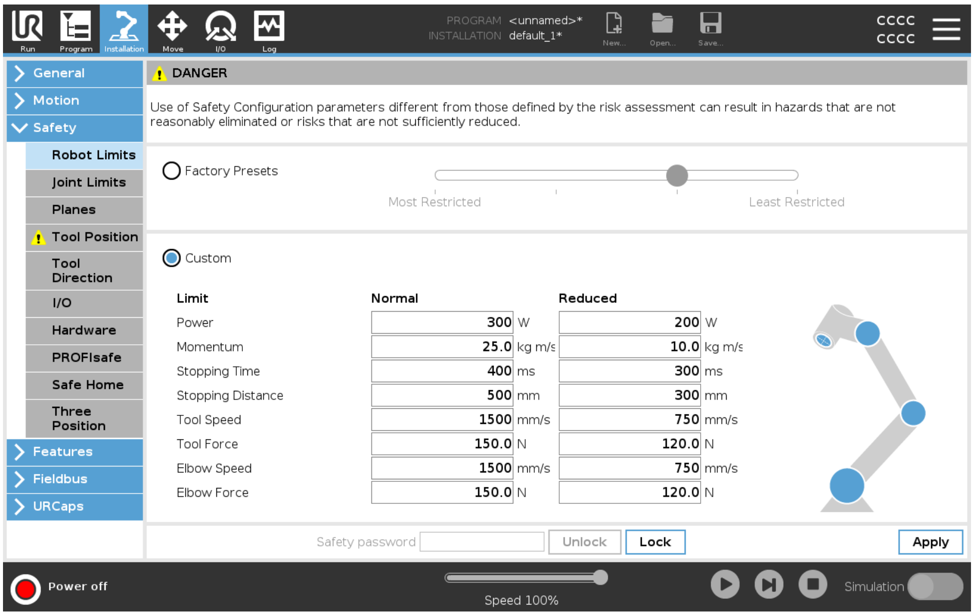
Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Custom Custom is where you can set Limits on how the robot functions and monitor the associated Tolerance.

Power	Limits maximum mechanical work produced by the robot in the environment. This limit considers the payload a part of the robot and not of the environment.
Momentum	Limits maximum robot momentum.
Stopping Time	Limits maximum time it takes the robot to stop e.g. when an emergency stop is activated.
Stopping Distance	Limits maximum distance the robot tool or elbow can travel while stopping. **NOTICE** Restricting stopping time and distance affect overall robot speed. For example, if stopping time is set to 300 ms, the maximum robot speed is limited allowing the robot to stop within 300 ms.
Tool Speed	Limits maximum robot tool speed.
Tool Force	Limits maximum force that the robot tool exerts on the environment to prevent clamping situations.
Elbow Speed	Limits maximum robot elbow speed.
Elbow Force	Limits maximum force that the elbow exerts on the environment to prevent clamping situations.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

The tool speed and force are limited at the tool flange and the center of the two user-defined tool positions.

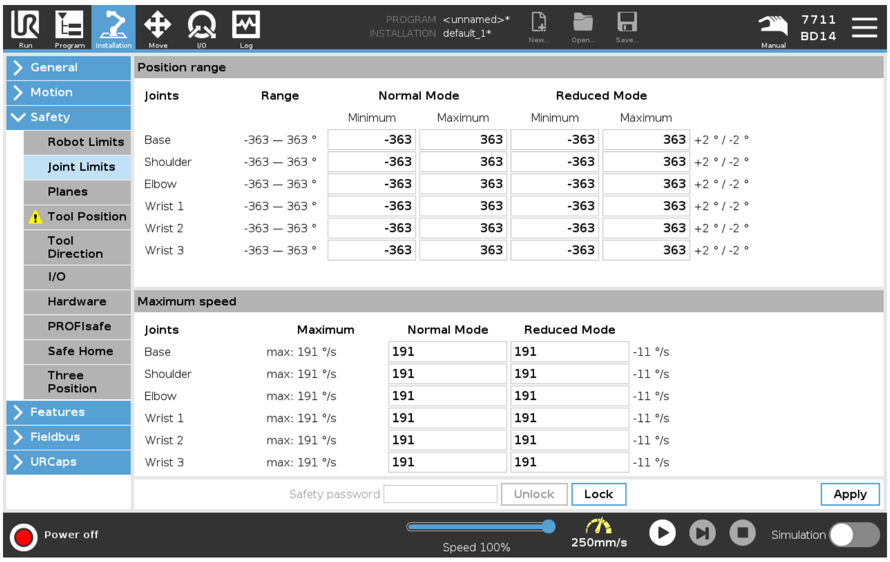


> **NOTICE**
>
> You can switch back to **Factory Presets** for all robot limits to reset to their default settings.

2.6.2. Joint Limits

Description

Joint limits allow you to restrict individual robot joint movements in joint space i.e. joint rotational position and joint rotational speed. Joint limiting can also be called software based axis limiting. The joint limit options are: **Maximum speed** and **Position range**.

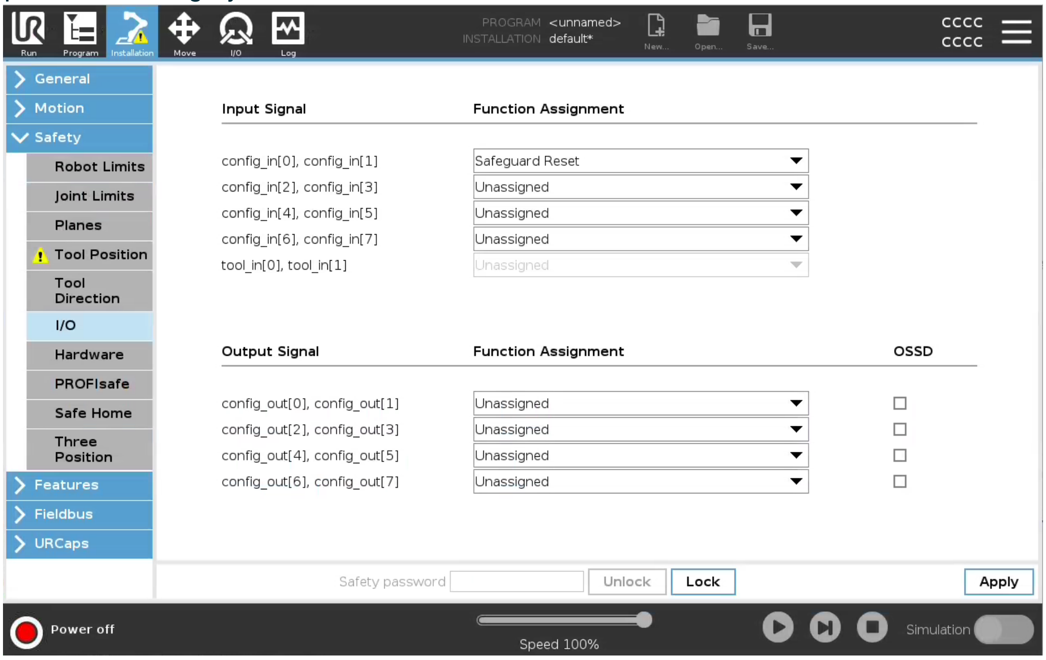


Copyright © 2009–2025 by Universal Robots A/S. All rights reserved.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

2.7. Safety I/O

Description The I/O are divided between inputs and outputs and are paired up so that each function provides a Category 3 and PLd I/O.



Control Box Inputs The inputs are described in the tables below:

Emergency Stop Button	Performs a Stop Category 1 (IEC 60204-1) informing other machines using the System Stop output if that output is defined. A stop is initiated in anything connected to the output.
Robot Emergency Stop	Performs a Stop Category 1 (IEC 60204-1) via Control Box input, informing other machines using the System Emergency Stop Output if that output is defined.
External Emergency Stop	Performs a Stop Category 1 (IEC 60204-1) on robot only.
Reduced	All safety limits can be applied while the robot is using a **Normal** configuration, or a **Reduced** configuration. When configured, a low signal sent to the inputs causes the safety system to transition to the reduced configuration. The robot arm decelerates to satisfy the reduced parameters. The safety system guarantees the robot is within reduced limits less than 0.5s after the input is triggered. If the robot arm continues to violate any of the reduced limits, a Stop Category 0 is triggered. Trigger planes can also cause a transition to the reduced configuration. The safety system transitions to the normal configuration in the same way.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Control Box Inputs The inputs are described in the tables below:

Operational Mode	When an external mode selection is used it switches between **Automatic Mode** and **Manual Mode**. The robot is in Automatic mode when input is *low* and Manual mode when the input is *high*.
Safeguard Reset	Returns from the Safeguard Stop state, when a rising edge on the Safeguard Reset input occurs. When a Safeguard Stop occurs, this input ensures that the Safeguard Stop state continues until a reset is triggered.
Safeguard	A stop triggered by a safeguard input. Performs a Stop Category 2 (IEC 60204-1) in all modes, when triggered by a Safeguard.
Automatic Mode Safeguard Stop	Performs a Stop Category 2 (IEC 60204-1) in Automatic mode ONLY. Automatic Mode Safeguard Stop can only be selected when a Three-Position Enabling Device is configured and installed.
Automatic Mode Safeguard Reset	Returns from the Automatic Mode Safeguard Stop state when a rising edge on the Automatic Mode Safeguard Reset input occurs.
3-Position Enabling Device	In Manual Mode, an external 3-Position Enabling Device must be pressed and held in the center-on position to move the robot. If you are using a built-in 3-Position Enabling Device, the button must be pressed and held in the mid position to move the robot.
Freedrive on robot	You can configure the Freedrive input to enable and use Freedrive without pressing the Freedrive button on a standard TP, or without having to press-and-hold any of the buttons on the 3PE TP in the light-press position.

WARNING

When the default Safeguard Reset is disabled, an automatic reset happens when the safeguard no longer triggers a stop.

This can happen if a person passes though the field of the safeguard.

If a person is not detected by the safeguard and the person is exposed to hazards, automatic reset is forbidden by standards.

- Use the external reset to ensure resetting only when a person is not exposed to hazards.

WARNING

When Automatic Mode Safeguard stop is enabled, a safeguard Stop is not triggered in Manual Mode.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Control Box Outputs

All safety outputs go low in the event of a safety system violation or fault. This means the System Stop output initiates a stop even when an E-stop is not triggered.
You can use the following Safety functions output signals. All signals return to low when the state which triggered the high signal has ended:

[1]System Stop	Signal is *Low* when the safety system has been triggered into a stopped state including by the Robot Emergency Stop input or the Emergency Stop Button. To avoid deadlocks, if the Emergency Stopped state is triggered by the System Stop input, low signal will not be given.
Robot Moving	Signal is *Low* if the robot is moving, otherwise high.
Robot Not Stopping	Signal is *High* when the robot is stopped or in the process of stopping due to an emergency stop or safeguard stop. Otherwise it will be logic low.
Reduced	Signal is *Low* when reduced parameters are active or if the safety input is configured with a reduced input and the signal is currently low. Otherwise the signal is high.
Not Reduced	This is the inverse of Reduced, defined above.
Safe Home	Signal is *High* if the Robot Arm is stopped and is located in the configured Safe Home Position. Otherwise, the signal is *Low*. This is often used when UR robots are integrated with mobile robots.
3-Position Enabling Stopped	Signal is low when a three position stop is active, high otherwise.
Not 3-Position Enabling Stopped	Signal is low when a three position stop is inactive, high otherwise.

NOTICE

Any external machinery receiving its Emergency Stop state from the robot through the System Stop output must comply with ISO 13850. This is particularly necessary in setups where the Robot Emergency Stop input is connected to an external Emergency Stop device. In such cases, the System Stop output becomes high when the external Emergency Stop device is released. This implies that the emergency stop state at the external machinery will be reset with no manual action needed from the robot’s operator. Hence, to comply with safety standards, the external machinery must require manual action in order to resume.

[1]System Stop was previously known as "System Emergency Stop" for Universal Robots robots. PolyScope can display "System Emergency Stop".

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

2.8. Software Safety Restrictions

Description

NOTICE

Configuring planes is entirely based on features. We recommend that you create and name all features before editing the safety configuration, as the robot is powered off once the Safety Tab has been unlocked, making it impossible to move the robot.

Safety planes restrict robot workspace. You can define up to eight safety planes, restricting the robot tool and elbow. You can also restrict elbow movement for each safety plane and disable it by deselecting the checkbox. Before configuring safety planes, you must define a feature in the robot installation. The feature can then be copied into the safety plane screen and configured.

WARNING

Defining safety planes only limits the defined Tool spheres and elbow, not the overall limit for the robot arm. This means that specifying a safety plane, does not guarantee that other parts of the robot arm will obey this restriction.

Safety Planes Modes

You can configure each plane with restrictive **Modes** using the icons listed below.

	Disabled	The safety plane is never active in this state.
	Normal	When the safety system is Normal , a normal plane is active and it acts as a strict limit on the position.
	Reduced	When the safety system is Reduced, a reduced plane is active, acting s as a strict limit on the position.
	Normal & Reduced	When the safety system is either normal or reduced, a normal and reduced plane is active and acts as a strict limit on the position.
	Trigger Reduced	The safety plane causes the safety system to switch to Reduced if the robot Tool or Elbow is positioned beyond it.
	Show	Pressing this icon hides or shows the safety plane in the graphics pane.
	Delete	Deletes the created safety plane. There is no undo/redo action. If a plane is deleted in error, it must be remade.
	Rename	Pressing this icon allows you to rename the plane.

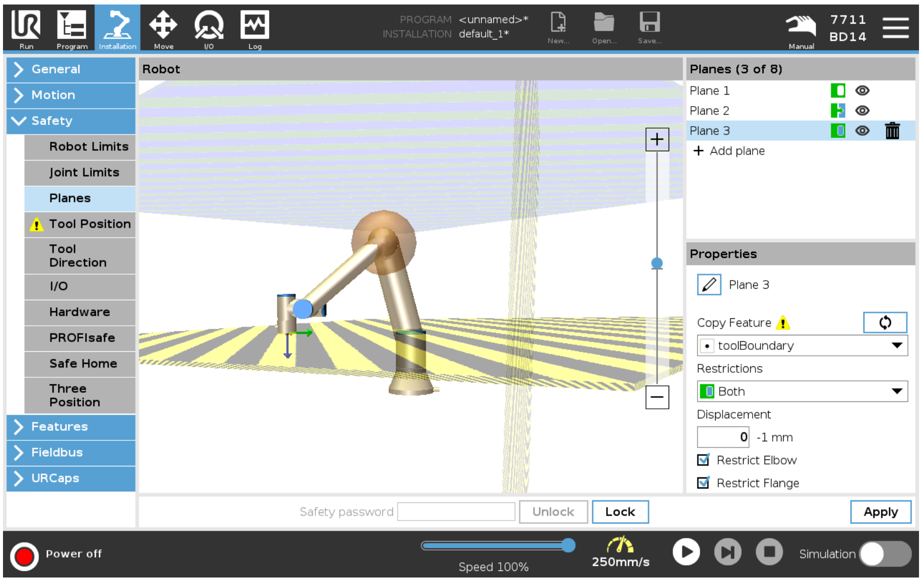
Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Configuring safety planes

1. In your PolyScope header, tap **Installation**.
2. In the Side Menu on the left of the screen, tap Safety and select **Planes**.
3. On the top right of the screen, in the Planes field, tap **Add plane**.
4. On the bottom right of the screen, in the **Properties** field, set up Name, Copy Feature and Restrictions.

Copy Feature

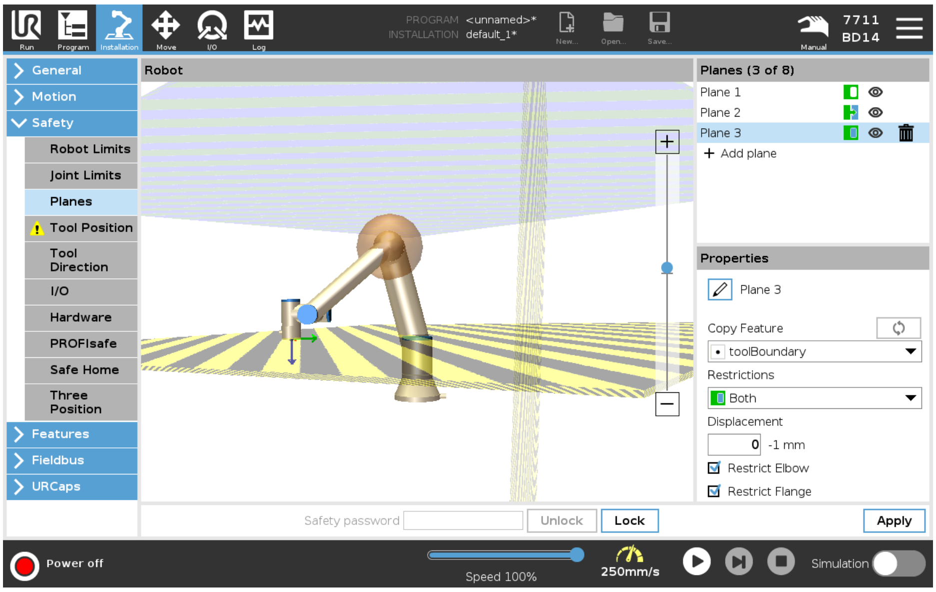
In **Copy Feature**, only Undefined and Base are available. You can reset a configured safety plane by selecting **Undefined**
If the copied feature is modified in the Features screen, a warning icon appears to the right of the Copy Feature text. This indicates that the feature is out of sync i.e. the information in the properties card is not updated to reflect the modifications that may have been made to the Feature.



Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Color Codes

Gray	Plane is configured but disabled (A)
Yellow & Black	Normal Plane (B)
Blue & Green	Trigger Plane (C)
Black Arrow	The side of the plane the tool and/or elbow is allowed to be on (For Normal Planes)
Green Arrow	The side of the plane the tool and/or elbow is allowed to be on (For Trigger Planes)
Gray Arrow	The side of the plane the tool and/or elbow is allowed to be on (For Disabled Planes)



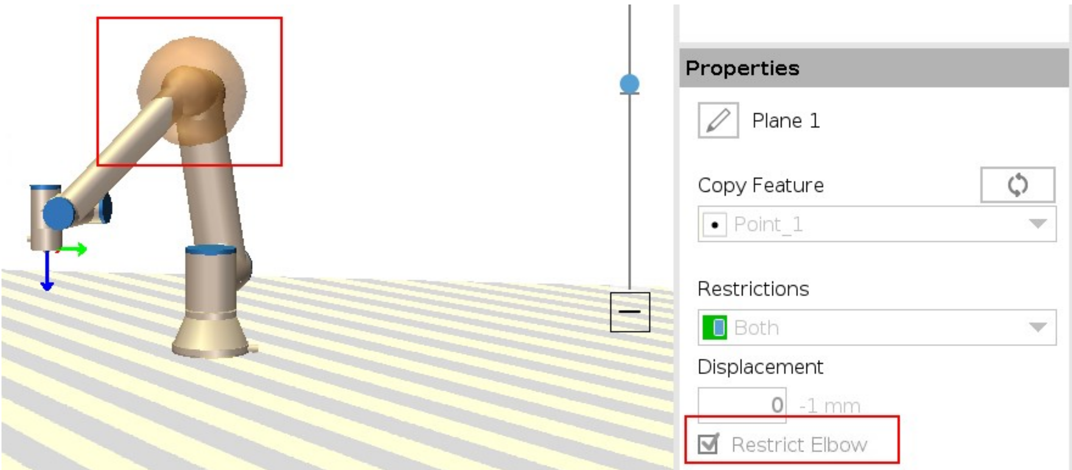
Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Elbow Restriction

You can enable **Restrict Elbow** to prevent robot elbow joint from passing through any of your defined planes. Disable Restrict Elbow for elbow to pass through planes.
The diameter of the ball that restricts the elbow is different for each size of robot.

UR3e	0.1 m
UR5e	0.13 m
UR10e / UR16e	0.15 m
UR15	0.15 m
UR20 / UR30	0.19 m

The information about the specific radius can be found in the *urcontrol.conf* file on the robot under the section [Elbow].



Tool Flange Restriction

Restricting the tool flange prevents the tool flange and the attached tool from crossing a safety plane. When you restrict the tool flange, the unrestricted area is the area inside of the safety plane, where the tool flange can operate normally.
The tool flange cannot cross the restricted area, outside of the safety plane.

Removing the restriction allows the tool flange to go beyond the safety plane, to the restricted area, while the attached tool remains inside of the safety plane.

You can remove the tool flange restriction when working with a large tool off-set. This will allow extra distance for the tool to move.

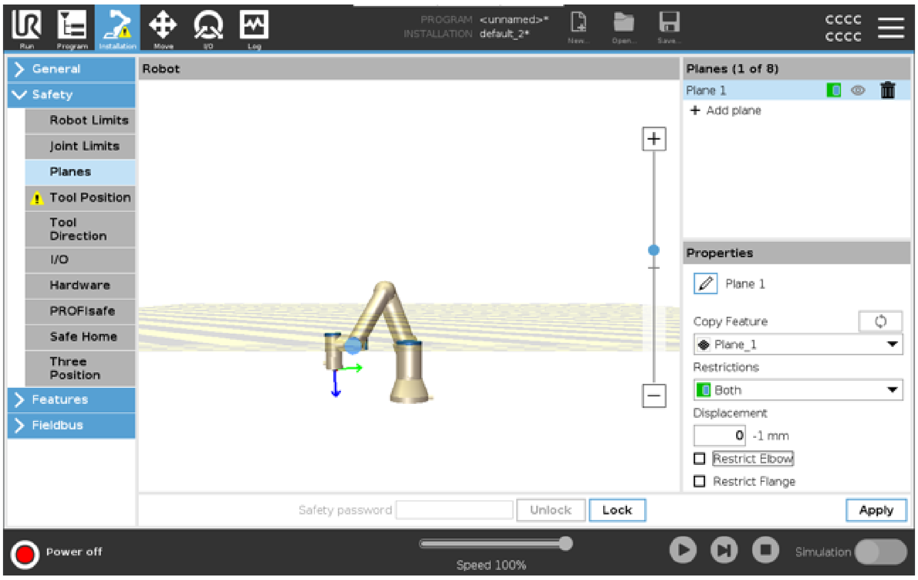
Restricting the tool flange requires the creation of a plane feature. The plane feature is used to set up a safety plane later in the safety settings.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Adding a plane feature example

Displacement offsets the plane in either the positive or negative direction along the plane normal (Z-axis of the plane feature).

Deselect the checkbox for the Elbow and the Tool Flange so they do not trigger the safety plane. The Elbow can remain checked as needed by your application.



The unrestricted tool flange can cross a safety plane, even when no tool is defined.

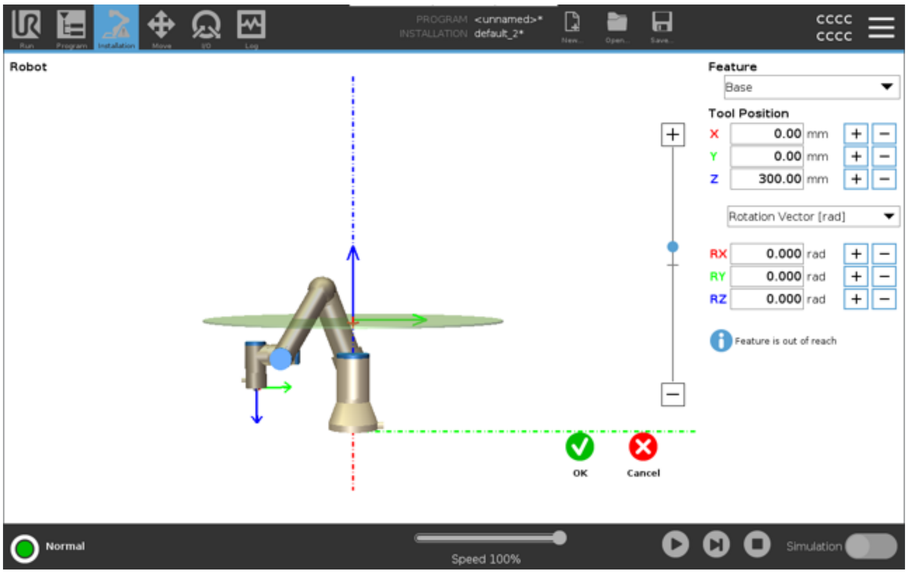
If no tool is added, a warning on the Tool Position button prompts you to correctly define the tool.

When working with an unrestricted tool flange and a defined tool, it is ensured that the dangerous part of the tool can't go above and/or beyond certain area. The unrestricted tool flange can be used for any application where safety planes are needed, like Welding or Assembly.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Tool flange restriction example

In this example, an X-Y-plane is created with an offset of 300mm along the positive Z-axis with reference to the base feature.
The Z-axis of the plane can be thought of as “pointing” towards the restricted area.
If the safety plane is needed on e.g., the surface of a table, rotate the plane 3.142 rad or 180° around either the X- or Y-axis so the restricted area is under the table.
(TIP: Change the display of rotation from “Rotation Vector [rad]” to “RPY [°]”)



If needed it is possible to offset the plane in either positive or negative Z-direction later in the safety settings.
When satisfied with the position of the plane, tap OK.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

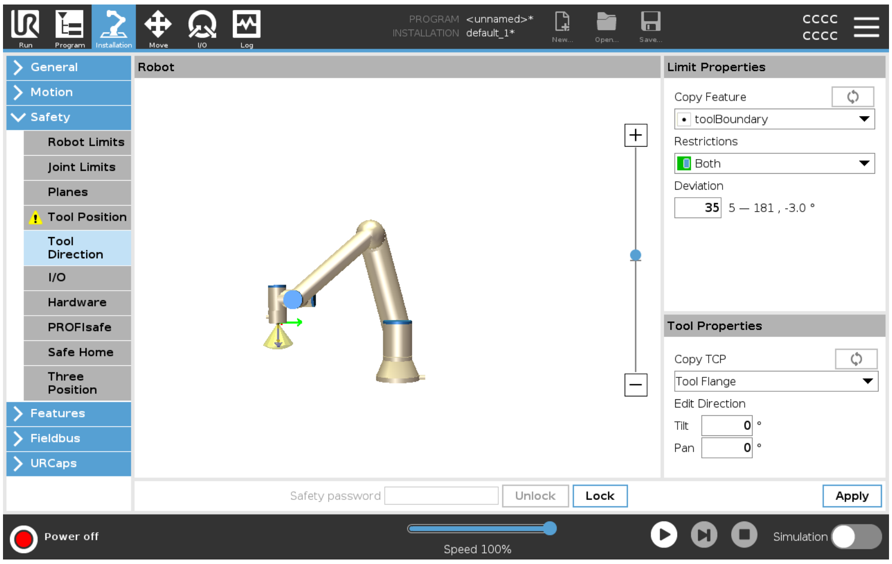
Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

2.8.1. Tool Direction Restriction

Description

The Tool Direction screen can be used to restrict the angle in which the tool is pointing. The limit is defined by a cone that has a fixed orientation with respect to the robot arm Base. As the robot arm moves around, tool direction is restricted so it remains within the defined cone. The default direction of the tool coincides with the Z-axis of the tool output flange. It can be customized by specifying tilt and pan angles.
Before configuring the limit, you must define a point or plane in the robot installation. The feature can then be copied and its Z axis used as the center of the cone defining the limit.

> **NOTICE**
>
> Configuration of the tool direction is based on features. We recommend you create desired feature(s) before editing the safety configuration, as once the Safety Tab has been unlocked, the robot arm powers off making it impossible to define new features.



Copyright © 2009–2025 by Universal Robots A/S. All rights reserved.

Limit Properties

The Tool Direction limit has three configurable properties:

1. **Cone center**: You can select a point or plane feature from the drop-down menu, to define the center of the cone. The Z axis of the selected feature is used as the direction around which the cone is centred.
2. **Cone angle**: You can define how many degrees the robot is allowed to deviate from center.

Disabled Tool direction limit	Never active
Normal Tool direction limit	Active only when safety system is in **Normal mode**
Reduced Tool direction limit	Active only when the safety system is in **Reduced mode**
Normal & Reduced Tool direction limit	Active when the safety system is in **Normal mode** as well as when it is in **Reduced mode**.

You can reset the values to default or undo the Tool Direction configuration by setting the copy feature back to "Undefined".

Tool Properties

By default, the tool points in the same direction as the Z axis of the tool output flange. This can be modified by specifying two angles:

- **Tilt angle**: How much to tilt the Z axis of the output flange towards the X axis of the output flange
- **Pan angle**: How much to rotate the tilted Z axis around the original output flange Z axis.

Alternatively, the Z axis of an existing TCP can be copied by selecting that TCP from the drop-down menu.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

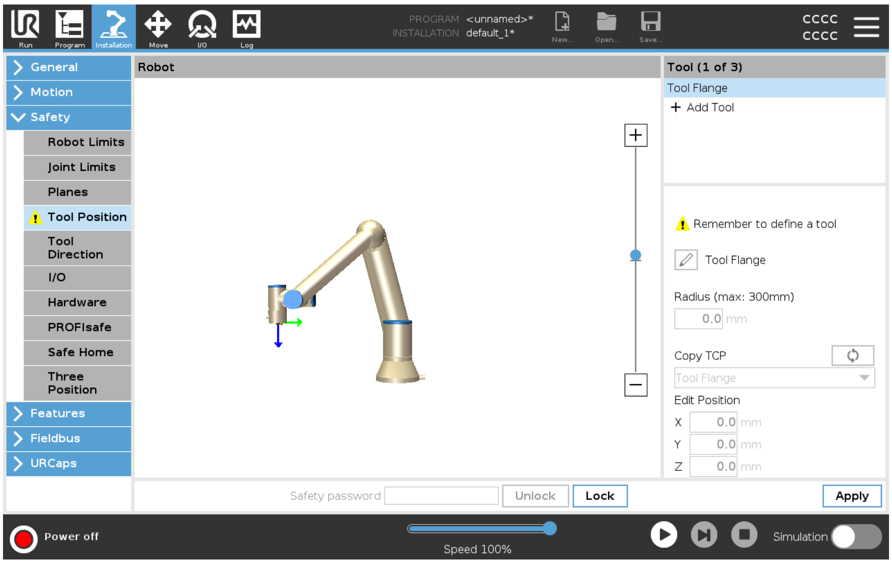
2.8.2. Tool Position Restriction

Description

The Tool Position screen enables more controlled restriction of tools and/or accessories placed on the end of the robot arm.

- **Robot** is where you can visualize your modifications.
- **Tool** is where you can define and configure a tool up to two tools.
- **Tool_1** is the default tool defined with values x=0.0, y= 0.0, z=0.0 and radius=0.0. These values represent the robot tool flange.

Under Copy TCP, you can also select **Tool Flange** and cause the tool values to go back to 0.
A default sphere is defined at the tool flange.



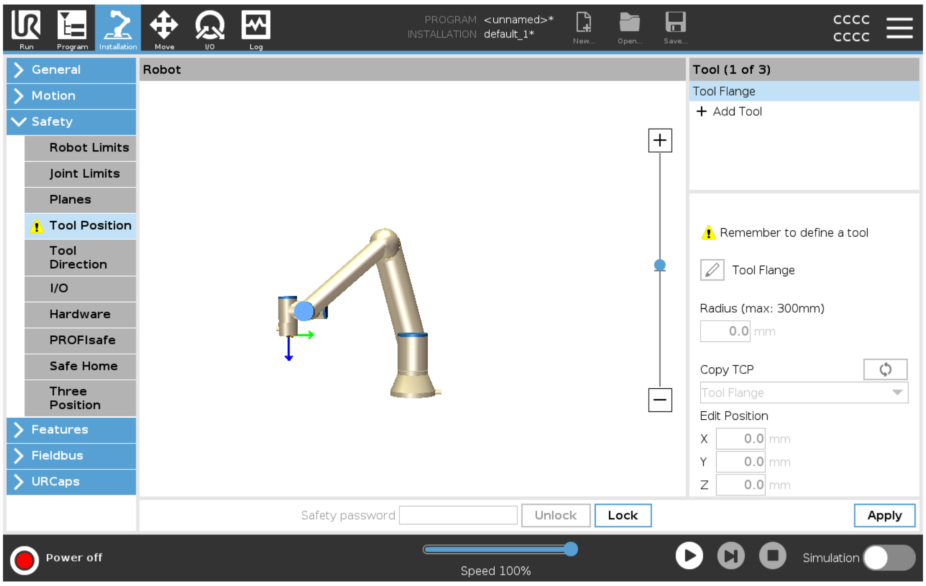
Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

User defined tools

For the user defined tools, the user can change:

- **Radius** to change the radius of the tool sphere. The radius is considered when using safety planes. When a point in the sphere passes a reduced trigger plane, the robot switches to a Reduced configuration . The safety system prevents any point on the sphere from passing a safety plane.
- **Position** to change the position of the tool with respect to the tool flange of the robot. The position is considered for the safety functions for tool speed, tool force, stopping distance and safety planes.

You can use an existing Tool Center Point as a base for defining new tool positions. A copy of the existing TCP, predefined in General menu, in TCP screen, can be accessed in Tool Position menu, in Copy TCP drop-down list.
When you edit or adjust the values in the **Edit Position** input fields, the name of the TCP visible in the drop down menu changes to **custom**, indicating that there is a difference between the copied TCP and the actual limit input. The original TCP is still available in the drop down list and can be selected again to change the values back to the original position. The selection in the copy TCP drop down menu does not affect the tool name.
Once you apply your Tool Position screen changes, if you try to modify the copied TCP in the TCP configuration screen, a warning icon appears to the right of the Copy TCP text. This indicates that the TCP is out of sync i.e. the information in the properties field is not updated to reflect modifications that may have been made to the TCP. The TCP can be synced by pressing the sync icon.
The TCP does not have to be synced in order to define and use a tool successfully.
You can rename the tool by pressing the pencil tab next to the displayed tool name. You can also determine the Radius with an allowed range of 0-300 mm. The limit appears in the graphics pane as either a point or a sphere depending on radius size.



Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Tool Position Warning

You must set a Tool Position within the safety settings, for the safety plane to trigger correctly when the tool TCP approaches the safety plane.
The warning remains on the Tool Position if:

- You fail to add a new tool under Tool Flange.

To configure the tool position

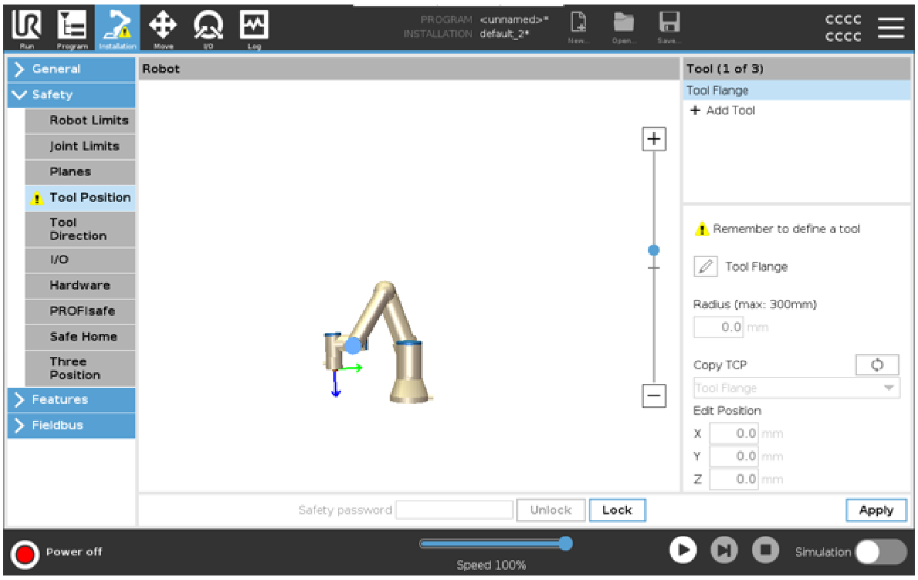
1. In the Header tap **Installation**.
2. On the left side of the screen, under Safety, tap **Tool Position**.
3. On the right side of the screen, select **Add Tool**.
   - The newly added tool has a default name: **Tool_x**.
4. Tap the edit button to rename **Tool_x** to something more identifiable.
5. Edit the Radius and Position to match that of the tool you are currently using, or use the Copy TCP drop-down and choose a TCP from the General>TCP settings if such is defined.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

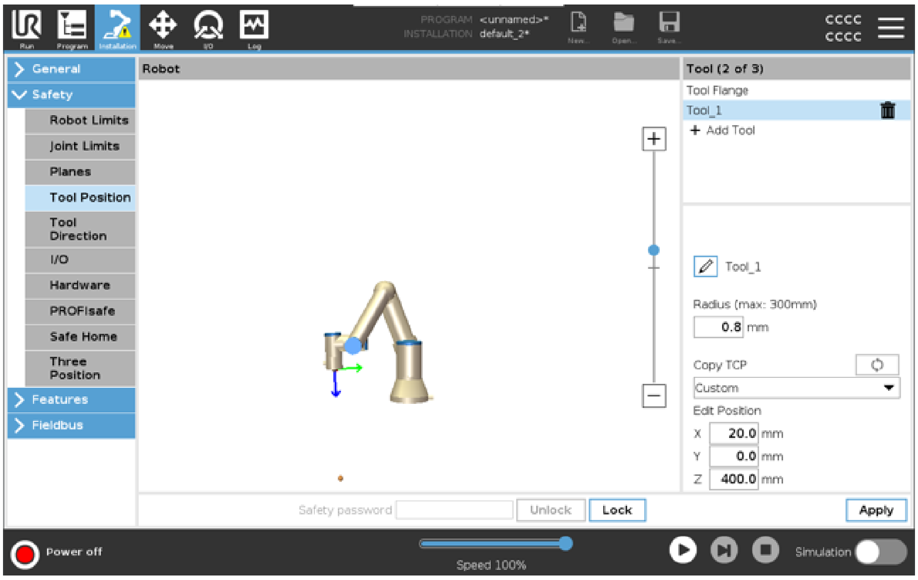
Tool Position Warning example

In this example, a Radius of 0.8mm is set and the TCP position to XYZ [20, 0, 400] in millimeters respectively. Optionally you can choose to "Copy TCP" by using the drop-down menu if one has already been set in the ->General/TCP settings. Once the Apply is tapped in the bottom right corner of the screen, you are DONE.

The warning on the Tool Position button indicates a tool is not added under Tool Flange.



Tool Position button without the warning indicates a tool (other than the Tool Flange) is added.

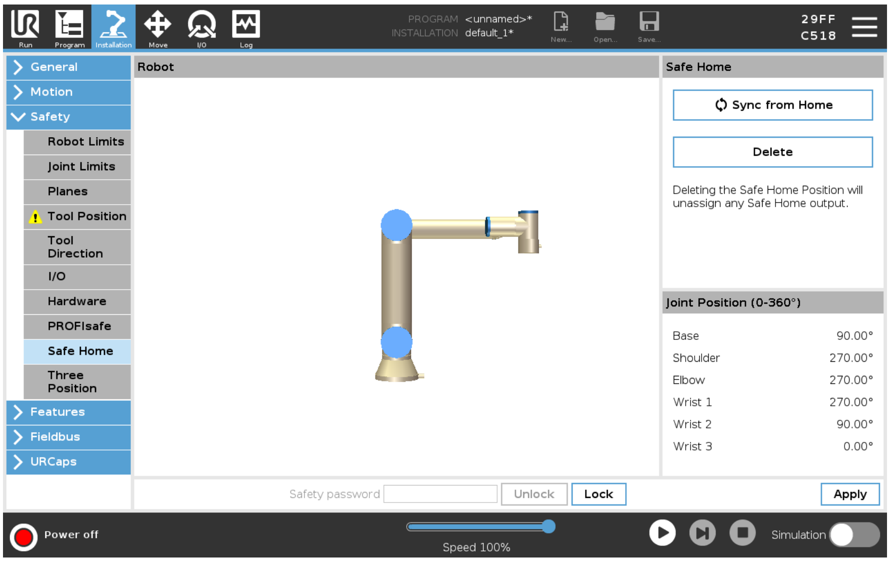


Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

2.9. Safe Home Position

Description

Safe Home is a return position defined by using the user-defined Home Position.
Safe Home I/Os are active when the Robot Arm is in the Safe Home Position and a Safe Home I/O is defined.
The Robot Arm is in the Safe Home Position if the joint positions are at the specified joint angles or a multiple of 360 degrees thereof.
The Safe Home Safety Output is active when the robot is standing still at the Safe Home Position.



Syncing from Home

To sync from Home

1. In the Header, tap **Installation**.
2. In the Side Menu on the left of the screen, tap **Safety** and select **Safe Home**.
3. Under **Safe Home**, tap **Sync from Home**.
4. Tap **Apply** and in the dialog box that appears, select **Apply and restart**.

Safe Home Output

The Safe Home Position must be defined before the Safe Home Output.

Defining Safe Home Output

To define Safe Home Output

1. In the Header, tap **Installation**.
2. In the Side Menu on the left of the screen, under **Safety**, select **I/O**.
3. On the I/O screen in the Output Signal, under Function Assignment, in drop-down menu, select **Safe Home**.
4. Tap **Apply** and in the dialog box that appears, select **Apply and restart**.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Editing Safe Home

To edit Safe Home

Editing Home does not automatically modify a previously defined Safe Home position. While these values are out of sync, Home program node is undefined.

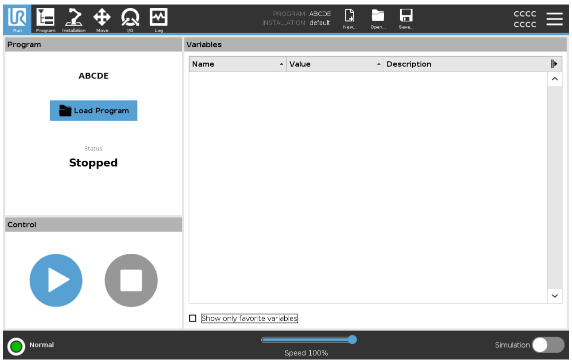
1. In the Header, tap **Installation**.
2. In the Side Menu on the left of the screen, under **General**, select **Home**.
3. Tap **Edit Position** and set the new robot arm position and tap **OK**.
4. In the Side Menu, under **Safety**, select **Safe Home**. You need a Safety password to **Unlock** the Safety Settings.
5. Under **Safe Home**, tap **Sync from Home**

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

3. Run Tab

Description

The **Run** tab allows you to do simple operations and monitor the state of your robot. You can load, play, pause and stop a program, as well as monitor variables. The Run Tab is most useful when the program is created and the robot is ready for operation.



Program

The Program pane displays the name and status of the current program.

To load a new program

1. In the Program pane, tap **Load Program**.
2. Select your desired program from the list.
3. Tap **Open** to load the new program.

   The variables, if present, are displayed when you play the program.

Variables

The Variables pane displays the list of variables used by programs to store and update values during runtime.

- Program variables belong to programs.
- Installation variables belong to installations that can be shared among different programs. The same installation can be used with multiple programs.

All program variables and installation variables in your program are displayed in the Variables pane as a list showing the Name, Value and Description of the variable.

Variable descriptions

You can add information to your variables by adding variable descriptions in the Description column. You can use the variable descriptions to convey the purpose of the variable and/or the meaning of its value to operators using the Run tab screen and/or other programmers.
Variable descriptions (if used) can be up to 120 characters, displayed in the Description column of the variables list on the Run tab screen and the Variables tab screen.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Favorite variables

You can display selected variables by using the **Show only favorite variables** option.
To show favorite variables

1. Under Variables, check the **Show only favorite variables** box.
2. Check **Show only favorite variables** again to show all variables.

You cannot designate favorite variables in the Run Tab, you can only display them. Designating favorite variables depends on the variable type.

To designate favorite program variables

1. In the Header, tap **Program**.
   The variables are listed under **Variable Setup**.
2. Select the desired variables.
3. Check the **Favorite variable** box.
4. Tap **Run** to return to your variable display.

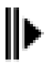
To designate favorite installation variables

1. In the Header, tap **Installation**.
2. Under General, select **Variables**.
   The variables are listed under **Installation Variables**.
3. Select the desired variables.
4. Check the **Favorite variable** box.
5. Tap **Run** to return to your variable display.

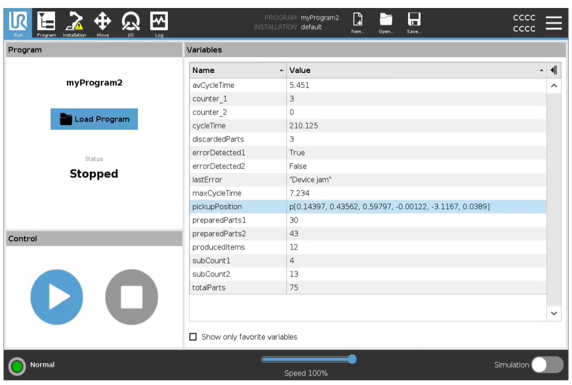
Collapse/expand the Description column

A variable description spans multiple lines to fit the width of the Description column if necessary. You can also collapse and expand the Description column by using the buttons shown below.

To collapse/expand the Description column

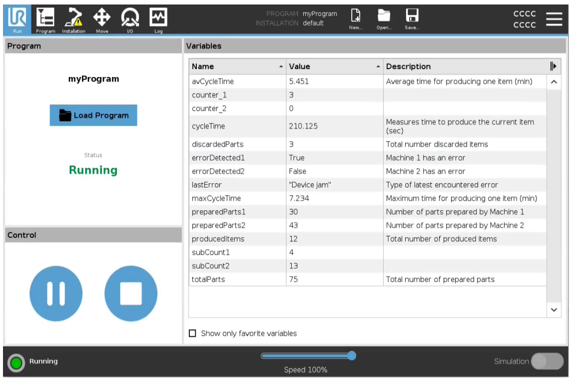
1. Tap  to collapse the Description column.
2. Tap  to expand the Description column.

Collapsed Description column



Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Expanded Description column



Control The Control pane allows you to control the running program. You can play and stop, or pause and resume a program, using the buttons listed in the table below:

- The Play button, Pause button and the Resume Button are combined.
- The Play button changes to Pause when the program is running.
- The Pause button changes to Resume.

Button		Function
Play		To play a program 1. Under Control, tap **Play** to start running a program from the beginning.
Resume		To resume a paused program 1. Tap **Resume** to continue running the paused program.
Stop		To stop a program 1. Tap **Stop** to stop the running program You cannot resume a stopped program. You can tap **Play** to restart the program.
Pause		To pause a program 1. Tap **Pause** to pause a program at a specific point. You can resume a paused program.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

3.1. Move Robot into Position

Description

Access the **Move Robot into Position** screen when the Robot Arm must move to a particular start position before running a program, or when the Robot Arm is moving to a waypoint while modifying a program.

In cases where the **Move Robot into Position** screen cannot move the Robot Arm to the program start position, it moves to the first waypoint in the program tree.
The Robot Arm can move to an incorrect pose if:

- The TCP, feature pose or waypoint pose of the first movement is altered during program execution before the first move is executed.
- The first waypoint is inside an If or Switch program tree node.

Accessing the Move Robot into Position Screen

1. Tap the Run tab in the header.
2. In the **Footer**, tap **Play** to access the **Move Robot into Position** screen.
3. Follow the on-screen instructions to interact with the animation and the real robot.

Move robot to

Hold down **Move robot to:** to move the Robot Arm to a start position. The animated Robot Arm displayed on-screen shows the desired movement about to be performed.

> **NOTICE**
> Collision can damage the robot or other equipment. Compare the animation with the position of the real Robot Arm to ensure the Robot Arm can safely perform the movement without colliding with any obstacles.

Manual

Tap **Manual** to access the **Move** screen where the Robot Arm can be moved by using the Move Tool arrows and/or configuring Tool Position and Joint Position coordinates.

3.2. Program

Description

The **Program** field, displays the name of the program that was loaded on to the robot and its current status. You can tap the **Load Program** tab to load a different program.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

3.3. Variables

Description

A robot program can make use of variables to store and update values during runtime. The variables in the table below are available:

Installation variables	These can be used by multiple programs and their names and values are persisted together with the robot installation. Installation variables keep their value after the robot and control box has been rebooted.
Regular program variables	These are available to the running program only and their values are lost as soon as the program is stopped.

Show waypoints allows the robot program to use script variables to store information about waypoints. You can select the **Show Waypoints** checkbox, under **Variables** to show script variables in the variables list.

Available Output Actions

bool	A boolean variable whose value is either `True` or `False`.
int	A whole number in the range from − 2147483648 to 2147483647 (32 bit).
float	A floating point number (decimal) (32 bit).
string	A sequence of characters.
pose	A vector describing the location and orientation in Cartesian space. It is a combination of a position vector (*x*, *y*, *z*) and a rotation vector (*rx*, *ry*, *rz*) representing the orientation, written `p[x, y, z, rx, ry, rz]`.
list	A sequence of variables.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

4. Program Tab

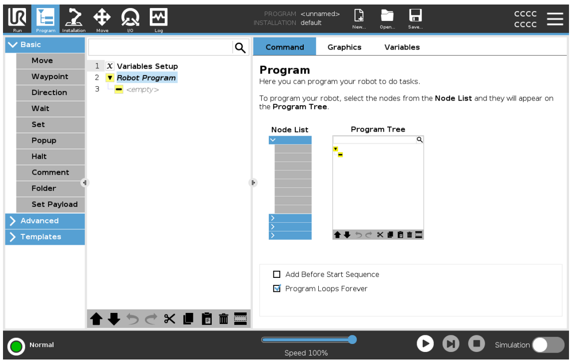
Description

The Program tab is the where you create and edit robot programs. There are two main areas:

- The left side contains the program nodes you can add to your robot program.

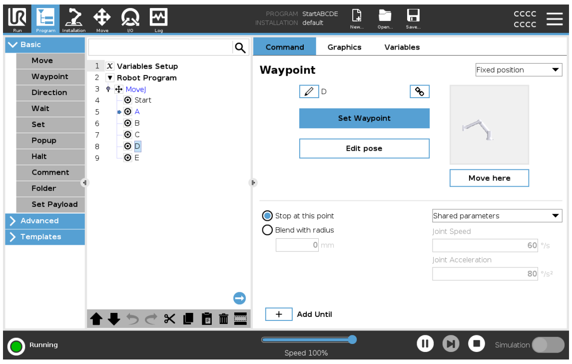
  You can use the Basic, Advanced and Template dropdowns to the very left.
- The right side contains the configuration of the program nodes you can add to your program.

  You can use Command, Graphics and Variables options.



Program Tree

The program tree is built as you add program nodes to your program.
You can use the Command tab to configure the functionality of the added program nodes.



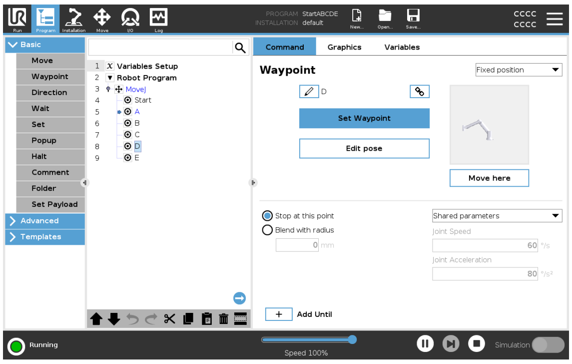
Adding program nodes

- You cannot run an empty program tree or a program containing incorrectly configured program nodes.
- Incorrectly configured programs nodes are higlighted in yellow.
- Correctly configured program nodes are highlighted in white.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Program Execution Indication

You can follow the the flow of a long robot program by looking at the active program node.



When the program is running, the program node currently being executed is indicated by a small icon next to that node.

The path of execution is highlighted with blue arrow.

Tapping the icon at the corner of the program allows it to track the command being executed

Search Button

You can also search for a specific command/program node. This is useful when you have a long program with many different program nodes.

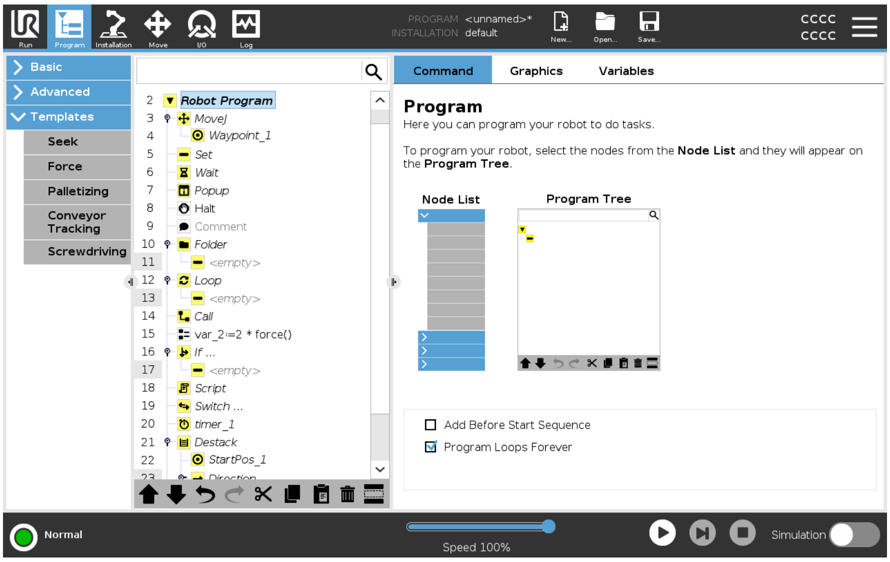
Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

4.1. Robot Program Configuration

Description

Use the Robot Program Configuration to create a configuration for the entire robot program.
While this type of configuration does not affect individual program nodes, it affects the entire robot program.
You can, for example do the following:

- Add instructions to be executed before the robot program is started.
- Set an initial variable value for the program start.
- Make the program loop forever.



Add Before Start Sequence

Select this checkbox to add instructions or program nodes that are executed before the main program starts. These nodes are only executed once even though the robot program is set to loop forever.

Program Loops Forever

Select this checkbox to allow the program to start again when it reaches the end of the program.

Example

You can initialize grippers, move the robot to a "home" position, or reset signals to and from external sources.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Set Initial Variable Value	Select this to set initial values of program variables. 1. Select a variable from the drop down list, or use the variable selector box. 2. Enter an expression for that variable. This expression is used to set the variable value at program start. 3. You can select **Keep value f rom previous run** to initialize the variable to the value found on the **Variables** tab (see 4.2.4 Variables Tab on page 78). This allows variables to maintain their values between program executions. The variable gets its value from the expression if the program is run for the first time, or if the value tab has been cleared.

A variable can be deleted from the program by setting its name to blank (only spaces).

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

4.2. Program Tree Toolbar

Description

You can work with the program nodes that have been added to the program tree by using the icons in the bottom of the program tree.

Icons in the Program Tree toolbar

Use the toolbar at the base of the Program Tree to modify the Program Tree.

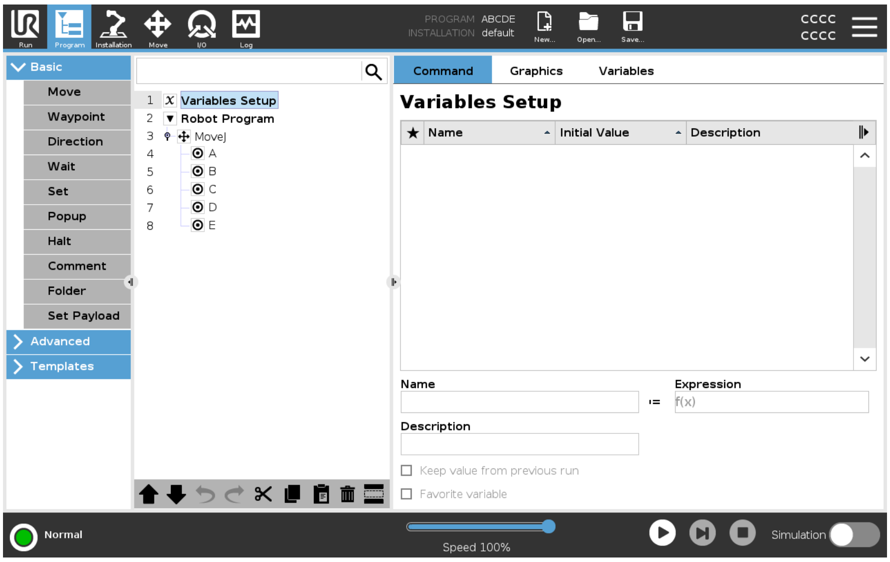
Undo & Redo		undo and redo changes to commands.
Move Up & Move Down		changes the position of a node.
Cut		cuts a node and allows it to be used for other actions (e.g., paste it on other place on the Program Tree).
Copy		copies a node and allows it to be used for other actions (e.g., paste it on other place on the Program Tree).
Paste		pastes a node that was previously cut or copied.
Delete		removes a node from the Program Tree.
Suppress		suppresses specific nodes on the Program Tree.
Search Button		search in the Program Tree. Tap the ✕ icon to exit search.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

4.2.1. Variable Setup

Description

The Variable Setup is always the first node on the program tree.
The Variable Setup pane is where you name and edit program variables, and designate them as favorites.
Designating a favorite variable displays it on the Variable pane in the Program tab screen and on the Run tab screen.



To designate a program variable as favorite

1. In the Header, tap **Program**.
2. Under Variables Setup, select a variable.
3. Check the **Favorite variable** box to designate the selected variable as favorite.

Editing program variables

Editing program variables includes naming, describing and setting an expression.

To name a program variable

1. Under Variables Setup, select a variable.
2. Select the **Name** field.
3. Type a name using the on-screen keyboard that appears.

To describe a program variable

1. Under Variables Setup, select a variable.
2. Select the **Description** field.
3. Type a description using the on-screen keyboard that appears.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

To set an expression program variable

1. Under Variables Setup, select a variable.
2. Select the **Expression** field.
3. Type the expression using the on-screen keyboard that appears.

Initial value

An initial value is the first value you assign to a program variable, when you start a program. You can check the **Keep value from previous run** box to replace the initial value with a value from a previously run program. However, if you load a new program, after using a value from a previously run program, the initial value is reinstated.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

4.2.2. Command Tab

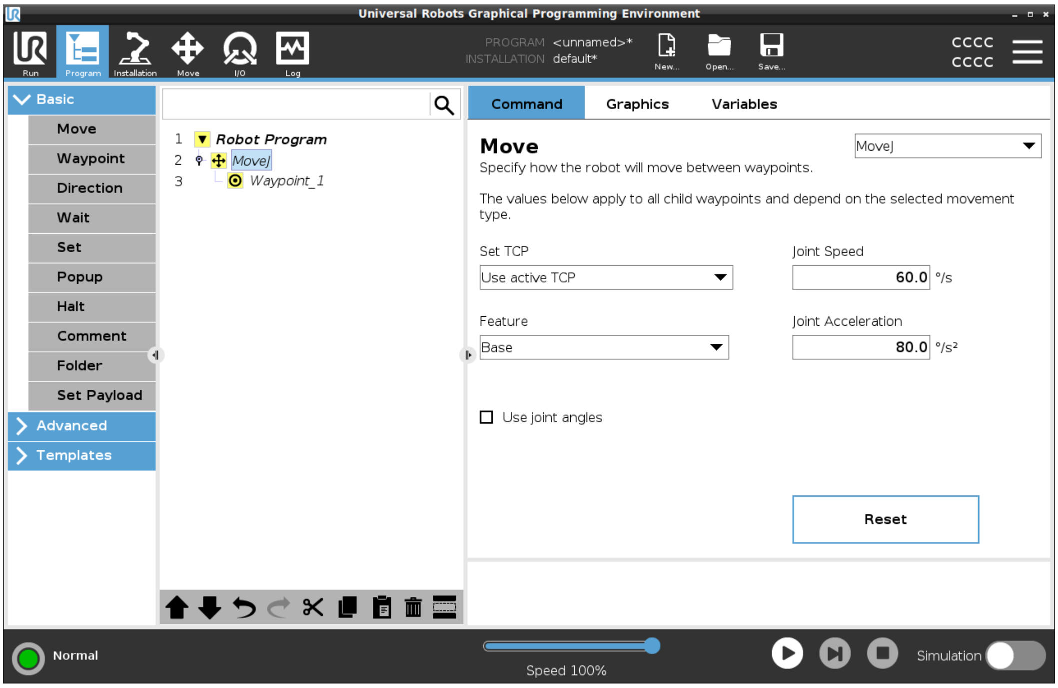
Description

The Command tab contains the configuration options for the selected program node. These configuration options appear under the Command Tab on the right side of the screen.
The Command tab's pane changes when you select each different program node, on the left side of the screen.

Examples of different commands are given below.

Move

The Move command is one of the most often used commands in PolyScope.
Here you can see a MoveJ command is selected. Other information such as TCP, joint speed and joint acceleration is visible.



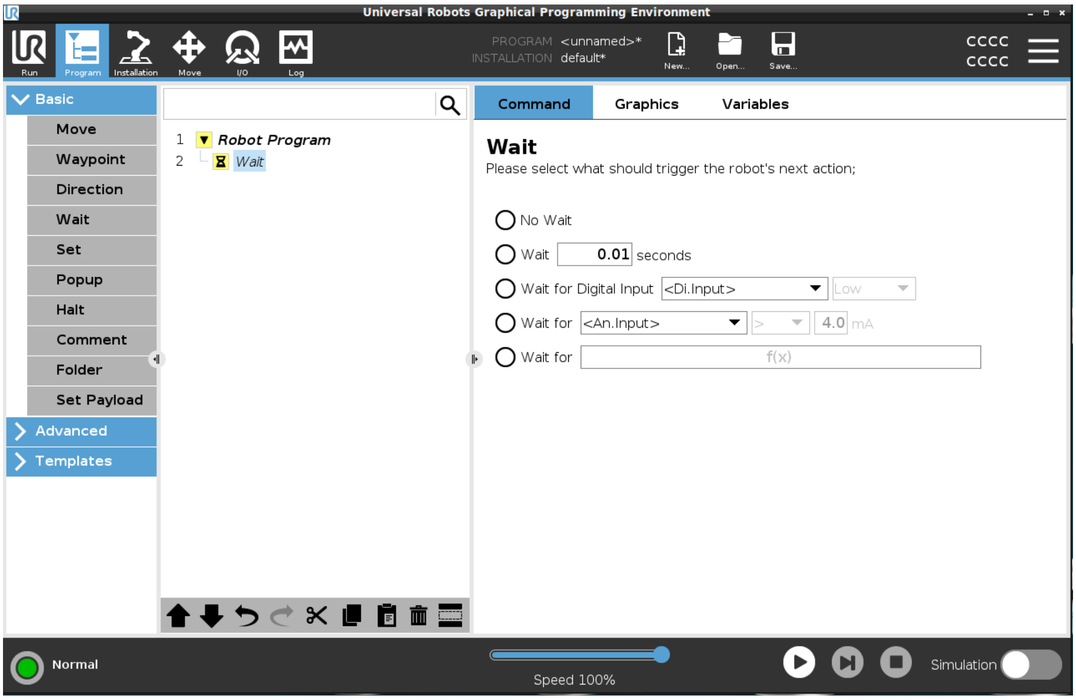
Example

The Move command controls the robot's motion via waypoints. You can also use Move to set acceleration and speed for the robot arm's movement between waypoints.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

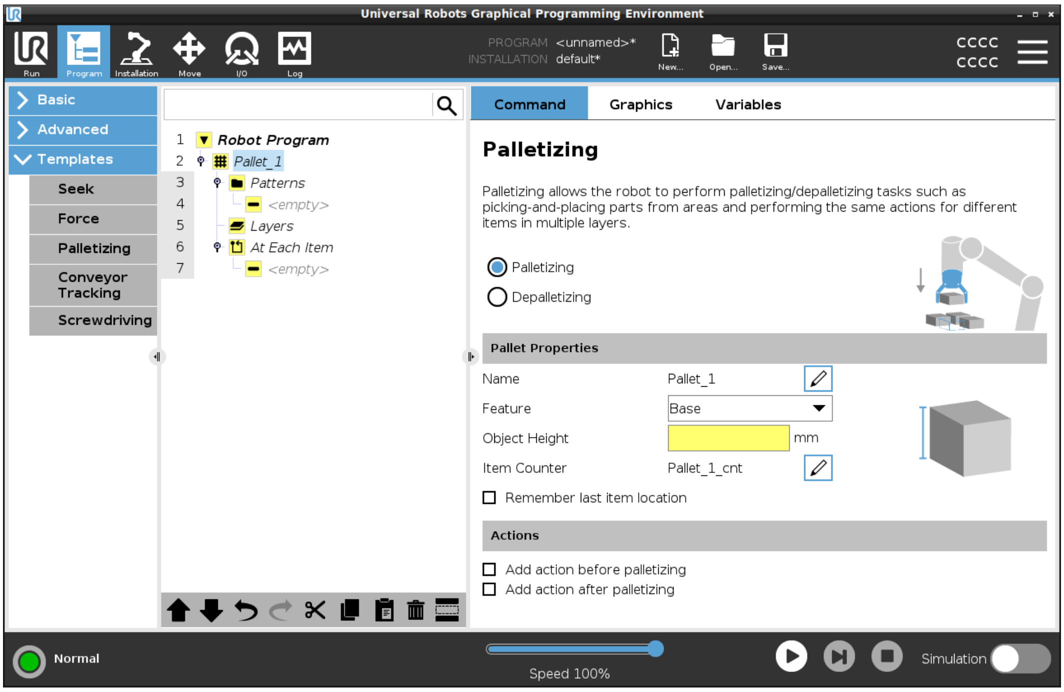
Wait

You can use the Wait command to make the robot program wait for a signal from a sensor that is attached to the control box.



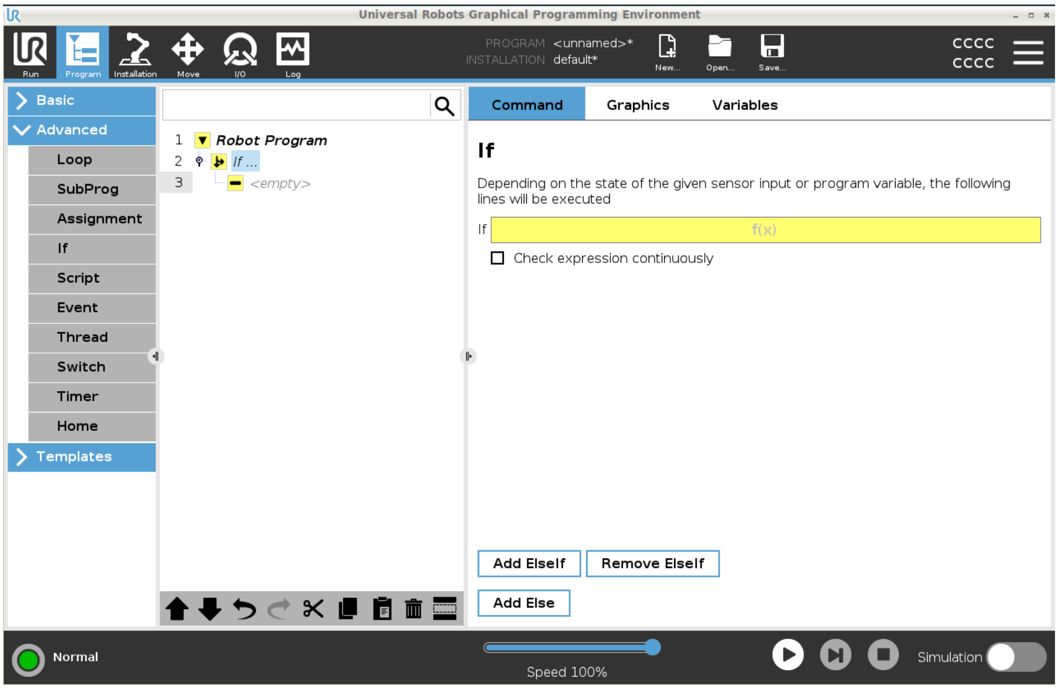
Palletizing

You can use the palletizing template to quickly create a palletizing program. The same palletizing program is also used to create a depalletizing program.



Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

If Allows you the option to add "If", "Else" and "ElseIf" conditions to a robot program.



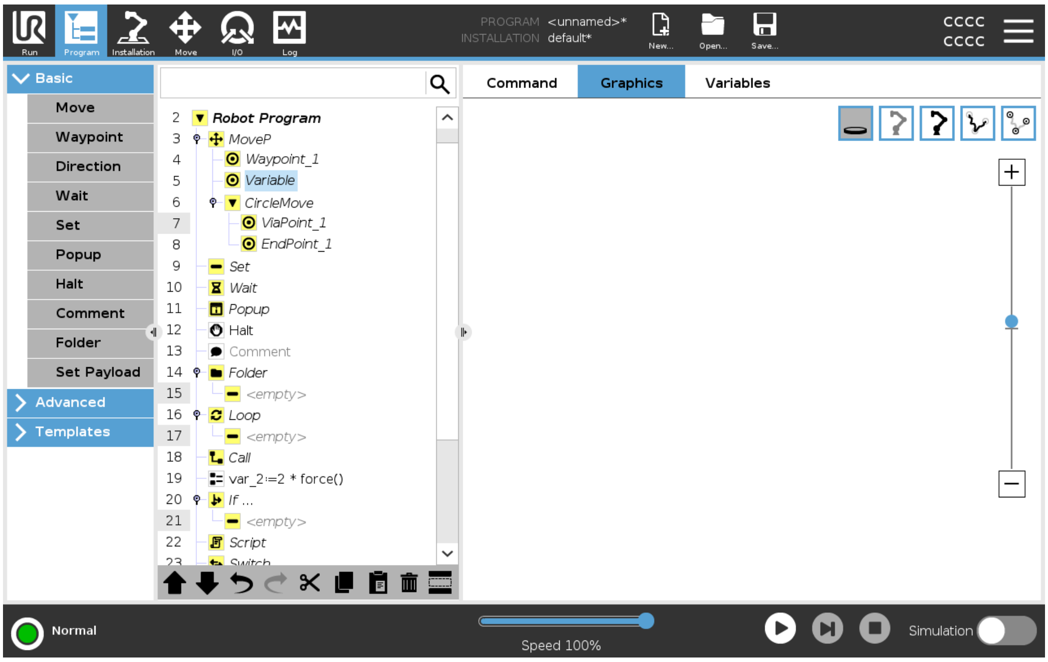
Example You can use the If command to create two different results for a robot program.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

4.2.3. Graphics Tab

Description

The Graphics tab provides a graphical representation of the running program.
The buttons in the top-right side of the Graphics tab pane can disable the graphics in 3D view.



The 3D drawing of the robot arm shows the current position of the robot arm.
The shadow of the robot arm shows the robot arm's intended path to reach the waypoint selected in the left hand side of the screen.

The path of the TCP is shown in 3D view as follows:

- The motion segments are black.
- The blend segments (transitions between motion segments) are green.
- The green dots specify the positions of the TCP at each of the waypoints in the program.

Planes

A plane is a boundary that limits the movement of the TCP. A plane can also limit the movement of a tool.
A 3D representation of the plane appears in the pane when the TCP, or tool, comes close to a plane.

- You can zoom in to the 3D view to get a better view of the robot arm, TCP or tool.
- You can use two types planes to limit TCP and tool movement.

Safety Planes

Safety planes appear in the 3D view in yellow and black. An arrow indicates which side of the plane the TCP is allowed to be positioned.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Trigger Planes Trigger planes appear in the 3D view in blue and green. A small arrow points to the side of the plane, where the Normal mode limits (see 2.5 Software Safety Modes on page 38) are active.
The tool orientation boundary limit is visualized with a spherical cone together with a vector indicating the current orientation of the robot tool. The inside of the cone represents the allowed area for the tool orientation (vector).

Limits When a program is running, the 3D view of boundary limits are disabled.
When the TCP is no longer close to any limit, the 3D view disappears.

- You can set tool orientation boundary limits that appear in the 3D view as a spherical cone together with a vector indicating the current orientation of the robot tool.

  The inside of the cone represents the allowed area for the tool orientation (vector).
- You can also set red limits that appear red in the 3D view when the TCP is in violation, or very close to violating a boundary limit.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

4.2.4. Variables Tab

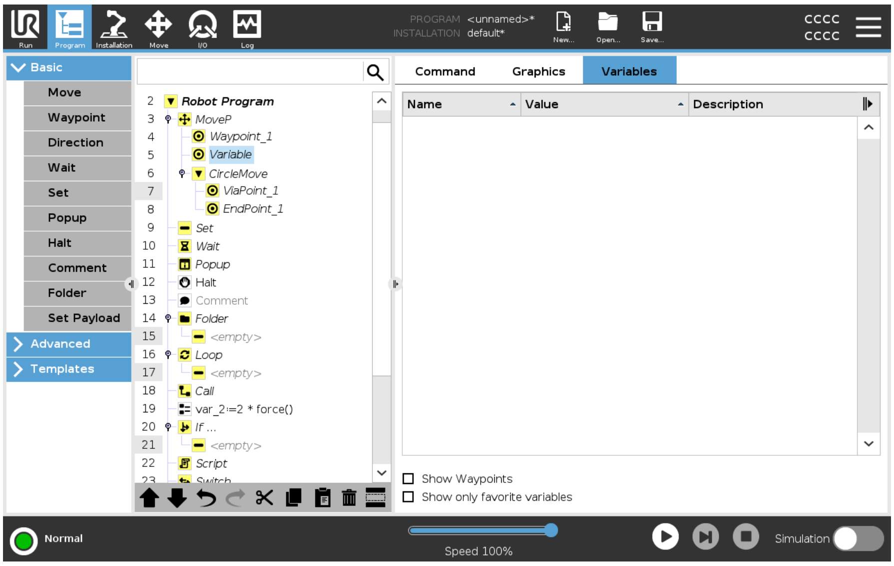
Description

The Variables tab in the Program Tab shows the live values of variables in the running program.
The variables appear as a list in the pane under the Variables tab on the right side of the screen.
Variables only appear when there is information to display and remain visible between program runs.

Show Waypoints

You can check the Show Waypoints box to display the waypoint variables in the running program.
In the Variables pane, you can also use the following options:

- Select **Show Waypoints** to show waypoint script variables in the variables list. The robot program uses script variables to store information about waypoints. Select the Show Waypoints checkbox, under Variables to show script variables in the variables list.
- Select **Show only favorite variables** to only view favorite variables on the Variables Tab. This is the same as in the Variables pane on the Run Tab (See 3 Run Tab on page 61).



Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Variable Value Types

A robot program uses variables to store and update various values during runtime. Variables only appear when there is information to display. Variable types include:

- Program variables - These are available to the running program only and their values are lost as soon as the program is stopped.
- Installation variables - These can be used by multiple programs and their names and values stay together with the robot installation (see 5.5 Installation Variables on page 182).
- Script variables - These come from script files and they can be assigned different variable types. Script variables do not appear in the Program tab or in the Installation tab. The robot program uses the script variables to store information about waypoints. You can select the Show Waypoints checkbox, under Variables, to show script variables in the variables list.

The table below list variable value types:

bool	A boolean variable whose value is either `True` or `False`
int	A whole number in the range from -2147483648 to 2147483647 (32 bit)
float	A floating point number (decimal) (32 bit)
string	A sequence of characters
pose	A vector describing the location and orientation in Cartesian space. It is a combination of a position vector `(x, y, z)` and a rotation vector `(rx, ry, rz)` representing the orientation written `p[x, y, z, rx, ry, rz]`
list	A sequence of variables

4.3. Expression Editor

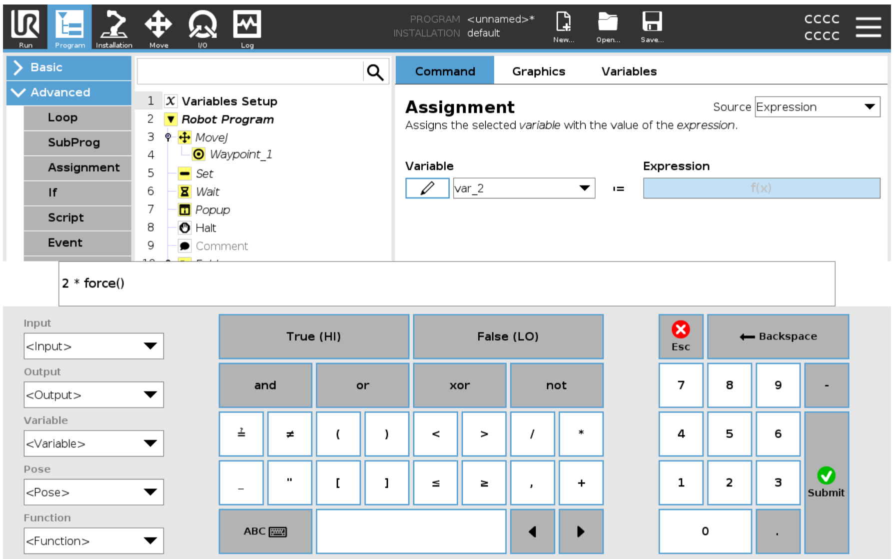
Description

The expression editor has a number of buttons and functions for inserting the special expression symbols, such as * for multiplication and ≤ for less than or equal to. Expressions are edited as text.
All defined variables can be found in the `Variable` selector, while the names of the input and output ports can be found in the `Input` and `Output` selectors. Some special functions are found in `Function`.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Expressions in Editor

When your expression is complete, you can tap the **Submit** button to check for grammatical errors. The **Esc** button exits the screen and discards all unsaved changes.



Example

```
digital_in[1]
?
=
True and analog_in[0]<0.5
```

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

4.4. Using a Selected Program Node

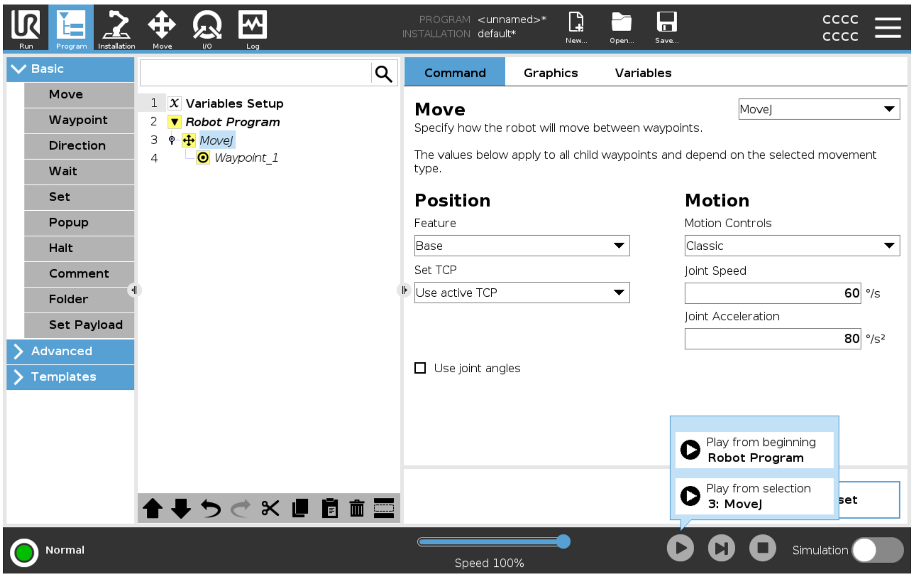
Description

You can start your robot program from any program node in the program tree. This is useful when you are testing your program.

When the robot is in Manual Mode you can allow a program to start from a selected node or you can start the entire program from the beginning.

Play From Selection

The Play button in the Footer provides options for how to start the program.
In the image below, the **Play** button is selected and **Play from Selection** is displayed.



- You can start a program only from a node in the robot Program tree. The **Play from Selection** stops if a program cannot be run from a selected node.

  The program also stops and displays an error message if an unassigned variable in encountered while playing a program from selected node.

- You can use **Play from Selection** in a subprogram. The program execution halts when the subprogram ends.
- You cannot use **Play from Selection** with a thread because threads always start from the beginning.

To play a program from a selected node

1. In the Program tree, select a node.
2. In the Footer, tap **Play**.
3. Select **Play from Selection** to run a program from a node in the program tree.

Example

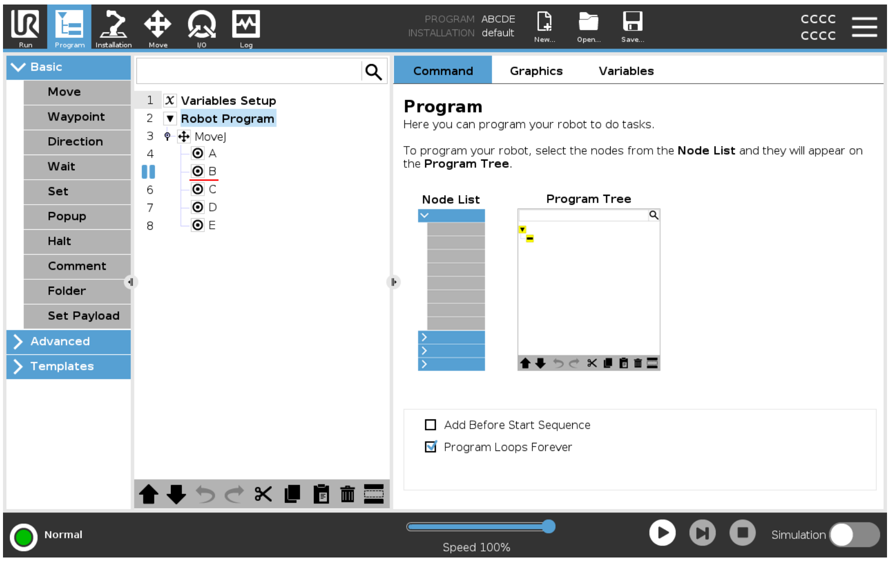
You can start a stopped program again from a specific node.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

4.5. Using Breakpoints in a Program

Description

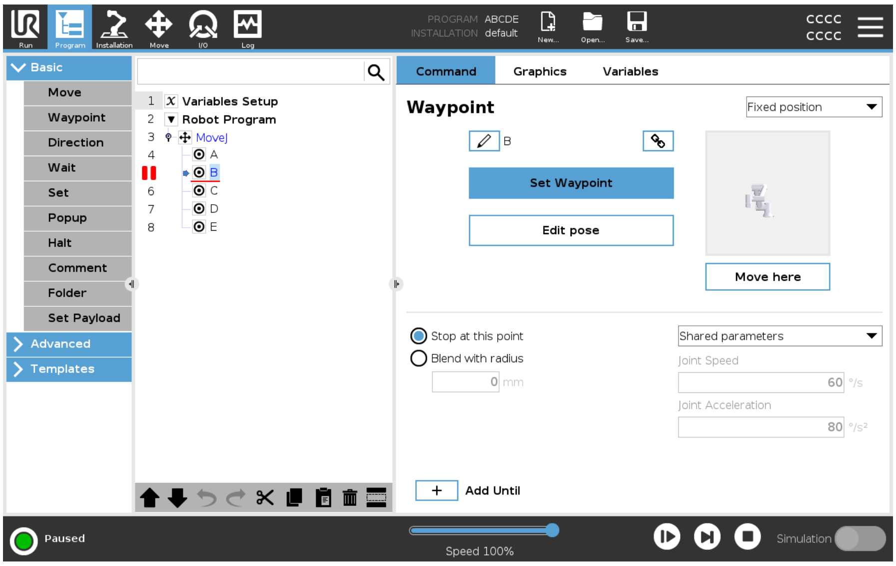
A breakpoint pauses program execution. You can use breakpoints to pause and resume a running program at a specific point. This is useful when you are inspecting robot position, variables, etc. (See Operational mode on page 30).
The image below shows a breakpoint added to a node.



Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Breakpoints

When the robot program reaches a breakpoint, it turns red. The following breakpoints remain blue.



The breakpoint appears as a red line either above or below a node. The red line appears below waypoint nodes and above all other nodes.

Most nodes in program pause before they are executed. Adding a breakpoint changes the behavior of the nodes.

- You can add a breakpoint to a waypoint to allow the program to pause at this waypoint. This means the blend in the node is ignored.
- You can add a breakpoint to an Until node to allow the program to pause when the Until condition is met. This means the blend in the node is not ignored.

Add a Breakpoint

To add a breakpoint to a program

1. In a Program tree, tap a line number to add a breakpoint.
2. The breakpoint is active until you clear the breakpoint from the robot program.

Clear a Breakpoint

To clear a breakpoint in a program

1. In a Program tree, tap a breakpoint to remove it again.
2. The robot program will run as intended.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

4.6. Single Step in a Program

Description

Single Step allows the current program to be exectued one node at a time, when the robot is in Manual mode.
(See Operational mode on page 30).

This is useful when you are checking your program for errors.

Single Step

Single Step allows the selected program node to run, then pauses at the beginning of a new node.
Single Step can only be used when the current program is paused.
If you want to use Single Step on a specific node, the node must also support Breakpoints.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

4.7. Basic program nodes

Description

Basic program nodes are used to create simple robot applications. Some basic program nodes are also used to organize your robot program and create comments in your robot program. This can be quite useful, if it is large robot program.

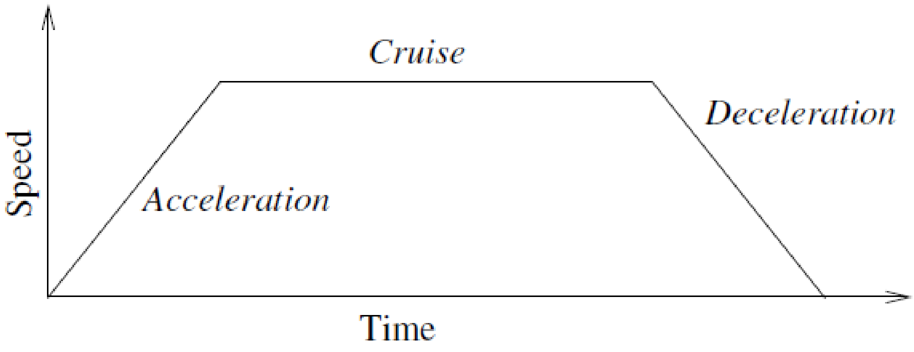
4.7.1. Move

Description

The Move command allows the robot to move from point A to point B. How the robot moves is important to the task the robot is performing. When you add a Move to your program tree, the Move pane appears to the right of the screen. The options in the Move pane allow you to configure a Move and the attached waypoint.

Speed settings

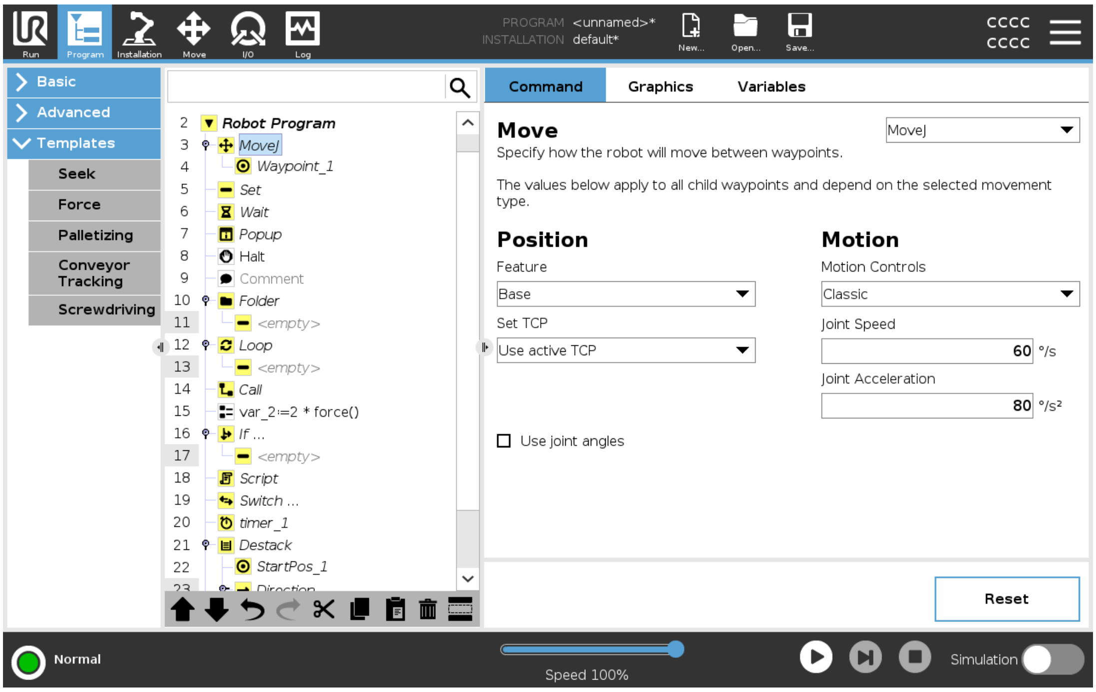
The shared parameters that apply to the movement types are the maximum joint speed and joint acceleration.



14.1: *Speed profile for a motion. The curve is divided into three segments: acceleration, cruise and deceleration. The level of the cruise phase is given by the speed setting of the motion, while the steepness of the acceleration and deceleration phases is given by the acceleration parameter.*

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

OptiMove is a motion control option that specifies the speed and acceleration of the robot, while maintaining hardware limits. This means the optimal motion of the robot does not exceed the desired limits.
So 100% is the maximum speed percentage and acceleration within the hardware limits.



Move commands

The Move command controls the robot's motion via waypoints.
Waypoints are automatically added when you add Move commands to a program.
You can also use Moves to set acceleration and speed for the robot arm's movement between waypoints.

The robot moves using four Move commands as described in the following sections:

- MoveJ below
- MoveL on the facing page
- MoveP on page 90
- MoveCircle on page 91

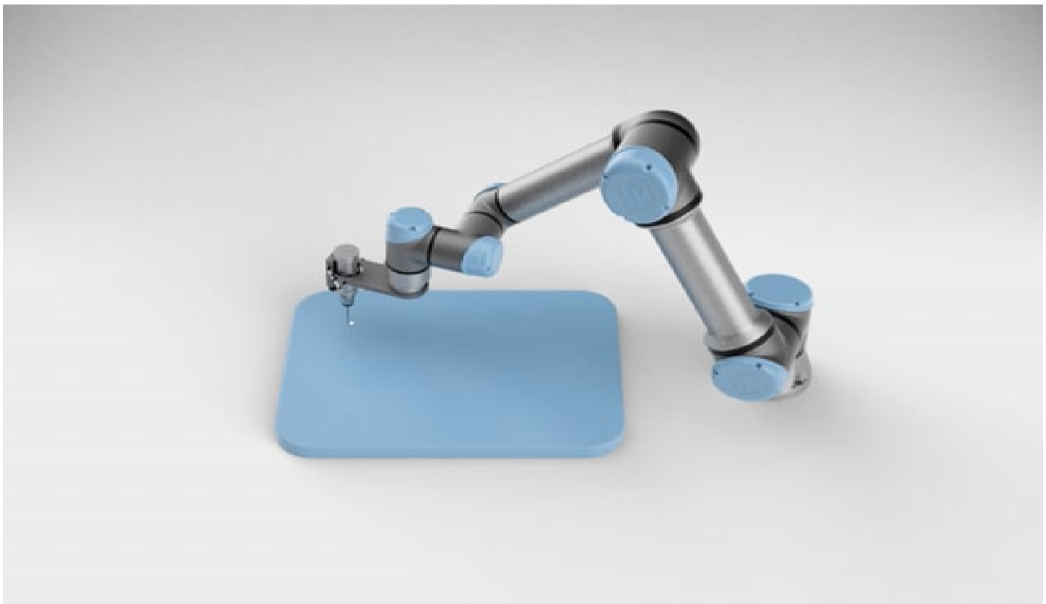
MoveJ

The MoveJ command creates a movement from point A to point B that is optimal for the robot. The movement may not be a direct line between A and B, but optimal for the start position of the joints and the end position of the joints.
MoveJ makes movements that are calculated in the robot arm joint space. Joints are controlled to finish their movements at the same time. This movement type results in a curved path for the tool to follow.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

To add a MoveJ

1. In your robot program tree, select the place where you wish to add a Move.
2. Under Basic, tap **Move** to add a Move node together with a waypoint.
3. Select the move node.
4. Select MoveJ in the drop-down menu.



To add a MoveJ with OptiMove

1. In your robot program tree, select the desired move node or waypoint node.
2. In the Motion Controls dropdown menu, select **OptiMove**.
3. Use the slider to set the speed.
4. You can select **Scaled acceleration** to keep the settings linked.

   You can deselect **Scaled acceleration** to modify the settings independently .

Using Use joint angles

The Use joint angles option is an alternative to the 3D pose when you are using MoveJ to define a waypoint.

Waypoints defined using the Use joint angle are not changed when a program is moved between robots. This is useful if you are installing your program in a new robot.

Using Use joint angles makes the TCP options and feature unavailable.

MoveL

The MoveL command creates a movement that is a direct line from point A and point B. MoveL moves the Tool Center Point (TCP) linearly between waypoints. This means that each joint performs a more complicated motion to keep the tool on a straight line path.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

To add a MoveL

Adding a MoveL is similar to adding a MoveJ.

1. In your robot program tree, select the place where you wish to add the MoveL.
2. Under Basic, tap Move and select MoveL from the drop-down menu.

Adding a MoveL with OptiMove is also similar to adding a MoveJ with OptiMove.
Once you select the node, simply navigate to the Motion Controls dropdown and select OptiMove.

MoveP

The MoveP command creates a movement with a constant speed between the waypoints. Blend between waypoints is enabled to ensure constant speed.

To add a MoveP

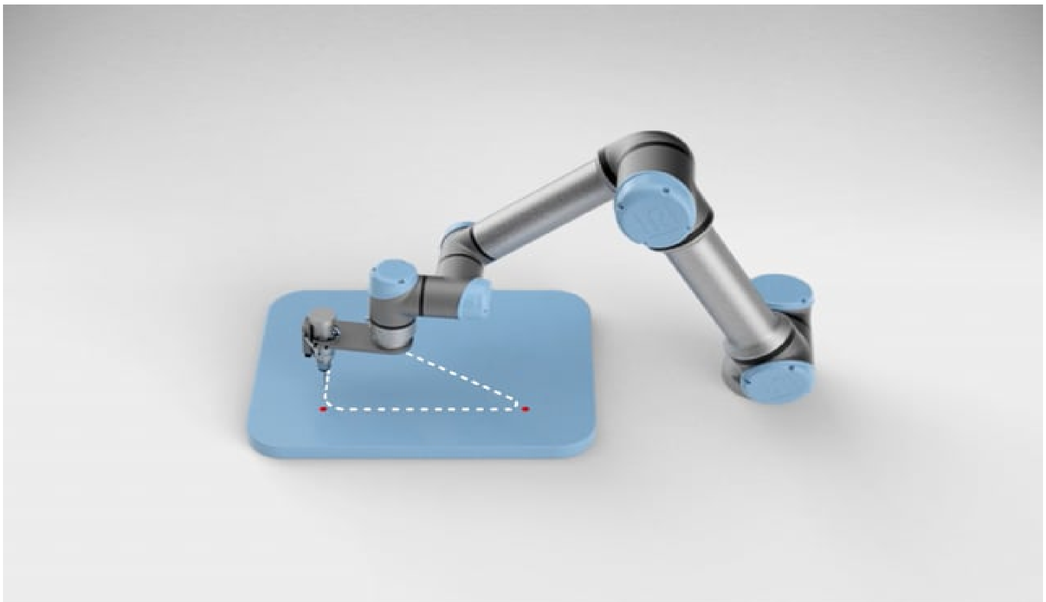
Adding a MoveP is similar to adding a MoveJ and a MoveL.

1. In your robot program tree, select the place where you wish to add the MoveP.
2. Under Basic, tap Move and select MoveP from the drop-down menu.

Adding a MoveP with OptiMove is also similar to adding a MoveJ with OptiMove.
Once you select the node, simply navigate to the Motion Controls dropdown and select OptiMove.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Detail MoveP moves the tool linearly with constant speed with circular blends, and is intended for some process operations, like gluing or dispensing. The size of the blend radius is by default a shared value between all the waypoints. A smaller value will make the path turn sharper whereas a higher value will make the path smoother. While the robot arm is moving through the waypoints with constant speed, the robot control box cannot wait for either an I/O operation or an operator action. Doing so might stop the robot arm's motion, or cause a robot stop.



MoveCircle The MoveCircle command creates a circular movement, by creating a half circle. You can only add CircleMove via a MoveP command.

To add a MoveCircle

1. In your robot program tree, select the place where you wish to add a Move.
2. Under Basic, tap **Move**.

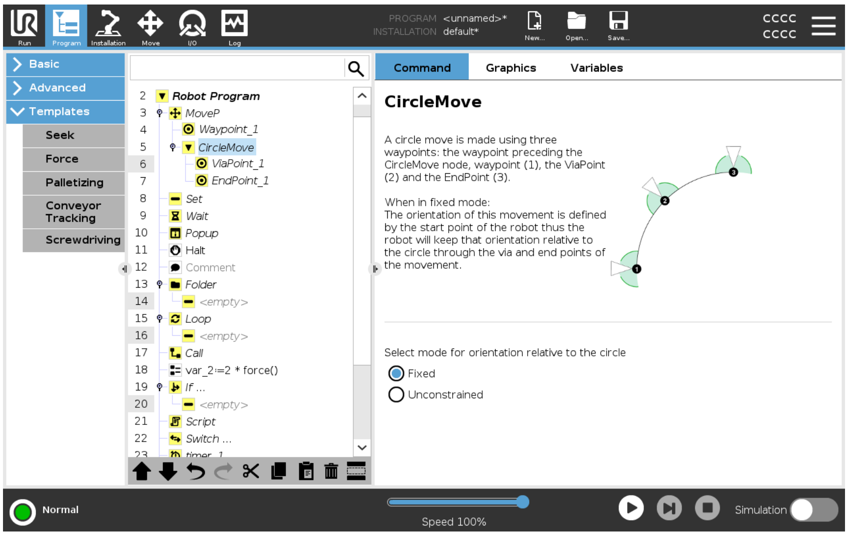
   A waypoint is added to the robot program together with the Move node.
3. Select the move node.
4. Select the MoveP from the drop-down menu.
5. Tap **Add circle move**
6. Select the orientation mode.

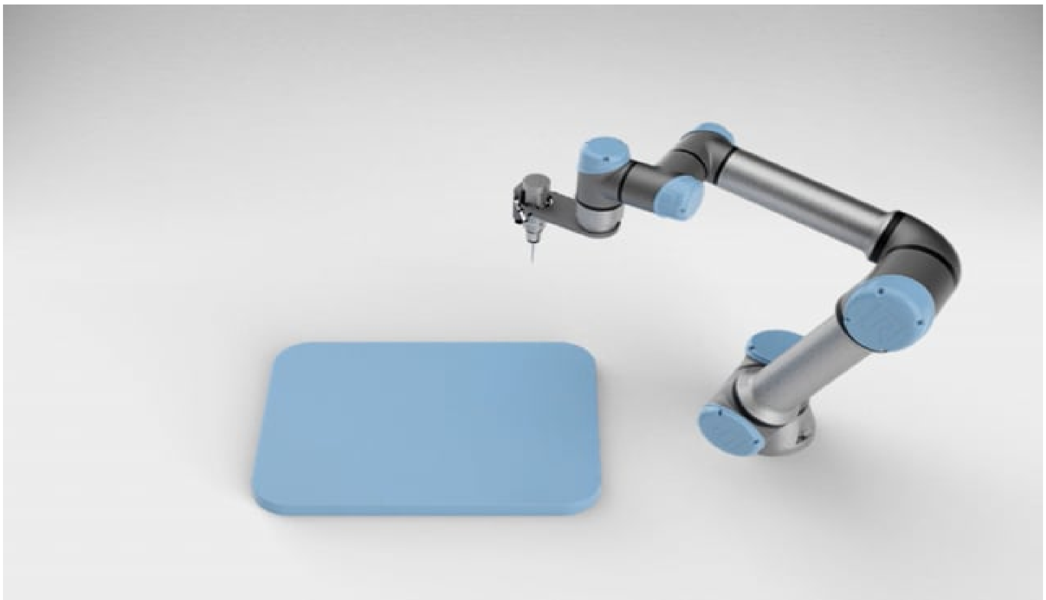
Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Detail

The robot starts the circular movement from its current position, or start point, and moves through a ViaPoint specified on the circular arc, to an EndPoint that completes the circular movement.
A mode is used to calculate tool orientation, through the circular arc.
The mode can be:

- Fixed: only the start point is used to define the tool orientation.
- Unconstrained: the start point transforms to the EndPoint to define tool orientation.





Using Set TCP

Use this setting, if you need to change TCP during the robot program execution. This is useful if you need to manipulate different objects in the robot program.
The way the robot moves is adjusted depending on which TCP is set as an active TCP.
Ignore Active TCP allows this movement to be adjusted in relation to the Tool Flange.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

To set the TCP in a Move

1. Access the Program Tab screen to set the TCP used for waypoints.
2. Under Command, in the drop down menu on the right select the Move type.
3. Under Move, select an option in the **Set TCP** drop down menu.
4. Select **Use active TCP** or select **a user defined TCP**.
   You can also choose **Ignore Active TCP**.

Feature

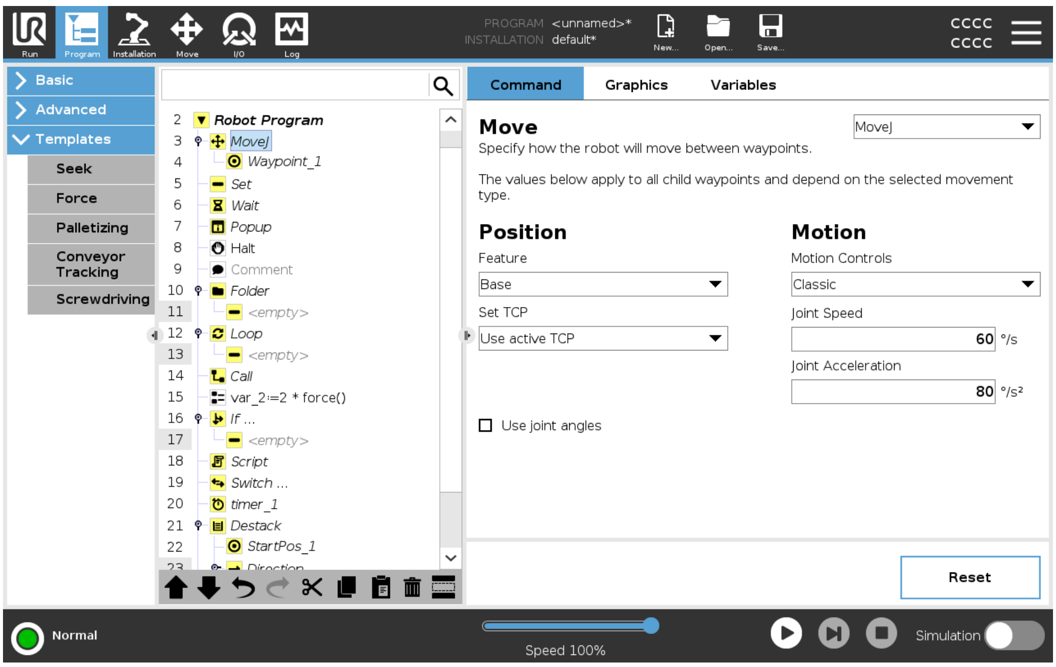
You can use Feature between waypoints for the program to remember the tool coordinates. This is useful when you are setting the waypoints (see 5.16 Features on page 197).

You can use Feature in the following circumstances:

- Feature has no effect on relative waypoints. The relative movement is always performed with respect to orientation of the **Base**.
- When the robot arm moves to a variable waypoint, the Tool Center Point (TCP) is calculated as the coordinates of the variable in the space of the selected feature. Therefore, the robot arm movement for a variable waypoint changes if another feature is selected.
- You can change a feature's position while the program is running by assigning a pose to its corresponding variable.

Shared Parameters in a Move Command

The shared parameters in the bottom right corner of the Move screen apply to the movement from the previous position of the robot arm to the first waypoint under the command, and from there to each of the following waypoints.
The Move command settings do not apply to the path going *from* the last waypoint under that Move command.



Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

4.7.2. Waypoints

Description

Waypoints are one of the most central parts of a robot program, telling the robot arm where to go one movement at a time.

See the following sections about using a waypoint:

- Fixed Waypoint on page 99
- Relative Waypoint on page 102
- Variable Waypoint on page 104
- Configuring Waypoints on page 96
- Set Fixed Waypoint on page 100
- Blending on page 105
- Add Until on page 112

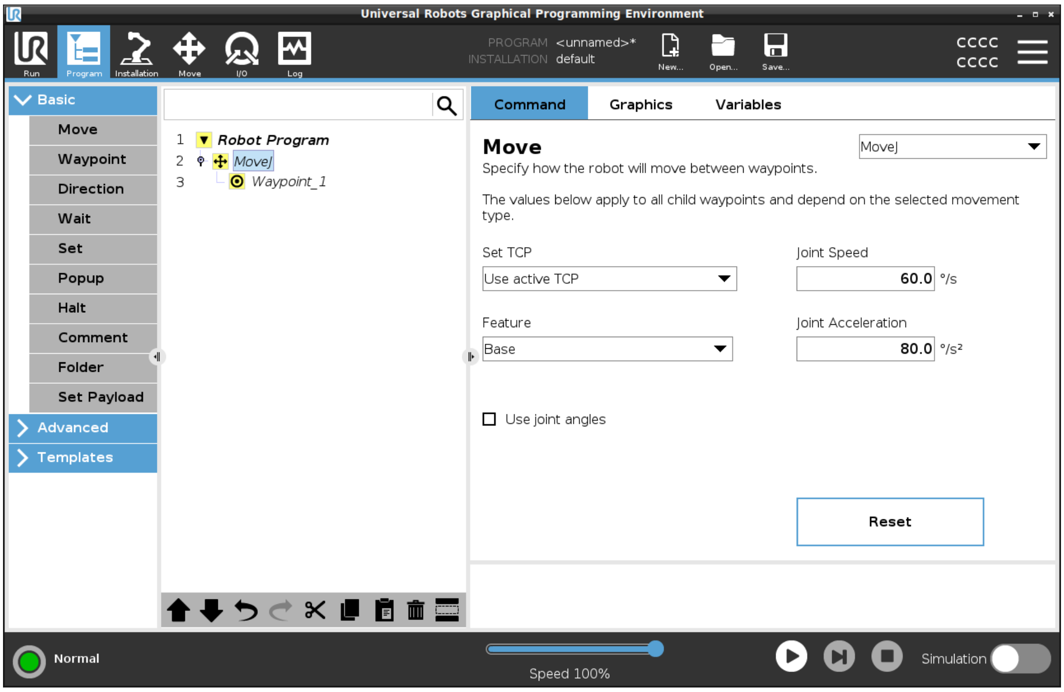
Add Waypoints

A waypoint accompanies a Move, so adding a Move is required for the first waypoint.

Add a waypoint to a robot program

1. In your Robot Program, select the place where you wish to add a Move.
2. Under Basic, tap **Move**.

   A waypoint is added to the robot program together with the Move node.

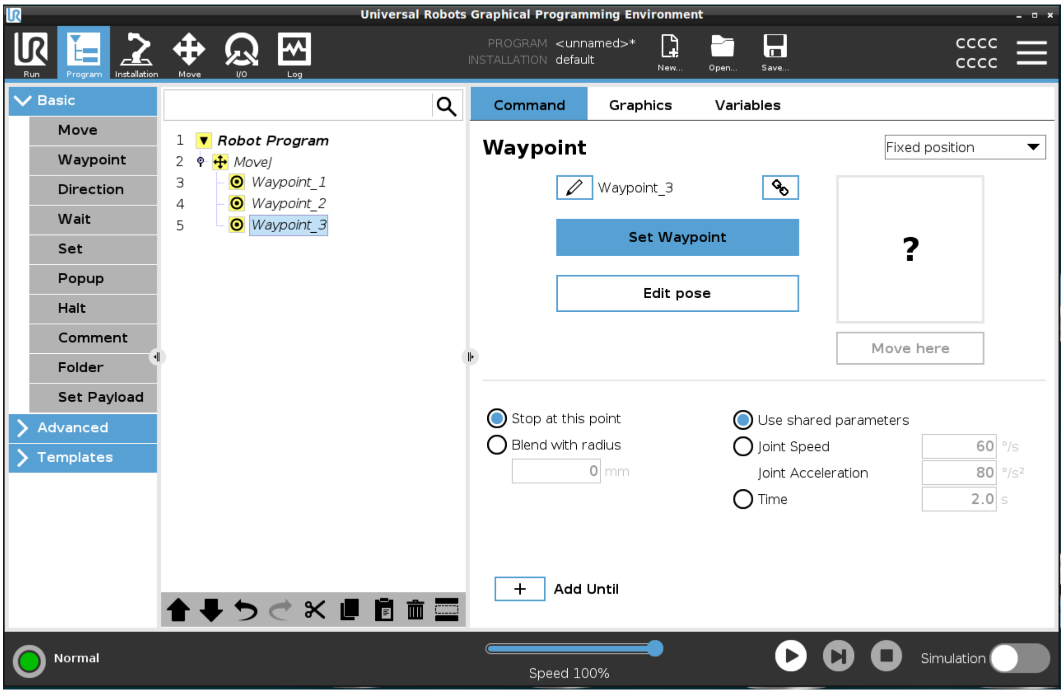


Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Add additional waypoints to a Move or Waypoint

1. In your Robot Program, select a Move node or Waypoint node.
2. Under Basic, tap **Waypoint**.

   The additional waypoint is added in the Move node. This waypoint is part of the Move command.



The additional waypoint is added under the waypoint that you selected in the robot program.

Detail

Using a waypoint means applying the taught relationship between the feature and the TCP from the Move command. The relationship between the feature and the TCP, applied to the current selected feature, achieves the desired TCP location. The robot calculates how to position the arm to allow the current active TCP to reach the desired TCP position.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Configuring Waypoints

Description

Waypoints can be configured in different ways depending on the setup, application and position of the waypoint in the robot program.

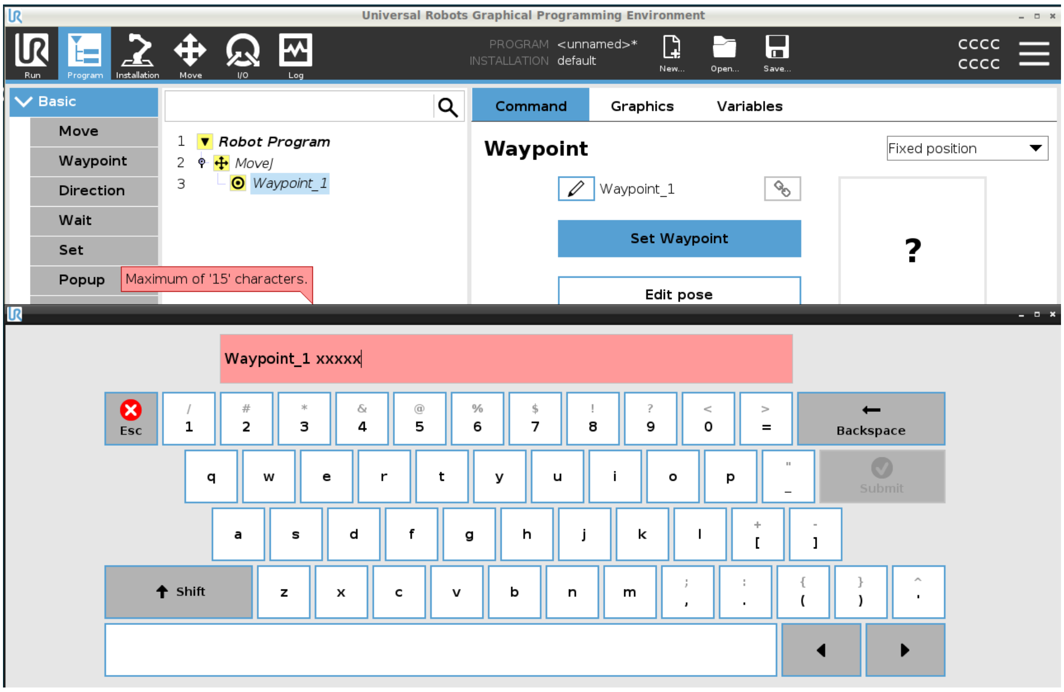
- Naming Waypoints below
- Linking Waypoints on the facing page
- Stop at this point on the facing page
- Blend with radius on the facing page
- Use Shared Parameters on the facing page
- Joint Speed / Acceleration on page 98
- Time on page 98
- Add Until on page 98

Naming Waypoints

Waypoints automatically get a unique name when you add them to the robot program. The user can change the name of a waypoint.

To name a waypoint

1. Make sure you are positioned on the Command tab on the right side of the screen
2. Select the waypoint you wish to rename
3. Tap the **Rename** button 
4. Type the new name
5. Tap **Submit** to save the new name



Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Tip

In a large robot program, you can often have more than 50 waypoints, so create a naming rule that makes sense to you.

Linking Waypoints

By selecting the link icon, waypoints are linked and share position information.

Detail

Other waypoint information such as blend radius, tool/joint speed and tool/joint acceleration is configured for individual waypoints even though they may be linked.

Stop at this point

You configure the robot program to stop at this waypoint. The robot will decelerate moving towards this point and continue to the next waypoint.

Use Stop at this point

1. Select the waypoint you wish to modify
2. On the right side of the screen, select Stop at this point

Tip

This is useful to make the robot move to an exact position.

Use case

When you wish to move the robot away from a welding or machine tending application.

Blend with radius

You can add a blend radius for blending the robot arm's movement between waypoints. See this link for an in-depth description of blending.

Use Blend with radius

When you add a blend to a waypoint, the transition between waypoints becomes more fluid and efficient. This will make the robot arm move more smoothly, but it is not applicable in every robot arm movement.

1. Select the waypoint you wish to modify
2. On the right side of the screen, select Blend with radius
3. Add a number [mm] to define the blend radius
4. Tap **Submit** to save the number

Tip

If you are using multiple waypoints to guide the transition between two points you can blend the waypoints in between to make the overall movement smoother and more efficient.

Use Shared Parameters

This is the default setting for the waypoint.
The settings are copied from the parent Move command.

Use case

This is used when the joint speed and joint acceleration is NOT changed in the parent Move command.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Joint Speed / Acceleration

You can modify the speed and acceleration of joints for each waypoint in a Move command.

Modify joint speed or acceleration

1. Select the waypoint you wish to modify
2. On the right side of the screen, set the joint speed or acceleration by selecting
   Joint Speed
   Joint Acceleration
3. Add the value for speed
4. Add the value for acceleration

Use case

If you create a palletizing program, then you can make the waypoint move more slowly as the program places the objects, and move faster when moving to pick up another object.

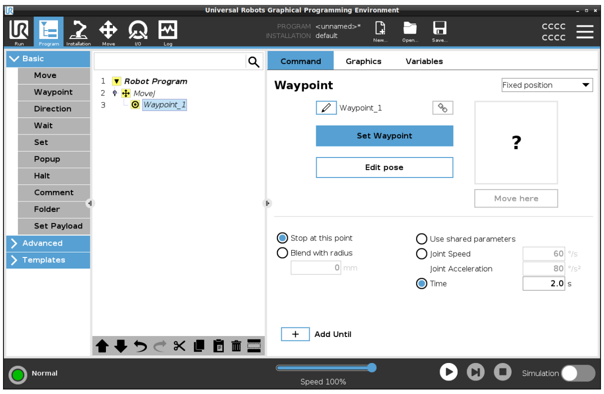
Time

You can change the amount of time that it takes to reach this waypoint.
The maximum is 21.600 seconds/6 hour.

Add time in seconds

1. Select the waypoint you wish to modify
2. On the right side of the screen, select Time
3. Add a number in [s]
4. Tap **Submit** to save the number

It now takes [s] to move the robot arm to this waypoint.

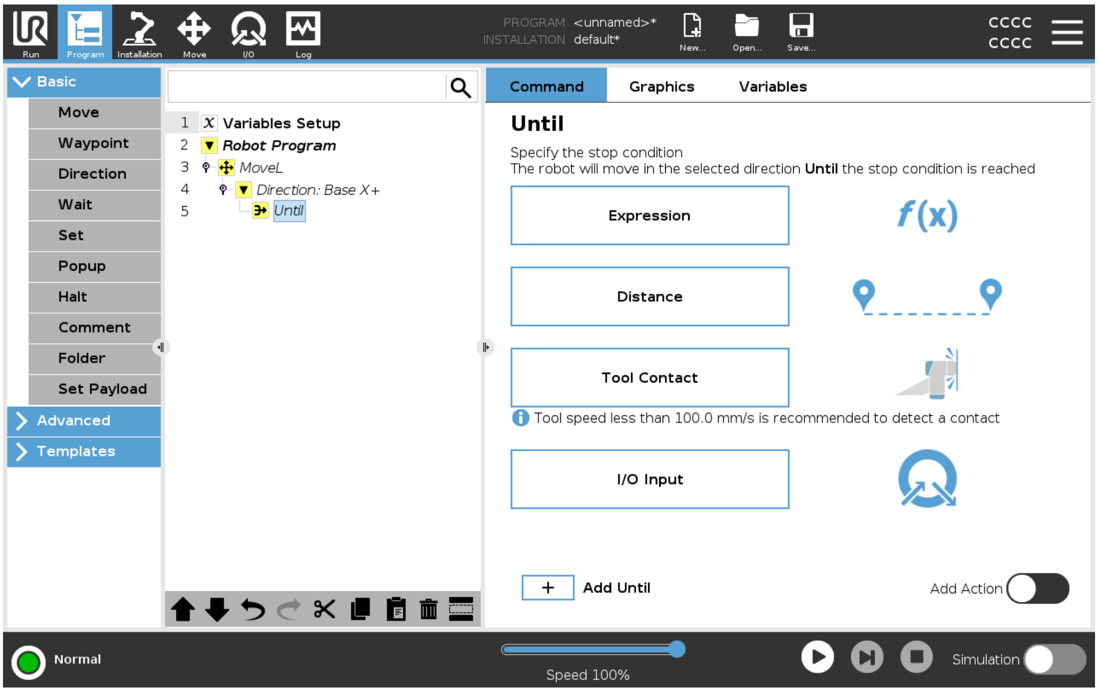


Add Until

Add Until will give you a number of different options for configuring your waypoint.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Steps Please see the link Add Until on page 112 for step by step instructions.



Fixed Waypoint

Description A fixed waypoint is the standard waypoint used in PolyScope.
It is a fixed location, relative to a set coordinate system (the robot's base by default), that the robot will move back to when prompted.
A fixed position waypoint is taught by physically moving the robot arm to the position. You can move the robot with the 1.5 Freedrive on page 16 or with the 6 Move Tab on page 214 in PolyScope.

When to use:

- If the robot must move to a specific location (to move around equipment or move to a specific location for operation like picking or placing).
- When you create an entire robot program using fixed waypoints relative to a location, i.e. relative to the robot base. You can move the robot, and redefine the location of base of the robot, then the robot program will work on the new location.

When NOT to use:

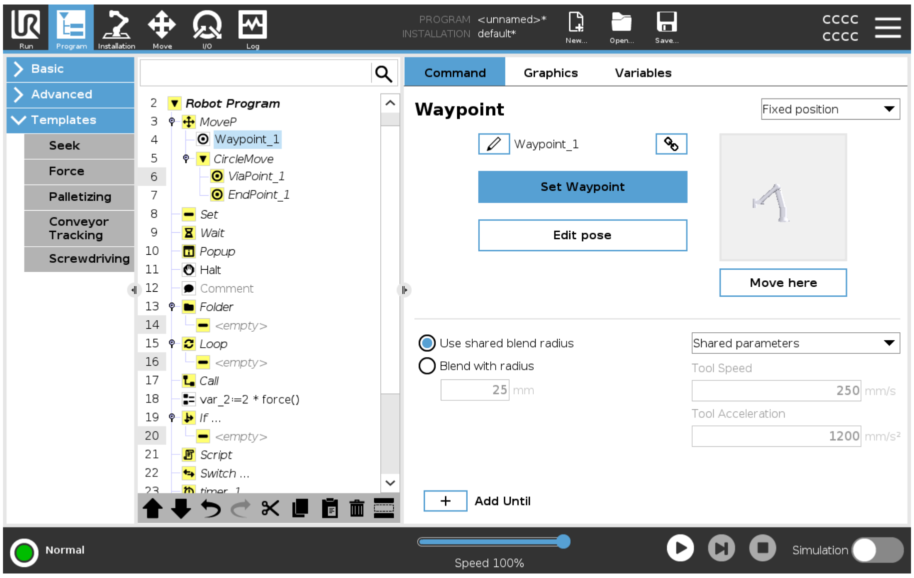
- When you want the location of the waypoint to be dynamic. Fixed waypoints are static positions and only change when purposely updated or when they're relative to a coordinate system.

Fixed Waypoint You can add a fixed waypoint to your robot program, when you add a Move command. There is no difference, if the Move command is a MoveJ, a MoveL or a MoveP.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

To add a fixed waypoint to a robot program

1. In the Program tab, tap Basic.
2. Add a move command
3. Select the automatically generated waypoint
   a. Notice the selector is automatically set to **Fixed position ▼**
4. Tap **Set Waypoint** to access the Move tab
5. Set the waypoint
6. Tap the check mark  to save the waypoint and return to the Program tab



Example When you are removing a work piece from a chuck in a machine tending application, make the waypoint a fixed position to avoid any contact with equipment.[1]

Set Fixed Waypoint

Description Setting a fixed waypoint means that you move the robot arm to the position of the specific waypoint that you want to set. Then the waypoint is "set" in the robot program as a position in the overall movement of the robot arm.

You can use the freedrive mode, or you can move the robot arm with the manual controls in the Move tab. See: 4.7.1 Move on page 87

[1]In machine tending, the chuck holds the tool performing work on a workpiece. For example, the chuck is the part of a drill that attaches to the drill bit.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Setting a Fixed Waypoint

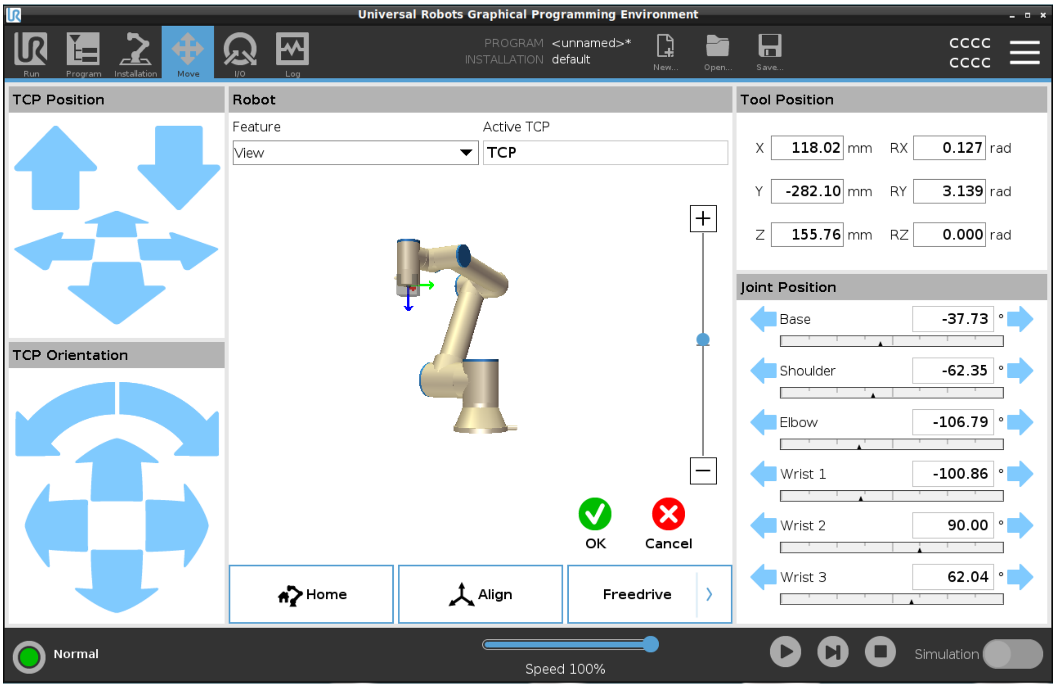
To add a fixed waypoint to a robot program

1. In your Robot Program, select the place or node where you wish to add a fixed waypoint.
2. Under Basic, tap **Waypoint**.
3. Select your newly added waypoint in the robot program.
4. Tap 

.

How you move the robot arm to set the waypoint depends very much on your robot program, the application what you are creating, the TCP configuration, etc.
There is no one best way to set the waypoint.

- If you want to move the robot arm so that the TCP moves linearly, use the TCP arrows.
- If you want to move the robot arm with Freedrive, hold the Freedrive button.
- If you know the exact coordinate location for the next waypoint, then you can use the tool position to move the robot arm.
- If you want to move a specific joint, use the Joint Position values



Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Relative Waypoint

Description

A relative waypoint is created by defining two waypoints. These two waypoints determine the distance and direction the relative waypoint should move
The relative waypoint can be defined to the robot arm’s previous position, such as “two centimeters to the left”.
The second waypoint is dependent on the first.
This waypoint can be created when adding the relative waypoint. It can also be a previously defined waypoint, such as a fixed waypoint.

When to use:

- When using a BeforeStart to move the robot straight up from any position. For example, if the robots is stopped in a position near parts.
- If the first waypoint is relative and you press play. You don't need to move the robot into position to start the program.
- When using SubPrograms to make repeatable movements at different locations around the robot. For example, screwdriving at multiple locations: move down 50mm, turn screwdriver on/off, move up 50mm.

When not to use:

- When a specific location must be reached repeatably.
- When a constant path is desired.

Example: Add relative waypoint

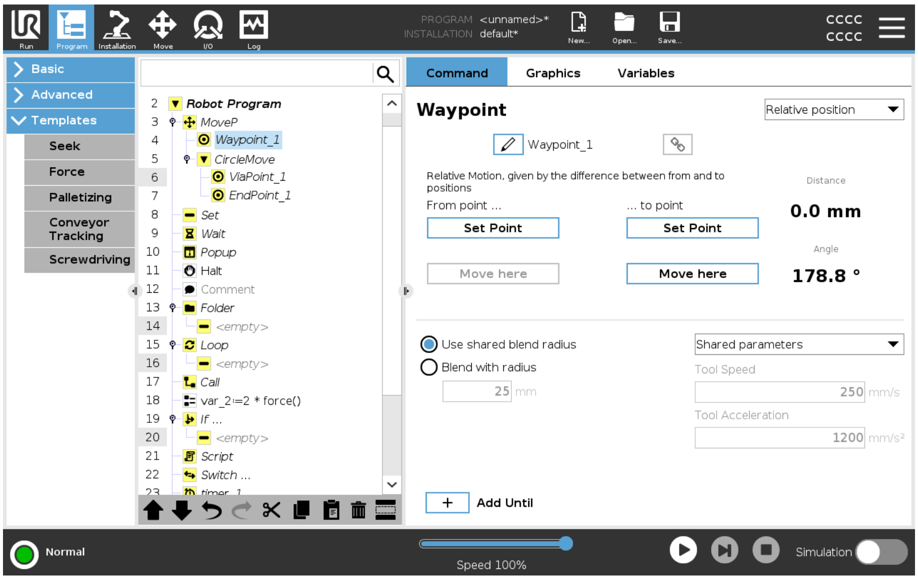
To move the robot 20 mm along the z-axis of the tool:

```
var_1=p[0,0,0.02,0,0,0]
Movel
Waypoint_1 (relative position):
Use variable=var_1, Feature=Tool
```

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Add a relative waypoint to a robot program

1. In the Program tab, select the Basic menu
2. Tap **Waypoint**
   a. Notice the selector is automatically set to Fixed position
   b. Change the waypoint to Relative position.
3. Set the first waypoint (from point...)
4. Set the second waypoint (...to point)



Detail

The movement between two relative waypoints is always the shortest path for the robot depending on the move type.
The distance for relative waypoints refers to the Cartesian distance between the TCP in the two positions. The angle states how much the TCP orientation changes between the two positions, or more precisely, the length of the rotation vector describing the change in orientation.
It does not matter where the relative waypoint position was located around the robot, before the program moved into the relative waypoint.
As soon as PolyScope moves to the relative waypoint in the program tree, the robot moves from its current position, to the distance and in the direction the relative waypoint has saved.
Repeated relative positions can move the robot arm out of its workspace.

Use Case: Welding and changing welding items

If you have a welding procedure, and you need to weld a seam around a rectangle, you can define the first corner with a fixed waypoint, and then make the robot and welding tool hit the remaining three corners using relative waypoints.
The first waypoint will start the welding seam around the rectangle, and the relative waypoints will finish the remaining corners.
Then if you need to weld something that is still a rectangle, but larger or smaller, then you can change the distance of the relative waypoints, and quickly modify the robot program.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Variable Waypoint

Description

A variable waypoint moves to a position determined by a variable in PolyScope.
The variable must be in the pose URScript format, p[x, y, z, rx, ry, rz], allowing a single waypoint in the Program Tree to be updated by changing the X, Y, Z, RX, RY, or RZ value without manually resetting the waypoint.

When to use:

- If combined with scripting elements.
- When receiving data from external devices for positioning like cameras, etc.

When not to use:

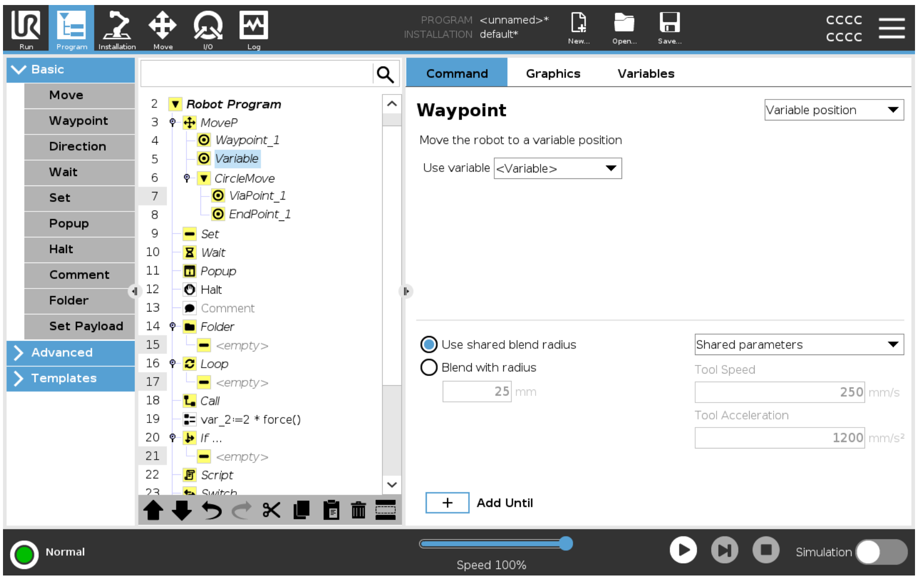
- When a specific location must be reached repeatedly.
- If you are not using variables or external devices.
- If a waypoint must be moved manually, or redefined by either jogging or freedrive.

Add a variable waypoint

Add a variable waypoint that can be modified during run through of the robot program.

Add a variable waypoint to a robot program

1. In the Program tab, hit the basic menu
2. Add a move command
3. Select the automatically generated waypoint
   a. Notice that the selector is automatically set to Fixed position.
   b. Change the selector to Variable position.
4. Select the variable from the drop-down menu.



Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Detail

A waypoint with the position given by a variable, in this case *calculated_pose*.

The variable has to be a *pose* such as var=p[0.5,0.0,0.0,3.14,0.0,0.0].

The first three are *x,y,z* and the last three are the orientation given as a *rotation vector* given by the vector *rx,ry,rz*.
The length of the axis is the angle to be rotated in radians, and the vector itself gives the axis about which to rotate.
The position is always given in relation to a reference frame or coordinate system, defined by the selected feature.

If a blend radius is set on a fixed waypoint and the waypoints preceding and succeeding it are variable or if the blend radius is set on a variable waypoint, then the blend radius will not be checked for overlap (see Blending below).
If, when running the program, the blend radius overlaps a point, the robot will ignore it and move to the next one unless it is part of a smooth motion or OptiMove.

Blending

Description

Blending enables the robot to smoothly transition between two trajectories, without stopping at the waypoint between them.
Therefore, your robot program will run faster.

See the following sections for additional information on blending:

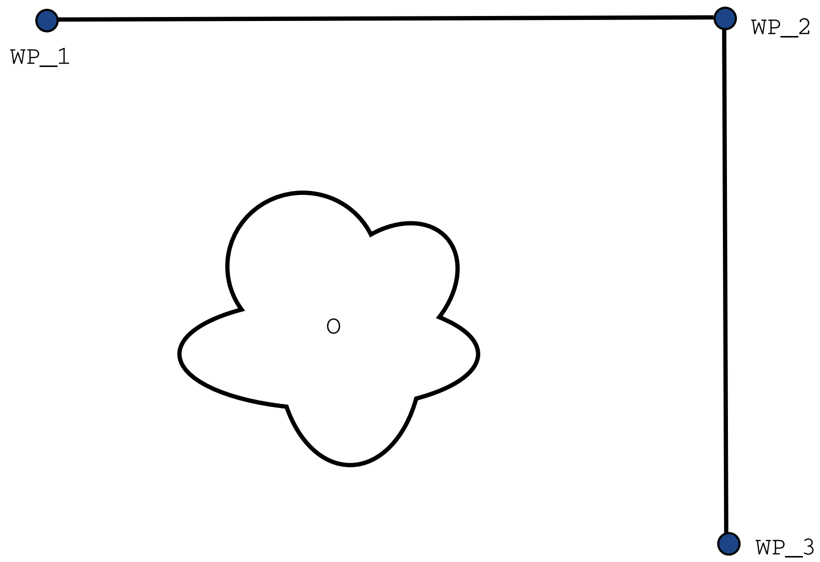
- Blend Parameters on page 107
- Blend Trajectories on page 109
- Conditional Blend Trajectories on page 110

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Example: Blending in a pick and place application

The robot starts at waypoint 1 (WP_1), and it needs to pick up an object at waypoint 3 (WP_3).
To avoid collisions with the object and other obstacles , the robot must avoid (O) by using waypoint 2 (WP_2).

Three waypoints are introduced to create a path that fulfills the requirements.



14.1: *(WP_1): initial position, (WP_2): via point, (WP_3): pick up position, (O): obstacle.*

Without configuring other settings, the robot will make a very brief but full stop at each waypoint, before continuing the movement.
For this task a stop at (WP_2) is not optimal since a smooth turn would require less time and energy while still fulfilling the requirements. It is even acceptable that the robot does not reach (WP_2) exactly, as long as the transition from the first trajectory to the second trajectory happens near this position.
The stop at (WP_2) can be avoided by configuring a blend for the waypoint.
This allows the robot to calculate a smooth transition into the next trajectory.

The primary parameter for the blend is a radius.

When the robot is within the blend radius of the waypoint, it can start blending and deviate from the original path. This allows for faster and smoother movements, as the robot does not need to decelerate and re-accelerate.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Blend Parameters

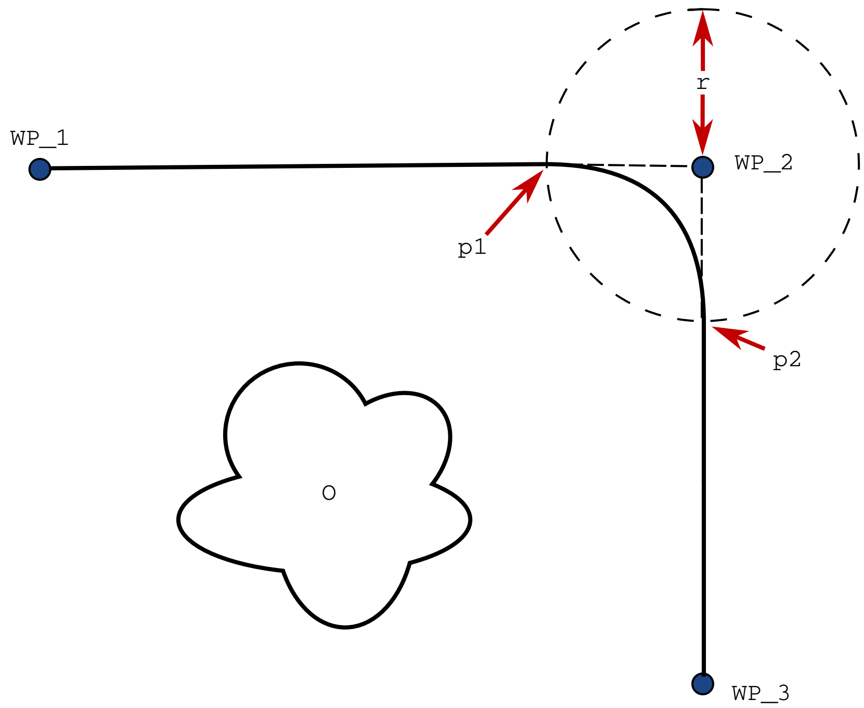
Description

Blending enables the robot to smoothly transition between two trajectories, without stopping at the waypoint between them.
Blending makes your robot program run faster because, when you create a smooth transition between trajectories, you avoid slowing down an acceleration between trajectories.

Blend parameters

Apart from the waypoints, multiple parameters will influence the blend trajectory.

- the blend radius `(r)`
- the initial and final speed of the robot (at positions `p1` and `p2` , respectively)
- the movement time (e.g. if setting a specific time for a trajectory this will influence the initial/final speed of the robot)
- the trajectory types to blend from and to (`MoveL`, `MoveJ`)

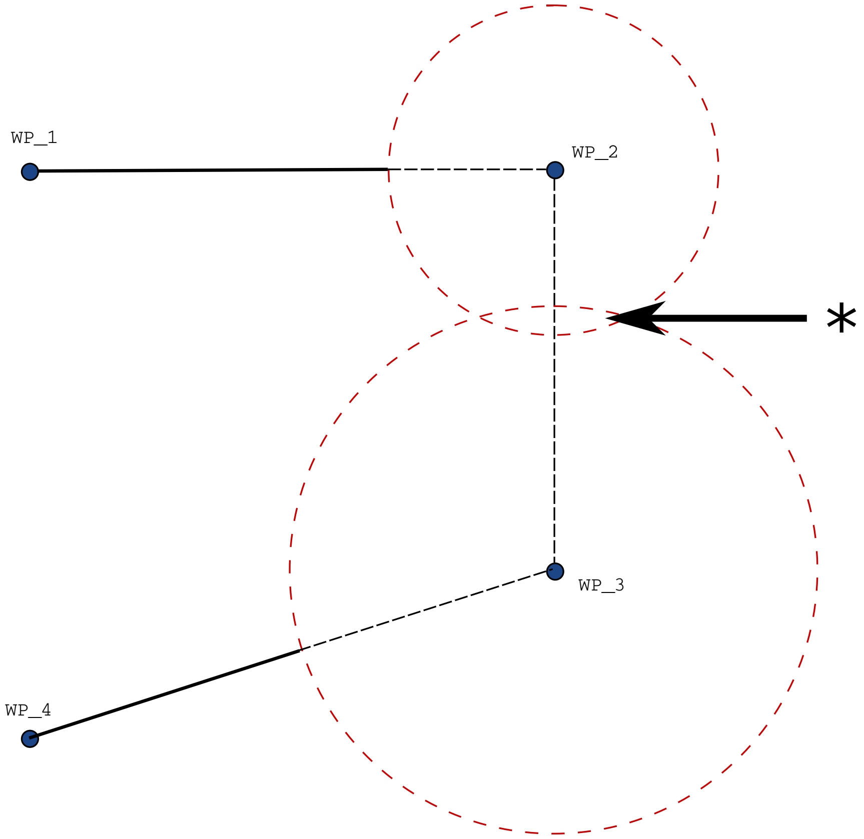


14.2: *Blend over `(WP_2)` with radius `(r)`, initial blend position at `p1` and final blend position at `p2`. `(O)` is an obstacle.*

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

If a blend radius is set, the robot arm trajectory blends around the waypoint, allowing the robot arm not to stop at the point.

Blends cannot overlap, so it is not possible to set a blend radius that overlaps with the blend radius of a previous or following waypoint.
If there is overlap of two blends, there will be a warning logged in the Log tab. See the *Log tab* for more information.
For smooth motions and OptiMove, the blend radii will be dynamically adjusted and no warning will be logged.



14.3: *Blend radius overlap not allowed (*).*

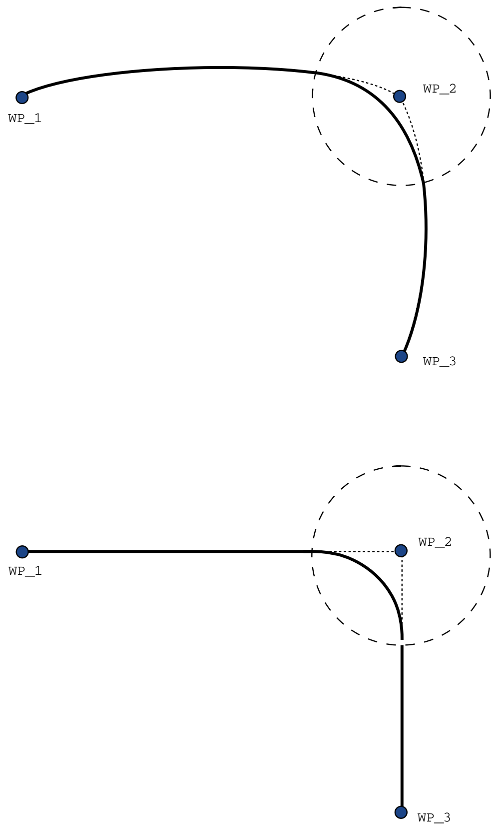
Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Blend Trajectories

Description

Blending enables the robot to smoothly transition between two trajectories, without stopping at the waypoint between them.
Therefore, your robot program will run faster.

Blend Trajectories



14.1: *Joint space (**MoveJ**) vs. cartesian space (**MoveL**) movement and blend.*

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Depending on the movement type (i.e., MoveL, MoveJ, or MoveP), different blend trajectories are generated.

- **Blends in MoveP** When blending in MoveP, the position of the blend follows a circle arc at constant speed. The orientation blends with a smooth interpolation between the two trajectories. You can blend a MoveJ or a MoveL into a MoveP. In such a case, the robot uses the circular arc blend of MoveP, and interpolate the speed of the two motions. You cannot blend a MoveP to a MoveJ or a MoveL. Instead, the last waypoint of the MoveP is regarded as a stop point with no blend. You cannot perform a blend if the two trajectories are at an angle close to 180 degrees (reverse direction) because it creates a circular arc with a very small radius which the robot cannot follow at constant speed. This causes a runtime exception in the program which can be corrected by adjusting the waypoints to cause a less sharp angle.
- **Blends involving MoveJ** MoveJ blends cause a smooth curve in joint space. This goes for blends from MoveJ to MoveJ, MoveJ to MoveL and MoveL to MoveJ. Blending is not performed if *time* is specified instead of *velocity* and *acceleration* for both motions.
- **Blends in MoveL** MoveL blends cause a smooth curve in Cartesian space. The orientation blends with a smooth interpolation between the two trajectories. The robot may decelerate on the trajectory before the blend to avoid very high accelerations (e.g., if the angle between the two trajectories are close to 180 degrees).

Conditional Blend Trajectories

Description

Blending enables the robot to smoothly transition between two trajectories, without stopping at the waypoint between them.
Blending makes your robot program run faster because, when you create a smooth transition between trajectories, you avoid slowing down an acceleration between trajectories.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

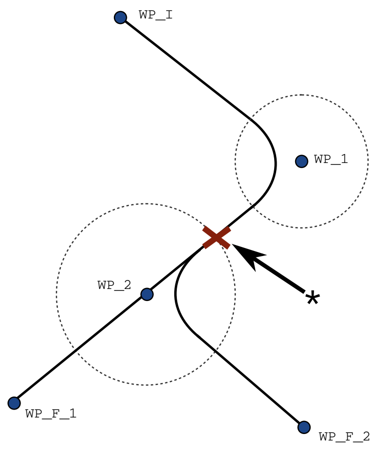
Conditional blend trajectories

This example is a very situational example, but it shows that the robot program may in a very rare situations calculate program nodes before execution.

The blend trajectory is affected both by the waypoint where the blend radius is set and the following one in the program tree.

In this example, the blend around is affected by `(WP_1)` and `(WP_2)`. The consequence of this becomes more apparent when blending around `(WP_2)` in this example.
There are two possible ending positions and to determine which is the next waypoint to blend to, the robot must evaluate the current reading of the `digital_input[1]` already when entering the blend radius.
That means the `if...then` expression is evaluated before we actually reach the destination which is somewhat counter-intuitive when looking at the program sequence. If a waypoint is a stop point and followed by conditional expressions to determine the next waypoint (e.g. the I/O command) it is executed when the robot arm has stopped at the waypoint. `(WP_2)`

```
MoveL
    WP_I
    WP_1 (blend)
    WP_2 (blend)
    if (digital_input[1]) then
        WP_F_1
    else
        WP_F_2
```



14.2: *`WP_I` is the initial waypoint and there are two potential final waypoints `WP_F_1` and `WP_F_2`, depending on a conditional expression. The conditional `if` expression is evaluated when the robot arm enters the second blend (*).*

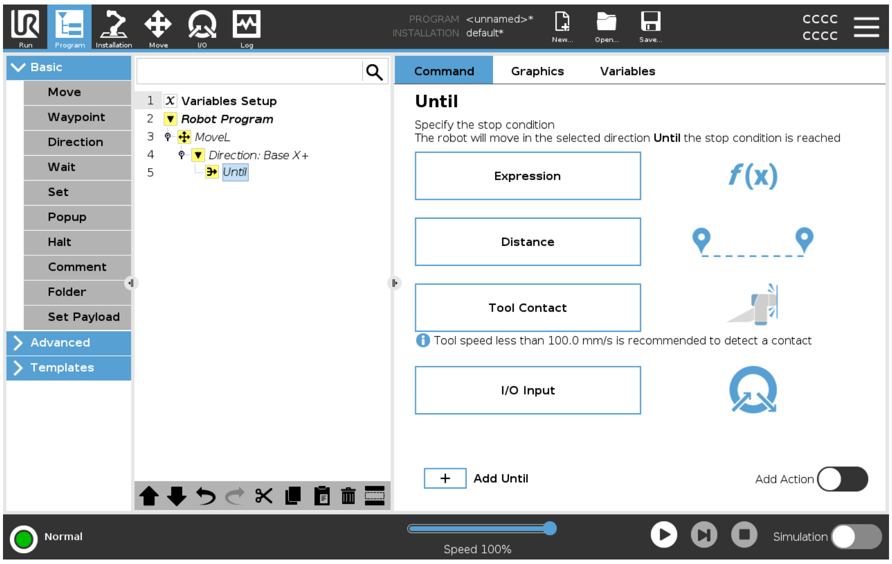
Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Add Until

Description

The program node **Until** defines stop criteria for a motion.
You can add Until nodes from Direction and Waypoint commands.
You can only add a direction node to a MoveL and MoveP.
The robot moves along a path and stops when contact is detected.

You can add multiple stop criteria to a single movement. The motion stops when the first **Until** condition is met.
You can add multiple Add Until one after the other, because there can be multiple conditions that must be met, before an action is done or executed.



- Distance
- Tool Contact: (see 4.7.3 Direction on page 117)
- Expression
- I/O Input:
- Add Action

Expression

This node uses a custom program expression to stop the robot's motion. You can also use an expression to specify a stop condition.
Stop conditions can also be specified using variables and script functions.

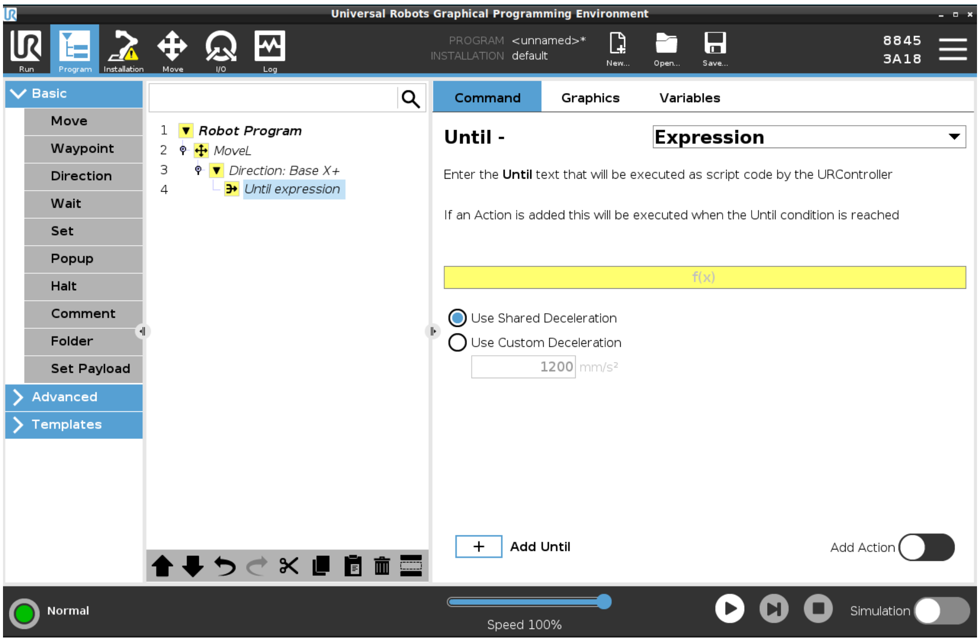
Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Add an Until expression to a robot program

1. When you have added the Until command, tap the

   Expression

2. Tap the yellow expression field
3. Add the expression with the keyboard
4. Tap Submit  to save the expression
5. Choose if you want to use the shared deceleration or a custom deceleration



Distance

This node stops a Direction move when the robot moves a certain distance. The velocity is ramped down so the robot stops exactly at the specific distance.
You can also use a specific distance as a stop condition.

Example

You can use the distance function to move the tool a specific distance before a full stop such as moving the tool away from a work piece.

Add an Until distance to a robot program

1. When you have added the Until command, tap the

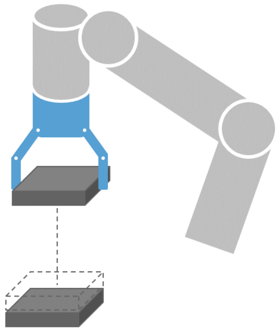
   Distance

2. Add the distance in mm
3. Select to stop after it has moved the distance or blend with a radius. Click here to read more about blending.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Tool Contact

This node allows the robot to stop motion when contact with the tool is established.
You can use this node to stop a movement when the robot tool detects a contact.



You can also define the deceleration of the stop and the retraction of the tool.

Example

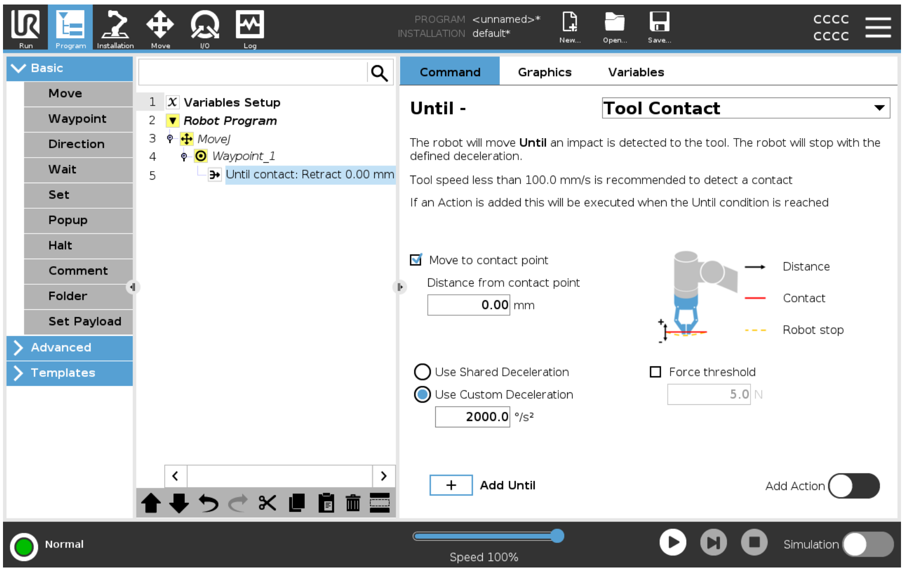
You can use the Until Tool Contact Node for applications like Stacking/Destacking, where Until Tool Contact determines the height of stacked objects.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Add a Until tool contact to a robot program

1. When you have added the Until command, tap the

Tool Contact



CAUTION

The default speed of motion is too high for contact detection. A faster speed of motion triggers a robot stop, before the Tool Contact condition can take effect. To avoid triggering a robot stop, lower the speed of motion. For example: 100m/s.

NOTICE

Until Tool Contact might not work if the mounted tool vibrates. For example: a vaccuum gripper with an embedded pump can introduce fast vibrations.

Detail

Use the **Retract to Contact** setting for the robot to return to the initial point of contact. You can set an additional reverse movement to make the robot move free of, or toward, contact. This is useful if you have a gripper that needs free space to move, or if a clamping action is needed.

I/O Input

This node uses an I/O input to stop a signal controlled motion.
You can also use an I/O input to specify a stop condition.

Copyright © 2009–2025 by Universal Robots A/S. All rights reserved.

add a No Action to a robot program

1. When you have added the Until command, tap the

   I/O Input

2. Select the analogue or digital input
3. Add additional configuration

Add Action

This node allows you to add a program node if a specific Until condition is met. You can also add an additional action to be executed right after an Until command.

Example

Until Tool Contact can engage the gripping action of a gripper tool. If no **Action** is defined, then program execution continues to the next program node in the Program Tree.

Add a No Action after an Until command

1. Tap the Add Action
2. Select a node from the Node List to be added immediately after the Until command
3. Configure the node you just added

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

4.7.3. Direction

Description	The Direction command specifies a motion relative to feature axes or TCPs. The robot moves in the path specified by the Direction program node until that movement is stopped by an Add Until on page 112 condition.

Direction The Direction command allows you to make the robot move in a specific direction.

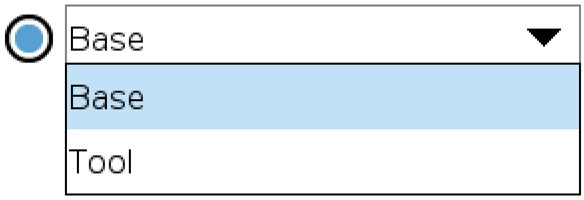
Example Direction vectors of [100,0,0] and [1,0,0] have the same effect on the robot; use the Speed Slider to moving along the x-axis at a desired speed. The values of the numbers in the direction vector only matter relative to each other.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Add a Direction movement to a robot program

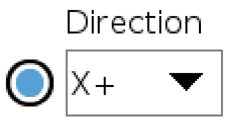
1. In your robot program, select the place or node where you wish to add a Direction command.
2. Under Basic, tap **Direction** to add a linear movement to your Program Tree.
3. In the Direction field, under Feature, define the linear movement.

   a. Select if you use the shared feature 

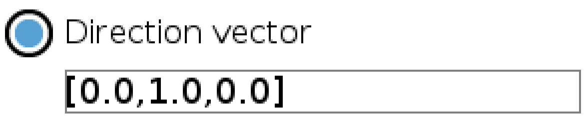
   

   or the Base/Tool

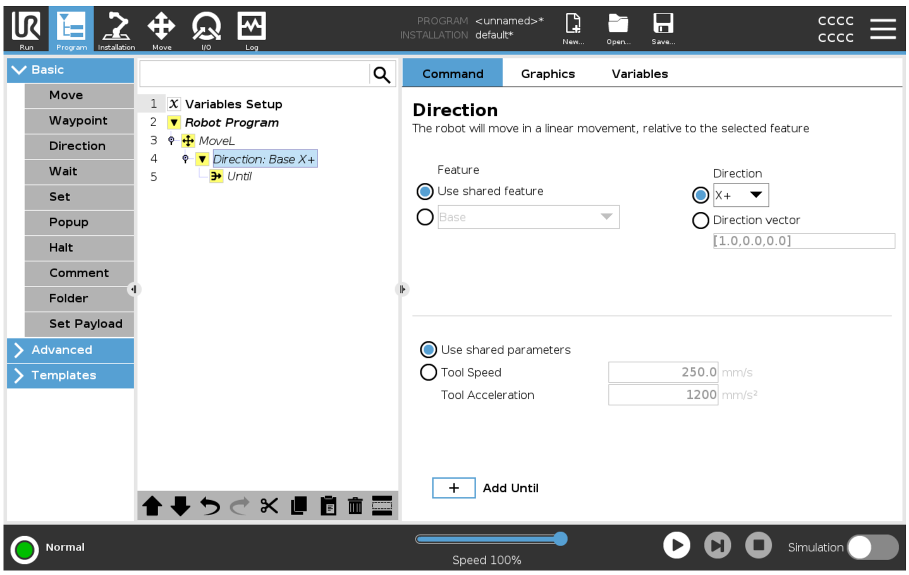
   b. Select the cartesian direction from the Direction drop-down

   or write the Direction vector manually

   c. Select Shared parameters or define the tool speed and acceleration.



4. Click Add Until to add a method of stopping the directional movement.

Select the feature and direction

Difference between Shared Feature or BASE/TOOL

Difference between Direction and writing the Direction Vector

The Direction Vectors define a custom code expression that is resolved to a unit vector.

Direction Movement

There are different ways to stop direction movement.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

In the Direction field, tap the **Add Until** button to define and add stop criteria to your Program Tree.

Direction Vector

The Direction Vector settings allow you to define the vector direction for linear motion. Use the **Tool Speed** and **Tool Acceleration** for the following:

- to define linear motion relative to multiple feature axes
- to calculate the direction as a mathematical expression

4.7.4. Wait

Description

The Wait command provides additional control of the robot's behavior. The Wait command pauses the robot's movement when new inputs are introduced into the program.

You can add a Wait command to a program with external sensors, to make the robot wait for one of the sensors to activate before the program continues.

When you add a Wait to your program tree, the Wait pane appears to the right of the screen.

See the following sections about using Wait:

- Add a No Wait to a robot program below
- Add Wait x seconds to a robot program on the next page
- Wait for digital input on page 121
- 4.7.4 Wait above
- Wait for f(x) expression on page 123

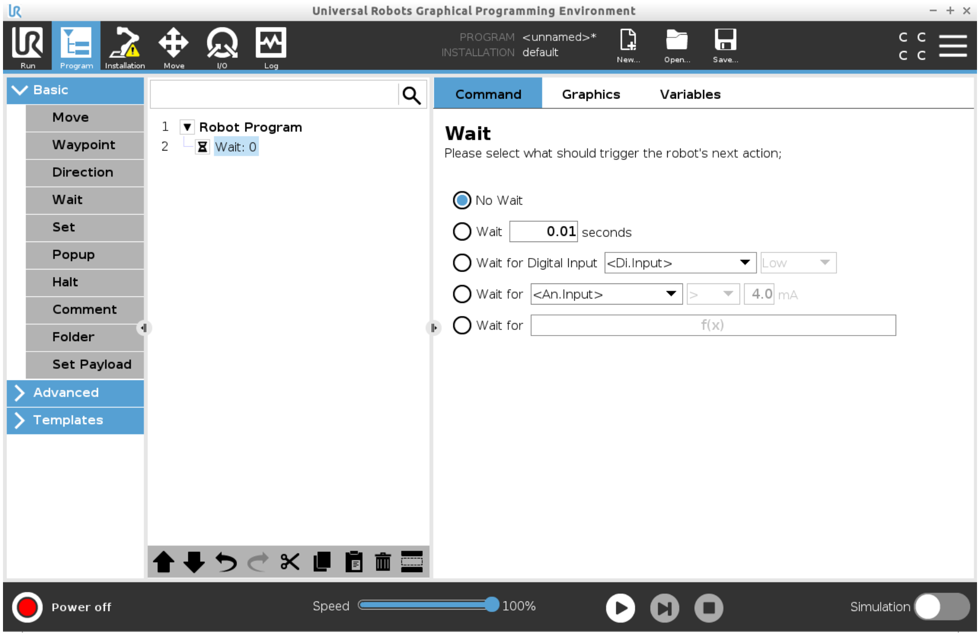
The robot pauses using different Wait commands.
See the Wait command types below.

Add a No Wait to a robot program

No Wait allows the robot to do nothing before continuing to run a program. There is no pause before new input.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

1. In your Robot Program, select the place or node where you wish to add a Wait command.
2. Under Basic, tap **Wait**.
3. On the right side of the screen, tap the **No wait** selector.



The No Wait command can be used a temporary placeholder be used, when you are programming your robot program.

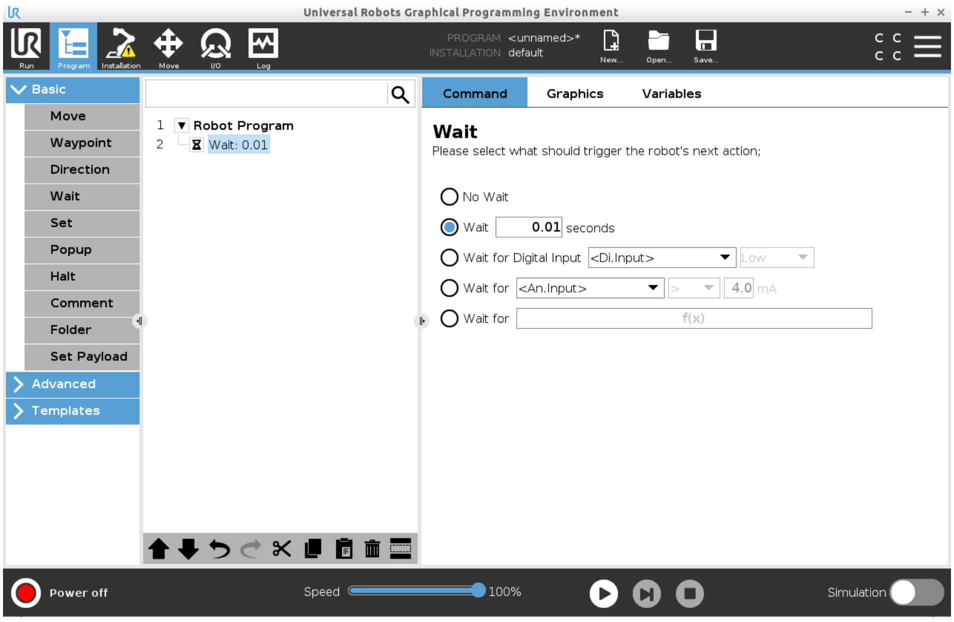
Add Wait x seconds to a robot program

This Wait command type allows the robot to hold position for a defined amount of time before continuing a program.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

1. In your Robot Program, select the place or node where you wish to add a Wait command.
2. Under Basic, tap **Wait**.
3. Tap the **Wait x seconds** selector.

4. Tap the number box to select a value that determines the length of the Wait.



Example: Wait x seconds

1. If you have an object that needs to cool down to a specific temperature, you can add this time delay into the robot program.
2. If you need an external operation to finish before continuing with the robot program.

Wait for digital input

This Wait command type allows the robot to hold position until a signal is received from a digital input.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

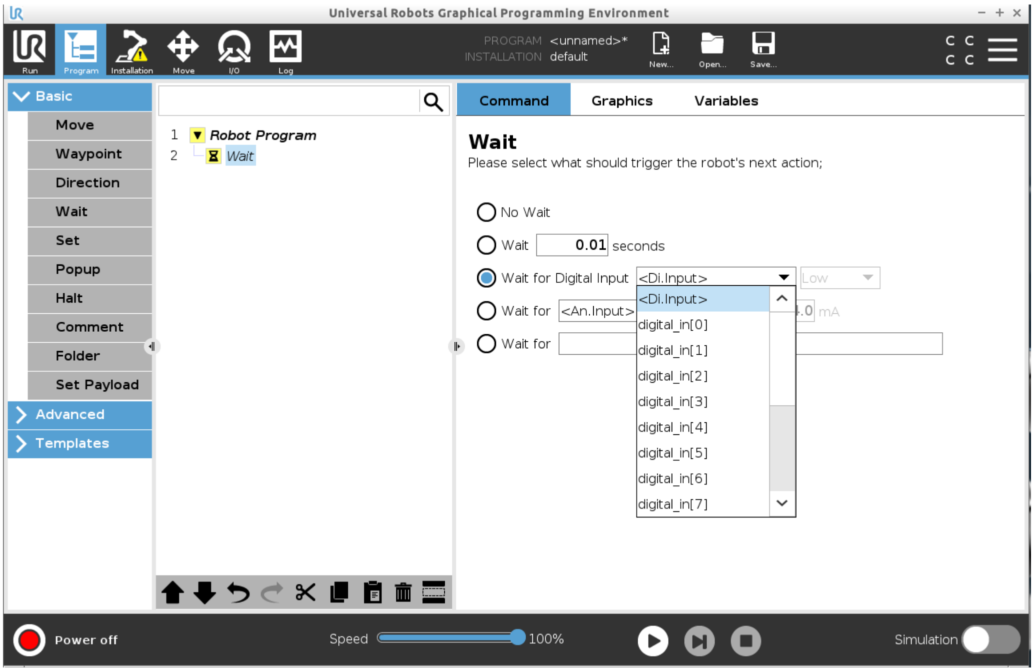
Add a Wait for digital input to a robot program

1. In your Robot Program, select the place or node where you wish to add a Wait command.
2. Under Basic, tap **Wait**.
3. Tap the **Wait for Digital Input** selector.

More definition is required for this Wait type.

4. In the **Di.Input** dropdown menu, select a new digital input.
5. In the signal box, assign a Low or High signal type to the new digital input.

If you have multiple digital inputs, you can rename each one to make it easier to find them.



Example: Wait for digital input

If you have an external sensor at the end of a conveyor, you will use this function to wait for a command from the sensor, which tells the robot program that there is a work piece at the end of the conveyor.

Wait for analog input

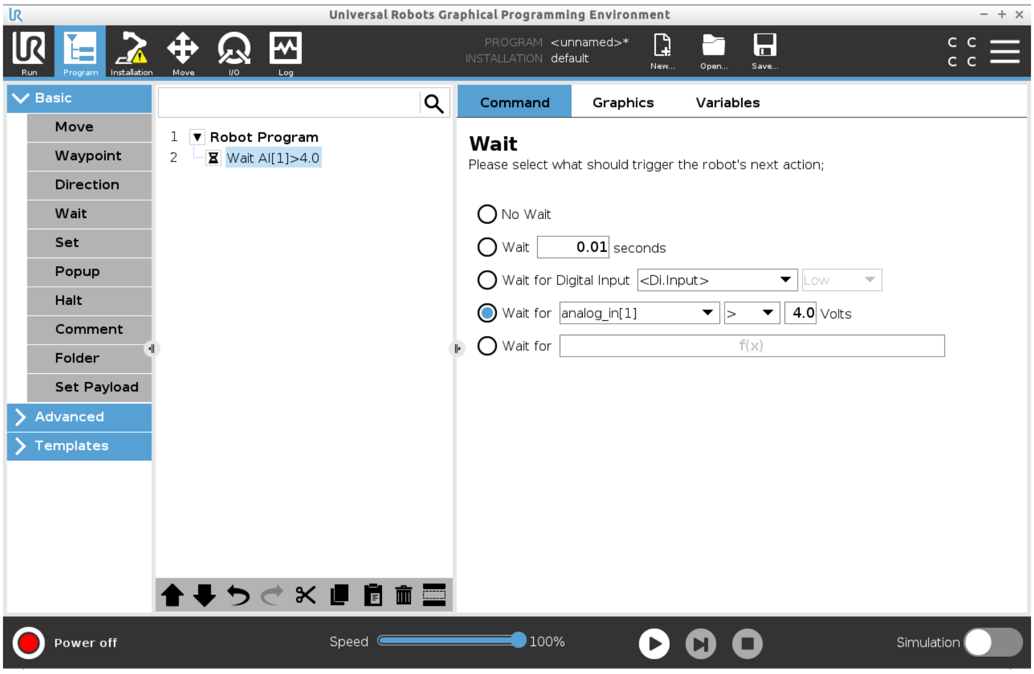
This Wait command type allows the robot to hold position until a signal is received from an analog input.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Add a Wait for analog input to a robot program

1. In your Robot Program, select the place or node where you wish to add a Wait command.
2. Under Basic, tap **Wait**.
3. Tap the **Wait for analog input** selector.

4. In the **An.input** menu, select the new analog input.
5. Tap the Volts box to select a value that determines the voltage.
6. In the angle brackets box, select either the less than **<**, or the greater than **>** symbol.



Tip

If you have multiple analog inputs, you can rename each one to make it easier to find them. [Go to I/O setup]

Example: Wait for analog input

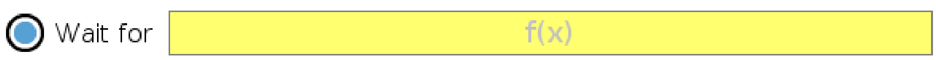
If you have a temperature sensor that is set up to the analogue I/O, you can configure the robot program to wait until a specific temperature.

Wait for f(x) expression

This Wait command type allows the robot to hold position until an expression is *True* or *[1]*

Add a Wait for f(x) variable to a robot program

1. In your Robot Program, select the place or node where you wish to add a Wait command.
2. Under Basic, tap **Wait**.
3. Tap the **Wait for f(x)** selector.



4. Tap the **f(x)** variable field to add an expression value.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Example: Wait for f (x) expression You can add a wait command with an expression that waits for two or more conditions to be true or false depending on configuration when using a machine and a conveyor sensor. Both the machine and the conveyor become ready for the robot.

4.7.5. Set

Description The Set command is one of the most often used commands. The Set command can turn external devices on and off. The Set command can also increase or decrease current/voltage to external devices.
You can always test your setup by using the test button in the right lower corner of the screen.
See the following sections about using Set:

- No Action below
- Set digital output on the facing page
- Set analog output on the facing page
- Set (variable) on page 126
- Set single pulse on page 126
- Increment installation variable by one on page 126
- Set TCP on page 126

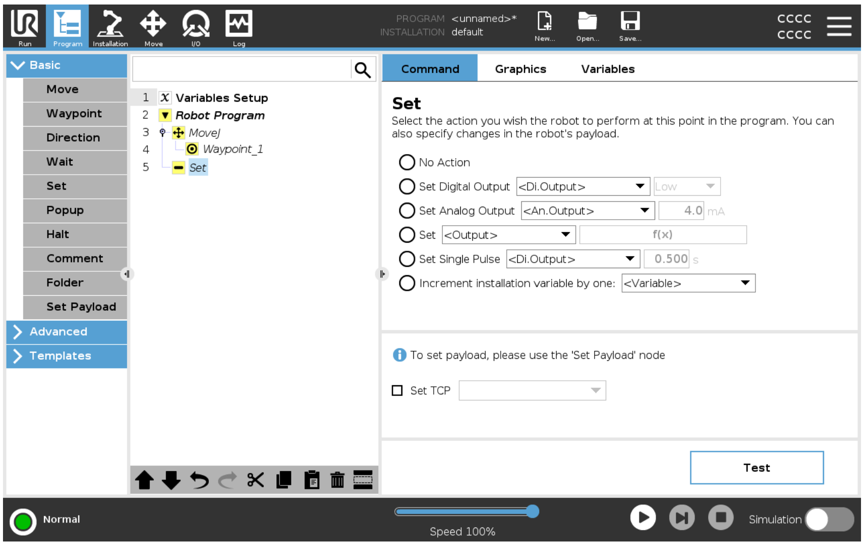
The robot uses different Set commands.
See the Set command types below.

No Action This is used in combination with setting the active TCP.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

To add a No Action to a robot program

1. In your Robot Program, select the place or node where you wish to add a Set command.
2. Under Basic, tap **Set**.
3. Select No Action



Set digital output

The digital output should be planned as a start/stop or on/off action.

1. In your Robot Program, select the place or node where you wish to add a Set command.
2. Under Basic, tap **Set**.
3. Select Set Digital Output
4. Select your pre-named digital output (See 5.4 I/O Setup on page 180)
5. Set to High/Low

Example: Set digital output

Use this Set command, if you want a conveyor to start or stop moving.

Set analog output

The analogue output should be planned as a variable increase/decrease in current/voltage.

1. In your Robot Program, select the place or node where you wish to add a Set command.
2. Under Basic, tap **Set**.
3. Select Analog Output
4. Select your pre-named analogue output (See 5.4 I/O Setup on page 180)
5. Input desired value (current or voltage depending on configuration in 7 I/O Tab on page 218

Example: Set analog output

Use this command, if you wish to increase the speed of a conveyor or dim the light in a lamp or diode.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Set (variable)

The output can also be modified by an expression.

1. In your Robot Program, select the place or node where you wish to add a Set command.
2. Under Basic, tap **Set**.
3. Select Set
4. Choose the pre-named output
5. Add the expression in the input field.

Example: Set (variable)

You can set an output to list the output torque of a robot joint.

Set single pulse

The Set command can be used to deliver a steady pulse of a specific duration. The output will remain High during the pulse, and return to Low after the pulse has ended.

1. In your Robot Program, select the place or node where you wish to add a Set command.
2. Under Basic, tap **Set**.
3. Select Set Single Pulse
4. Select your pre-named digital output
5. Add your duration for the pulse in (s)

Example: Set single pulse

In order to ensure valid communication with older machinery, you can set a pulse to a high command for a duration, so that you ensure that the older machinery has time to register the command.

Increment installation variable by one

This is used to increase the number of a counter variable.

1. In your Robot Program, select the place or node where you wish to add a Set command.
2. Under Basic, tap **Set**.
3. Select Increment installation variable by one
4. Select your variable from the dropdown menu.

Example

If you need to know how many items that the robot has handled, you can add a counter and this Set command to increase the counter.

Set TCP

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

1. In your Robot Program, select the place or node where you wish to add a Set command.
2. Under Basic, tap **Set**
   a. You can set the active TCP with any command variation from the above.
3. Tap Set TCP
4. Select TCP from dropdown menu

Example: Set TCP

If you have a double gripper, you can use the Set command to change the active TCP to the other gripper.

4.7.6. Popup

Description

The Popup is a message that appears on the screen when the program reaches the Popup node in the program tree.
Popup messages are limited to a maximum of 255 characters. You can choose to use different popup message types.

- Message
- Warning
- Error

A popup always pauses the main robot program. The main robot program is paused while the popup is active.

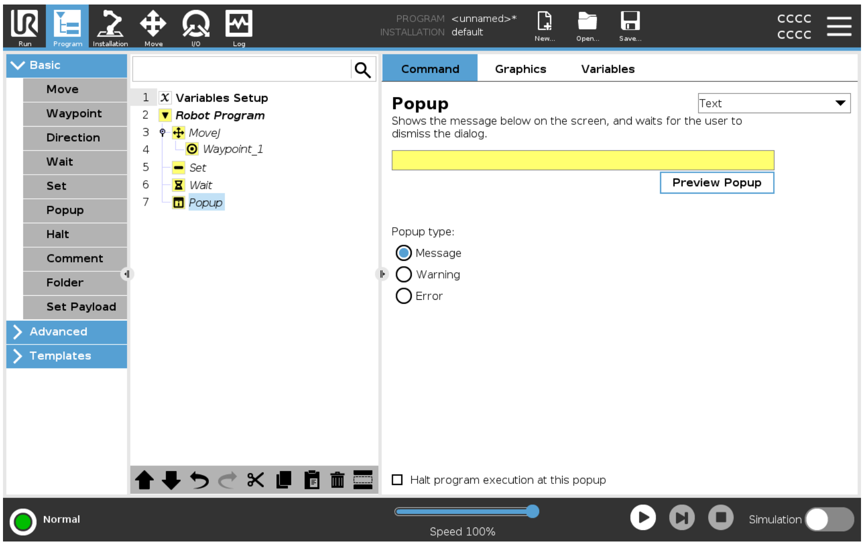
When to use:

- When you wish to continue or stop your program.
- When you wish to create a message to inform users.
- When you wish to control your part of a robot program.

Add a Popup to a robot program

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

1. In your robot program, select the place or node where you wish to add a Popup command.
2. Under Basic, tap **Popup**.
3. Add your text or choose a variable
4. Choose the popup type. The only change is the icon of the popup.



Tip You can also select **Halt program execution at this popup** for the program to stop when the popup appears.

Example: Popup In a pick and place program, you can add a Message popup when the pallet is full and you need to add an empty pallet.
During an inspection, you can add a Warning popup where you inspect an object. If the inspection is OK, continue the program. If the inspection is NOT OK, stop the program.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

4.7.7. Halt

Description

The Halt command allows you to stop the robot at a specific node in the robot program. It is the equivalent of pressing the STOP button.

You must restart the program after the halt command.

When you add a Halt to your robot program, the Halt pane appears to the right of the screen.

When to use:

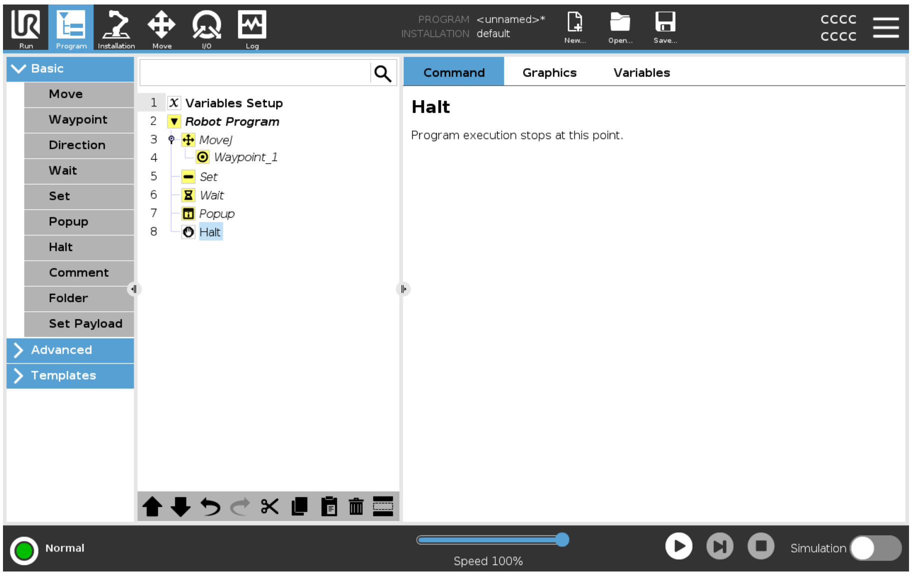
- Include a Halt command if you reach a position with the robot program, where recovery is not possible, and you need to stop the program.

Halt

Add stops to specific points in the robot program.

Add a Halt to a robot program

1. In your robot program, select the place or node where you wish to add a Halt command.
2. Under Basic, tap **Halt**.



Example: Halt

If there are no more parts for a welding application, or one of the parts have been incorrectly placed in the welding pattern, you can add a Halt command.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

4.7.8. Comment

Description

The Comment command allows you to keep track of decisions you make while you create or update your robot program.

You can add comments directly inside a robot program used by different users.

When you add a Comment to your robot program, the Comment pane appears to the right of the screen. The content of the comment displays in the Comment pane.

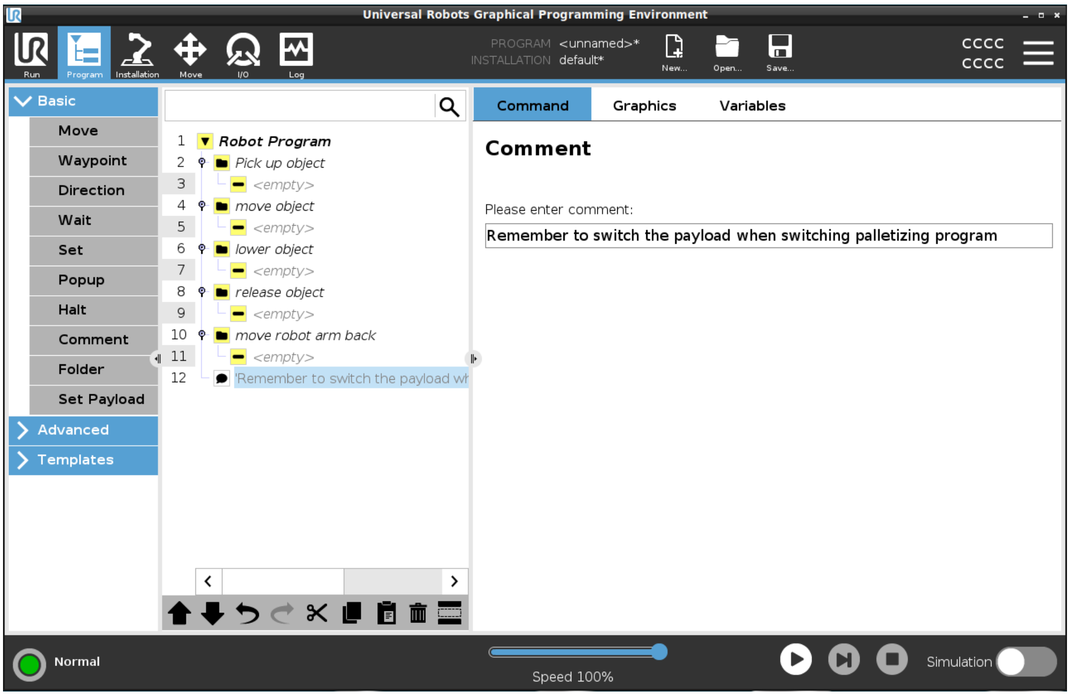
Comments have no impact on the program's execution.

Comment

Add useful comments throughout the robot program.

Add a Comment to a robot program

1. In your robot program, select a node to add a comment.
2. Under Basic, tap Comment.
3. Add your comment in the text field.



Example: Comment

The comments are mainly used by programmers to provide insight and help to other programmers on the same robot program.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

4.7.9. Folder

Description

The Folder command allows you to use folders to create an easy to read overview of the main parts of your robot program.

Since each main part of the robot program can contain many program nodes, you can use folders to separate them into manageable sections.

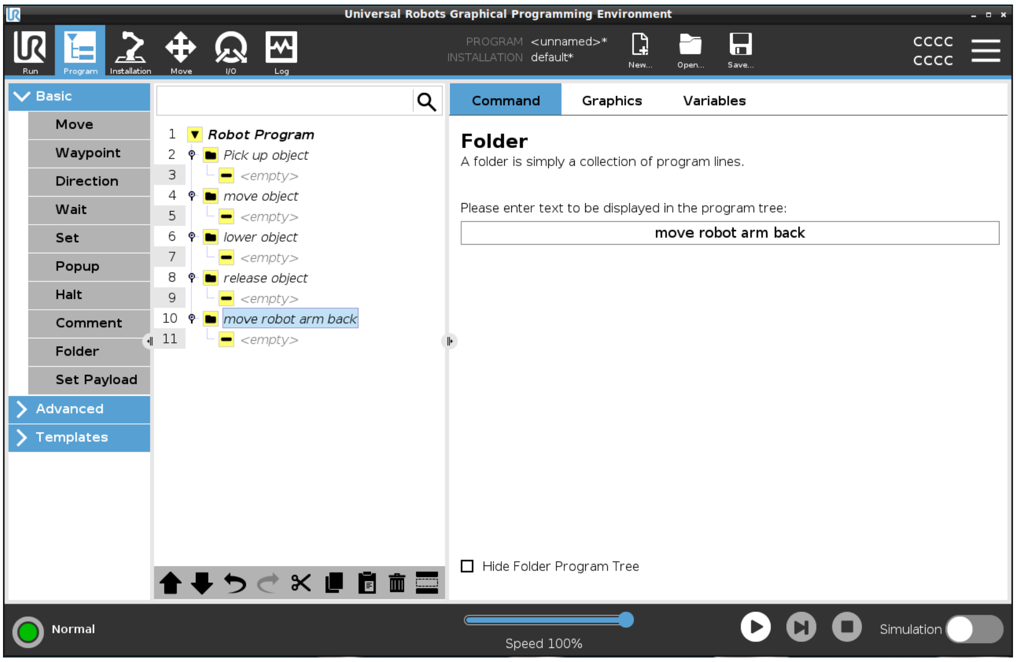
Folders have no impact on the program's execution.

Folder

Organize your robot program in folders.

Add a Folder to a robot program

1. In your robot program, select a placement for your folder.
2. Under Basic, tap Folder.
3. You can now move nodes into the folder, or add nodes in the folder.



Example: Folder

One of the main uses of Folders is to collapse main sections of the robot program to provide a better overview of the robot program.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

4.7.10. Set Payload

Description

The Set Payload command allows you to configure the payload for the robot. Payload is the combined weight of everything attached to the robot tool flange.
When to use:

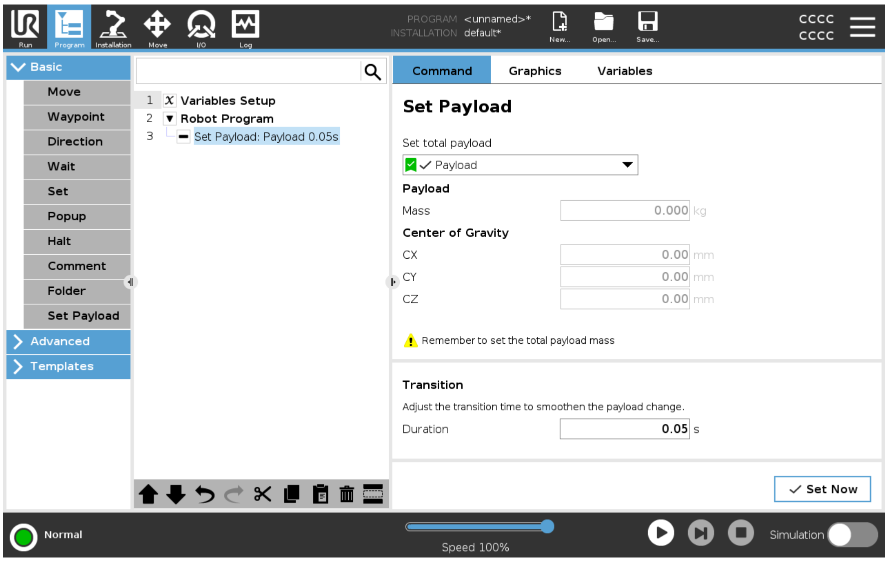
- When adjusting the payload weight to prevent the robot from triggering a robot stop. A correctly configured payload weight ensures optimal robot movement.

  Setting the payload correctly ensures optimal motion performance and avoids robot stops.
- When setting up the payload for use in a pick and place program, using a gripper.

Set Payload

Use the Set Payload command

1. In your robot program, select the place or node where you wish to add a Set command.
2. Under Basic, tap **Set Payload**.
3. Use the drop-down, under **Select Payload**.
   a. Select one of the payloads already configured.
   b. Or, use the drop-down to configure a new payload by selecting **Custom Payload** and completing the mass and CoG fields.



Tip

You can also use the **Set Now** button to set the values on the node as the active payload.

Use tip

Remember to always update your payload when making any changes to the configuration of the robot program.

Example: Set Payload

In a pick and place program, you would create a default payload in the installation. Then you add a Set Payload when picking up an object. You would update the payload after the gripper closes, but before starting to move.
Additionally, you would use the Set Payload after the object has been released.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Payload Transition Time

This is the time it takes the robot to adjust for a given payload. At the bottom of the screen, you can set the transition time between different payloads.
You can add a payload transition time in seconds.
Setting a transition time larger than zero, prevents the robot from doing a small "jump", when the payload changes. The program continues while the adjustment is taking place.
Using the Payload Transition Time is recommended when
picking up or releasing heavy objects or using a vacuum gripper.

4.8. Advanced program nodes

Description

The advanced program nodes are used to add additional functionality for your robot program, such as; subprograms, if-parameters, scripts and loops.

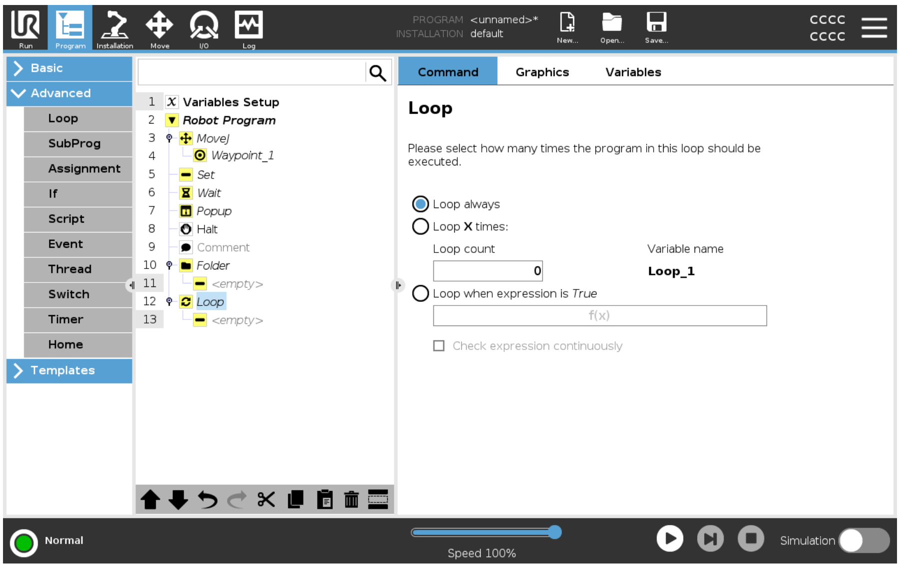
4.8.1. Loop

Description

Loops the underlying program commands. Depending on the selection, the underlying program commands are either looped infinitely, a certain number of times or as long as the given condition is true. When looping a certain number of times, a dedicated loop variable (called `loop_1` in the screen shot above) is created, which can be used in expressions within the loop. The loop variable counts from 0 to $N-1$.

Looping an Expression

When looping using an expression as end condition, PolyScope provides an option for continuously evaluating that expression, so that the “loop” can be interrupted anytime during its execution, rather than just after each iteration.

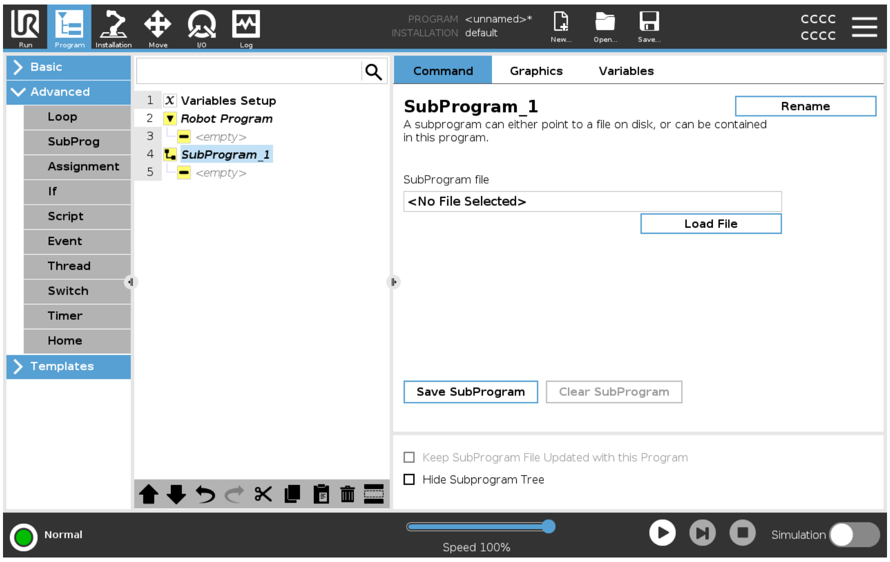


Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

4.8.2. SubProgram

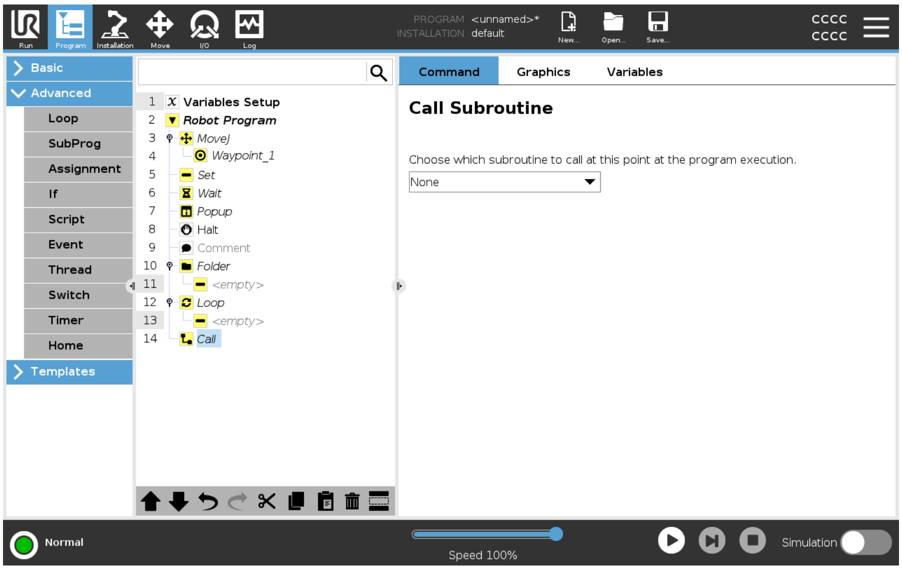
Description

A SubProgram can hold program parts that are needed several places. A SubProgram can be a separate file on the disk, and can also be hidden to protect against accidental changes to the SubProgram.



Call Subroutine

A call to a Subroutine will run the program lines in the SubProgram, and then return to the following line.

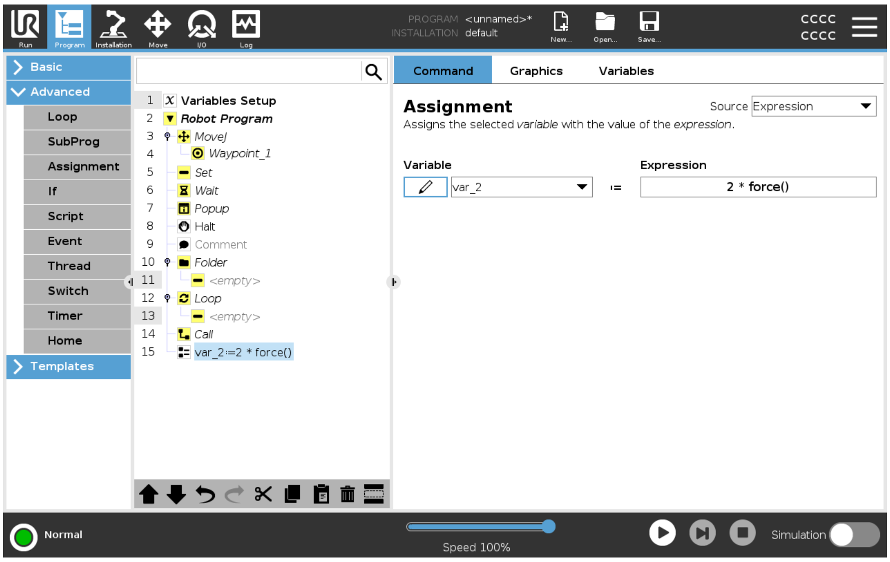


Copyright © 2009–2025 by Universal Robots A/S. All rights reserved.

4.8.3. Assignment

Description

Assigns values to the variables. The variable value can be the result of expressions created in the Expression Editor (see section 4.3 Expression Editor on page 79).
You can also request a variable value from an operator. When requesting a value from an operator, it is possible to display an Operator Message to validate input against common variable types.



4.8.4. If

Description

If and **If...Else** statements change the robot's behavior based on sensor inputs or variable values.

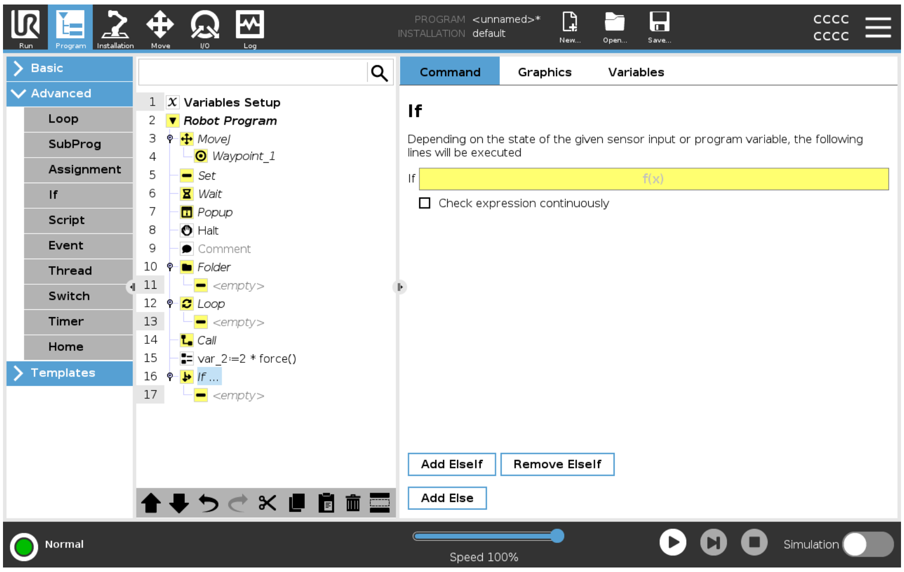
> **NOTICE**
>
> If there are waypoints inside an If expression or inside a Loop expression with the Check Expression Continuously option, you can add a stopj() or a stopl() after the expression to gently decelerate the robot arm. This is valid for both If and Loop Commands (see section).

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Select Conditions

Select conditions in the Expression Editor that make up expressions using an **If** statement. If a condition is evaluated as True, the statements within this **If** command are executed. An **If** statement can have only one **Else** statement. Use **Add ElseIf** and **Remove ElseIf** to add and remove ElseIf expressions.

Select **Check Expression Continuously** to allow **If**, **ElseIf** and **Loop** statements to be evaluated while the contained lines are executed. If an expression inside an **If** statement is evaluated as False, the **ElseIf** or **Else** statements are followed.



Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

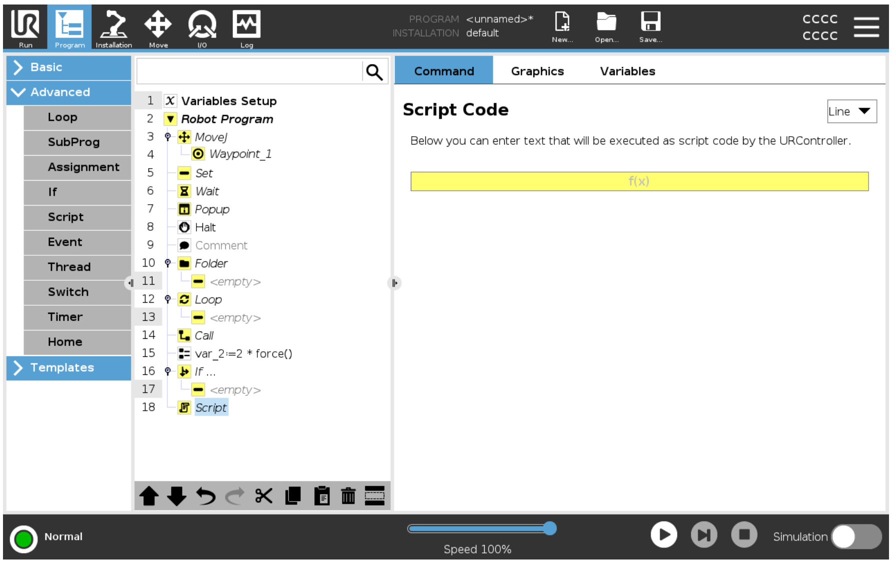
4.8.5. Script

Description

The following options are available in the drop down list under Command:

- **Line** allows you to write a single line of URscript code, using the Expression Editor (4.3 Expression Editor on page 79)
- **File** allows you to write, edit or load URscript files.

You can find instructions for writing URscript in the Script Manual on the support website (http://www.universal-robots.com/support).
Functions and variables declared in a URscript file are available for use througout the program in the PolyScope.

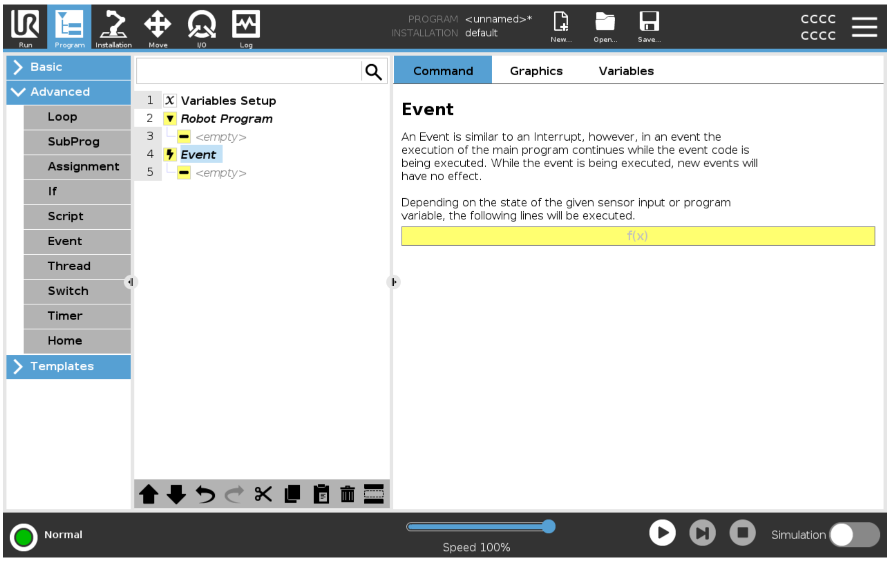


Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

4.8.6. Event

Description

An event can be used to monitor an input signal, and perform some action or set a variable when that input signal goes high.
For example, in the event that an output signal goes high, the event program can wait for 200ms and then set it back to low again.
This can make the main program code a lot simpler in the case on an external machine triggering on a rising flank rather than a high input level.
Events are checked once every control cycle (2ms) .

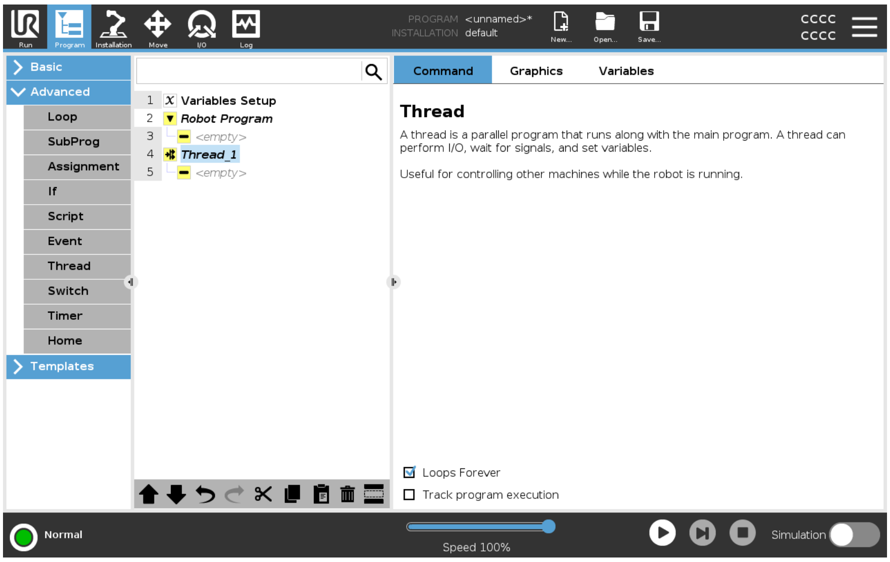


Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

4.8.7. Thread

Description

A thread is a parallel process to the robot program. A thread can be used to control an external machine independently of the robot arm. A thread can communicate with the robot program with variables and output signals.



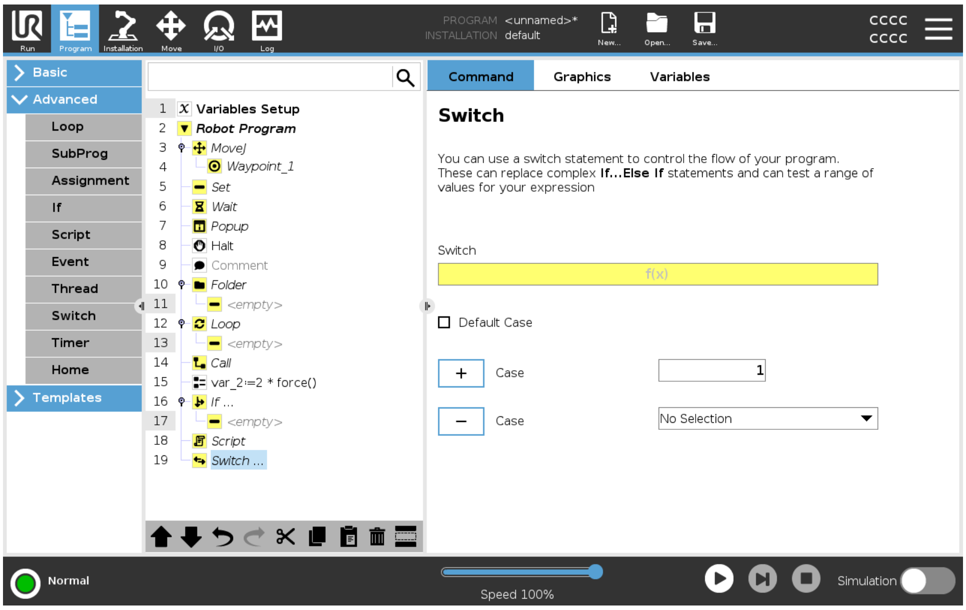
4.8.8. Switch

Description

A **Switch Case** construction can make the robot change behavior based on sensor inputs or variable values.
Use the **Expression Editor** to describe the base condition and define the cases under which the robot should proceed to the sub-commands of this `Switch`.
If the condition is evaluated to match one of the cases, the lines inside the `Case` are executed. If a `Default Case` has been specified, then the lines will be executed only if no other matching cases were found.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Each `Switch` can have several `Cases` and one `Default Case`. `Switches` can only have one instance of any `Case` values defined. `Cases` can be added using the buttons on the screen. A `Case` command can be removed from the screen for that switch.

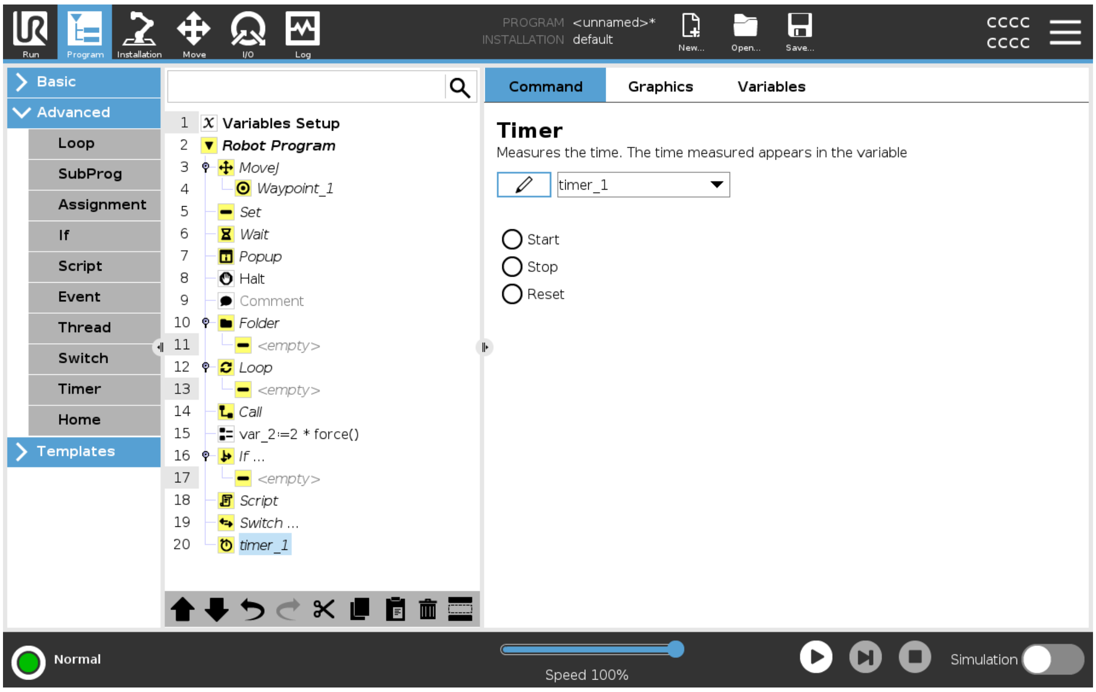


Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

4.8.9. Timer

Description

A Timer measures the length of time it takes for specific parts of the program to run. A program variable contains the time passed since a Timer started, and can be seen in the Variables Tab and in the Run Tab.

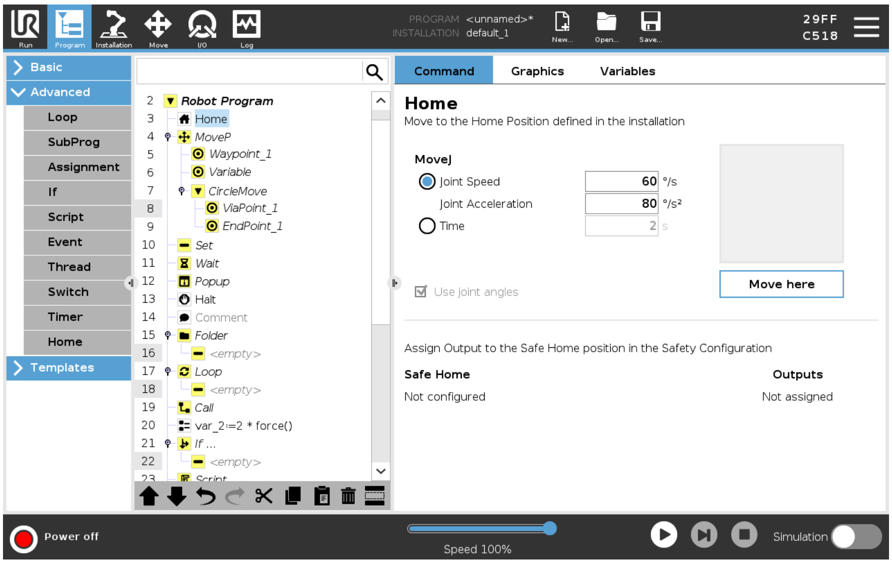


Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

4.8.10. Home

Description

The Home node uses joint angles to move the robot to a predefined Home position. If defined as a Safe Home position,the Home node displays as Home(Safety) in the Program Tree. If the Home position is out of sync with Safety, the node is undefined.



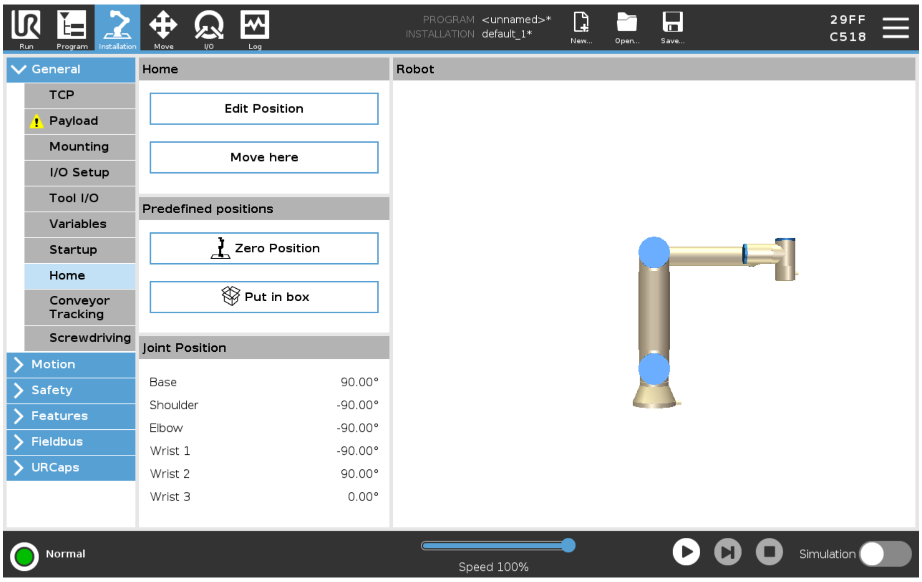
Defining Home

Home is a user-defined return position for the Robot Arm. Once defined, the Home Position is available when creating a robot program. You can use the Home Position to define a Safe Home Position. (See 2.9 Safe Home Position on page 59) Use the Home screen buttons for the following:

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

- **Edit Position** modifies a Home Position.
- **Move here** moves the Robot Arm to the defined Home Position.
- **Zero Position** returns the Robot Arm to an upright position.

1. In the Header, tap **Installation**.
2. Under **General**, select **Home**.
3. Tap **Set Position**.
4. Teach robot using either **Freedrive** or **Transition buttons**.



Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

4.9. Templates

Description The templates can be used to add specific functionality to a robot program. The different templates will enable the you to perform complex tasks with your robot program.

4.9.1. Seek

Description The seek function uses a sensor to determine the correct position to grab or drop an item. This function allows for working on stacks of items of varying thickness, and determining the exact positions of the items are either unknown or too hard to program.
The sensor can be a push button switch, a pressure sensor or a capacitive sensor.

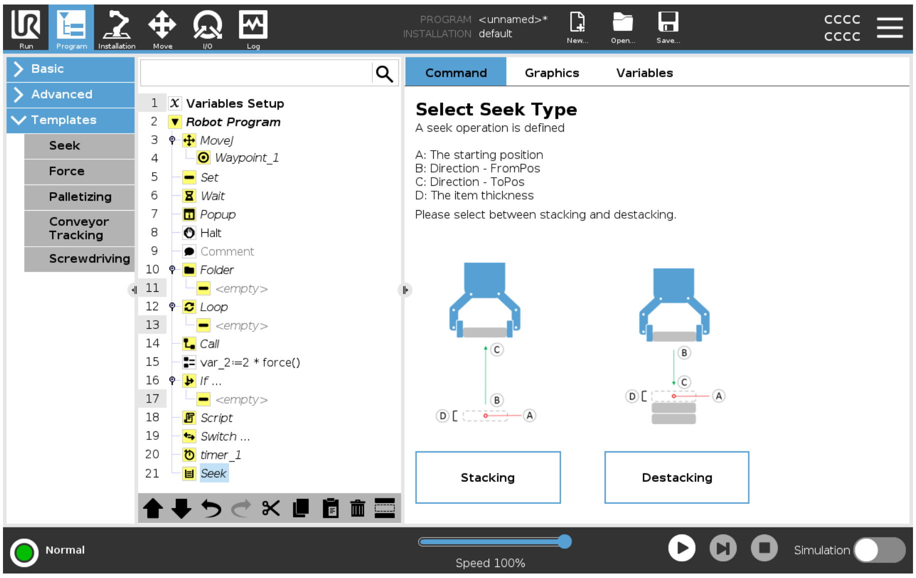
Seek To program a seek operation, define the following:

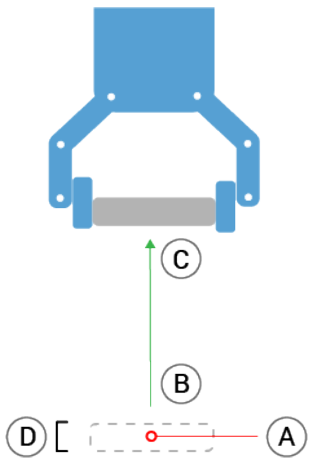
- *A* - the starting point.
- *B to C* - the stack direction. This means growing the stack when Stacking and shrinking the stack when Destacking.
- *D* - the thickness of the items in the stack.

You must also define the condition for when the next stack position is reached, and a special program sequence that is performed at each stack position.
Speed and accelerations need to be given for the movement involved in the stack operation.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Stacking During Stacking, the robot arm moves to point *A*, then moves *opposite* the direction to search for the next stack position. When the next stack position is found, the robot remembers it and performs the special sequence.
In subsequent rounds, the robot starts the search from the remembered position, incremented by the item's thickness along the direction.
Stacking is complete when the stack height is more than some defined number, or when a sensor gives a signal.

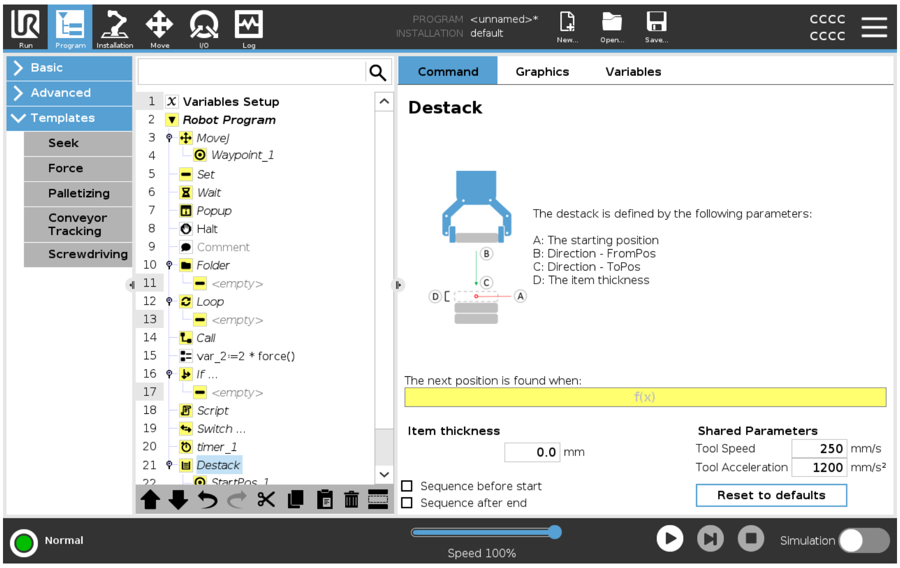


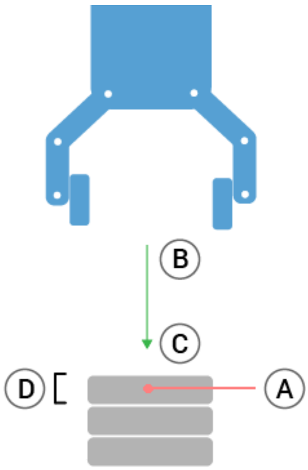


Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Destacking

During Destacking, the robot arm moves from point *A* in the given direction to search for the next item. The condition on the screen determines when the next item is reached. When the condition is satisfied, the robot remembers the position and performs the special sequence.
In subsequent rounds, the robot starts the search from the remembered position, incremented by the item's thickness along the direction.





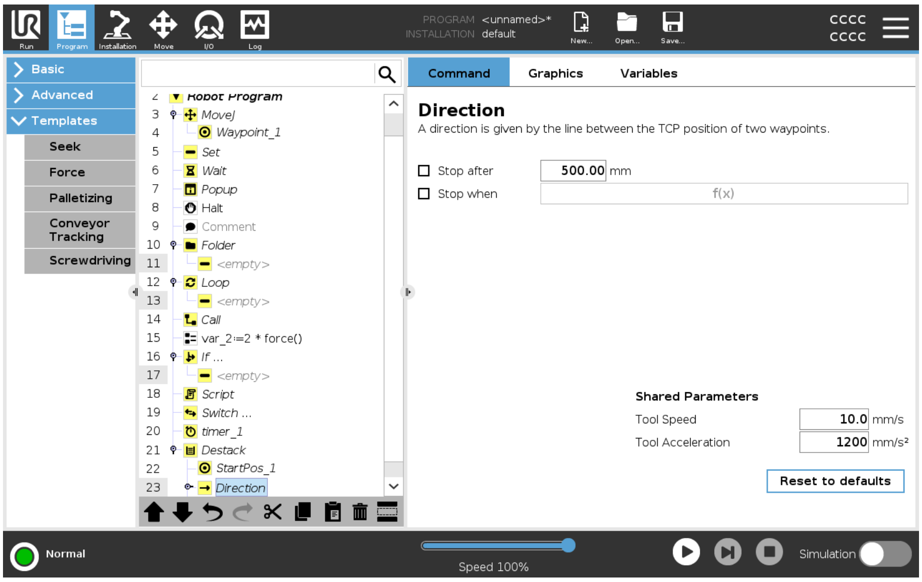
Starting position

The starting position is where the stack operation starts. If the starting position is omitted, the stack starts at the robot arm's current position.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Direction The direction, given by positions *B to C*, is calculated as the position difference from the TCP of *B* to the TCP of *C*.
Direction does not consider the orientations of the points.

Command	Action
Next Stacking Position Expression	The robot arm moves along the direction vector while continuously evaluating whether the next stack position has been reached. When the expression is evaluated to `True` the special sequence is executed.
BeforeStart	The optional `BeforeStart` sequence is run just before the operation starts. This can be used to wait for ready signals.
AfterEnd	The optional `AfterEnd` sequence is run when the operation is finished. This can be used to signal conveyor motion to start, preparing for the next stack.
Pick/Place Sequence	The Pick/Place Sequence is a special program sequence performed at each stack position, similar to the Pallet operation.



Copyright © 2009–2025 by Universal Robots A/S. All rights reserved.

4.9.2. Force

Description

Force mode is suited to applications where the actual TCP position along a predefined axis is not important, but instead a desired force along that axis is required. For example, if the robot TCP rolls against a curved surface, pushes or pulls a workpiece.

Force mode also supports applying certain torques around predefined axes. The robot arm attempts to accelerate along that axis, if no obstacles are met in an axis where a non-zero force is set. Although an axis is selected to be compliant, the robot program still tries to move the robot along that axis. However, force control assures that the robot arm still approaches the specified force.

NOTICE

Using this function at the same time as Conveyor Tracking and/or Path Offset can lead to program conflict.

- Do not use this function together with Conveyor Tracking or Path Offset.

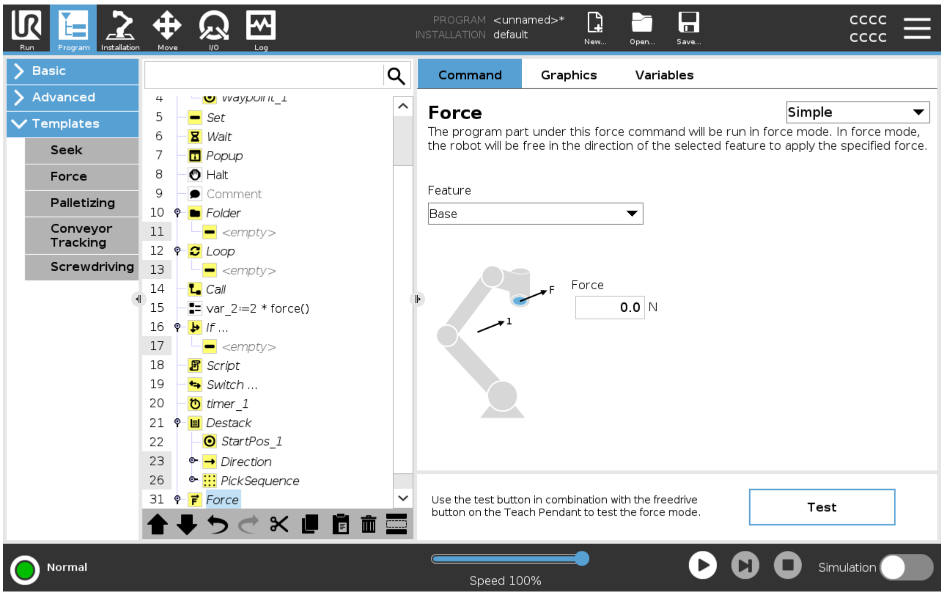
NOTICE

If there is a Force node inside an `If`, `ElseIf` or `Loop`, and the `Check Expression Continuously` option is selected, you can add an `end_force_mode()` script at the end of the expression to exit force control.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

WARNING

1. Avoid high deceleration just before entering force mode.
2. Avoid high acceleration in force mode, since it decreases force control accuracy.
3. Avoid movements parallel to compliant axes before entering force mode.



Feature selection The **Feature menu** is used to select the coordinate system (axes) the robot will use while it is operating in force mode. The features in the menu are those which have been defined in the installation.

Copyright © 2009–2025 by Universal Robots A/S. All rights reserved.

Force mode type

The types of force mode, listed below, determine how the selected feature is interpreted.

- **Simple**: Only one axis will be compliant in force mode. The force along this axis is adjustable. The desired force will always be applied along the z-axis of the selected feature. However, for Line features, it is along their y-axis.
- **Frame**: The Frame type allows for more advanced usage. Here, compliance and forces in all six degrees of freedom can be independently selected.
- **Point**: When Point is selected, the task frame has the y-axis pointing from the robot TCP towards the origin of the selected feature. The distance between the robot TCP and the origin of the selected feature is required to be at least 10 mm. The task frame changes at runtime as the position of the robot TCP changes. The x- and z-axis of the task frame are dependent on the original orientation of the selected feature.
- **Motion**: Motion means that the task frame will change with the direction of the TCP motion. The x-axis of the task frame will be the projection of the TCP movement direction onto the plane spanned by the x- and y-axis of the selected feature. The y-axis will be perpendicular to the robot arm's motion, and in the x-y plane of the selected feature. This can be useful when de-burring along a complex path, where a force is needed perpendicular to the TCP motion.

  When the robot arm is not moving: If force mode is entered with the robot arm standing still, there will be no compliant axes until the TCP speed is above zero. If later, while still in force mode, the robot arm is again standing still, the task frame has the same orientation as the last time the TCP speed was larger than zero.

For the last three types, the actual task frame can be viewed at runtime on the graphics tab (see), when the robot is operating in force mode.

Force value selection

- Force or torque value can be set for compliant axes, and robot arm adjusts its position to achieve the selected force.
- For non-compliant axes robot arm will follow the trajectory set by the program.

For translational parameters, the force is specified in Newtons [N] and for rotational the torque is specified in Newton meters [Nm].

NOTICE

You must do the following:

- Use `get_tcp_force()` script function in separate thread, to read actual force and torque.
- Correct wrench vector, if actual force and/or torque is lower than requested.

Speed limits

Maximum Cartesian speed can be set for compliant axes. The robot moves at this speed in force control, as long as it does not come into contact with an object.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Test force settings

The on/off button, labelled **Test**, toggles the behavior of the **Freedrive** button on the back of the Teach Pendant from normal Freedrive mode to testing the force command.
When the **Test button** is on and the **Freedrive** button on the back of the Teach Pendant is pressed, the robot will perform as if the program had reached this force command, and this way the settings can be verified before actually running the complete program. Especially, this possibility is useful for verifying that compliant axes and forces have been selected correctly. Simply hold the robot TCP using one hand and press the **Freedrive** button with the other, and notice in which directions the robot arm can/cannot be moved.
Upon leaving this screen, the Test button automatically switches off, which means the **Freedrive** button on the back of the Teach Pendant is again used for regular **Freedrive** mode.
The **Freedrive** button is only effectual when a valid feature is selected for the Force command.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

4.9.3. Palletizing

Description

Palletizing is a template to easily program palletizing and depalletizing tasks, picking-and-placing parts (i.e., from trays, fixtures, etc.), and having the robot perform repeatable actions for different items in multiple layers with different patterns.
You can create different patterns and apply them to specific layers. You can also place a separator between each layer (see).
Furthermore, you can use Features from Pallet Properties to easily adjust the placement of your pallet.
To learn about Features, see
Follow the **Creating a Palletizing Program** section below to use the Palletizing template.

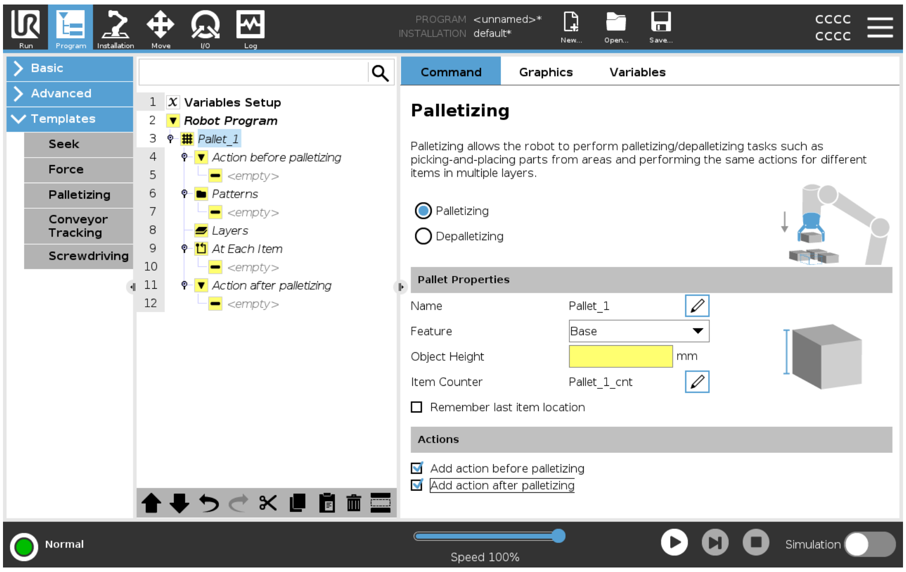
Creating a Palletizing Program

1. Decide if you want to teach a Feature (see) or use a Base as a reference plane.
2. In the **Program Tab**, under **Templates**, tap **Palletizing**.
3. On the Palletizing screen, select one of the following actions depending on the desired action.
   1. Select **Palletizing** to organize items onto a pallet.
   2. Select **Depalletizing** to remove items from a pallet.
4. Under **Pallet Properties**, specify the Name, Feature (see Step 1), Object Height, and Item Counter name for your program. Select the **Remember last item location** box if you want the robot to restart at the item it was handling when it stopped.
5. On the Palletizing screen, under **Actions**, add additional actions to be performed before or after palletizing sequence by selecting the following:
   1. **Add Action Before Palletizing**: These actions are performed before starting to palletize.
   2. **Add Action After Palletizing**: These actions are performed after finishing palletizing.
6. On the Program Tree, tap the **Patterns** node to designate patterns for your layers. You can create the following type of patterns: Line, Grid, or Irregular (see figure below). On this screen, you can select if you want to include a separator between layers (see).
7. Tap the pattern node(s) on the Program Tree to teach the robot layer-specific positions (e.g., start/end points, grid corners, and/or number of items). See for teaching instructions. All positions must be taught at the bottom of the pallet. To duplicate a pattern, tap the **Duplicate pattern** button on the Pattern node screen that you wish to duplicate.
8. In the Program Tree, tap the **Layers** node to configure the layers of your palletizing sequence. Use the **Choose Pattern** drop-down menu to select the pattern for each layer. Tap the **Add layer** button to add additional layers to your program. Layers must be added in the correct order, as they cannot be reordered later.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Creating a Palletizing Program

9. In the Program Tree, tap **At Each Item** node. Choose to use the default option (A) At Each Item Wizard, or (B) Manually Configure At Each Item. Instructions for each option are below.



At Each Item Wizard

The At Each Item Wizard assists in defining the actions performed at each item on a pallet, such as the ReferencePoint, the Approach Waypoint, ToolActionPoint Waypoint, and Exit Waypoint (described in the table below). The Approach and Exit Waypoints for each item remains in the same orientation and direction regardless of the different items' orientation.

1. Tap the **At Each Item** node on the Program Tree.
2. On the At Each Item screen, tap **Next**.
3. Tap the **Move Here** button. Then, hold the **Auto** button or use the **Manual** button to move the robot to the ReferencePoint. Tap the **Continue** button. Tap **Next**.
4. Tap **Set Waypoint** to teach the Approach Waypoint (see). Tap **Next**.
5. Repeat Step 3.
6. Tap **Set Waypoint** to teach the Exit Waypoint (see). Tap **Next**.
7. Tap **Finish**.
8. You can now add appropriate gripper action nodes in the Tool Action folder in the Program Tree.

Copyright © 2009–2025 by Universal Robots A/S. All rights reserved.

Manual Configuration

1. Tap the **At Each Item** node on the Program Tree.
2. On the **At Each Item** start screen, tap **Manual Configuration**.
3. Use the drop-down menus to select a Pattern and a ReferencePoint item. Tap the **Use this ReferencePoint** button to set the ReferencePoint.
4. Move the robot to the ReferencePoint by tapping **Move Here**.
5. Tap the Approach node in the Program Tree to teach the robot the Approach Waypoint (see). The Approach Waypoint remains in the same orientation and direction regardless of the different items' orientation.
6. Tap the At Each Item node in the Program Tree. Repeat Step 4.
7. Tap the **Exit** node in the Program Tree to teach the robot the Exit Waypoint (see).
8. You can now add appropriate gripper action nodes in the Tool Action folder in the Program Tree.

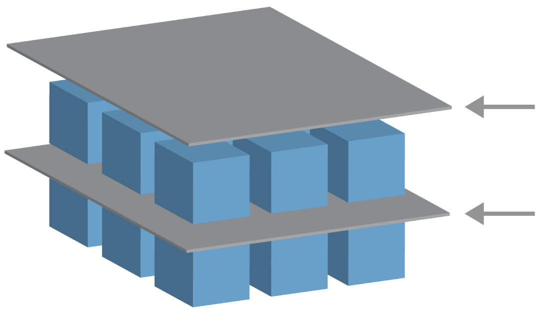
Adding a Separator Between Layers in a Palletizing Sequence

Separators, such as paper or Styrofoam, can be placed between layers in a palletizing sequence. To add separators between layers, follow the instructions below:

1. On the Program Tree, select the **Patterns** node.
2. On the **Patterns** screen, select **Separator** and define the height using the **Separator Height** text box. If the height is not defined, the program will not run.
3. Select **Layers** in the Program Tree. On the Layers screen, select which layers you want the separators to go between (separators are automatically placed between each layer).
4. Tap the **Separator** node in the Program Tree. Tap **Set Separator** to teach the Separator Position.
5. Choose between using the default option (A) Separator Wizard, or (B) Manually Configure the Separator sequence. Instructions for each option are below.

When the wizard is complete, or if you cancel the wizard, a template appears in the Program Tree under **Separator Action**. In addition to the Tool Action folder under the Separator Action node, you can select one of the following folders:

- **Pick Up Separator** to program the robot to pick up separators for palletizing
- **Drop Off Separator** to drop off separators for depalletizing



Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

(A) Separator Wizard

1. Tap the **Separator Action** node on the Program Tree.
2. On the Separator Action screen, tap **Next**.
3. Tap the **Move Here** button and hold the **Auto** button or use the **Manual** button to move the robot to the Separator Point. Tap the **Continue** button. Tap **Next**.
4. Tap **Set Waypoint** to teach the Approach Waypoint (see). Tap **Next**.
5. Repeat Step 3.
6. Tap **Set Waypoint** to teach the Exit Waypoint (see). Tap **Next**.
7. Tap **Finish**.
8. You can now add appropriate action nodes in the Pick Up Separator, Drop Off Separator, and Tool Action folders in the Program Tree.

(B) Manual Configuration

1. Tap the **Separator Action** node on the Program Tree.
2. On the **Separator Action** start screen, tap **Manual Configuration**.
3. Move the robot to the Separator Point by tapping **Move to Separator Point**.
4. Tap the Approach node in the Program Tree to teach the robot the Approach Waypoint (see).
5. Tap the Separator Action node in the Program Tree. Repeat Step 3.
6. Tap the Exit node in the Program Tree to teach the robot the Exit Waypoint (see).
7. You can now add appropriate action nodes in the Pick Up Separator, Drop Off Separator, and Tool Action folders in the Program Tree.

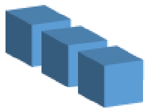
Options to Customize A Palletizing Program

You can customize your palletizing program in the following ways:

- If your pallet needs to be adjusted or re-positioned after you have created a palletizing program, you only need to re-teach the pallet Feature (see) because the palletizing sequence is fixed relative to the Feature. Thus, all other program components automatically adjust to the newly taught position.
- You can edit the properties of the move commands (see).
- You can change the speeds and blends radii (see).
- You can add other program nodes to the At Each Item sequence or the Separator Action sequence.

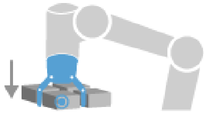
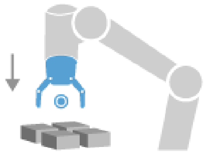
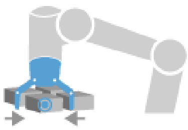
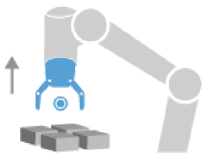
Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Positions

Line		To teach the positions, select each item in the Program Tree: • StartItem1 • EndItem1 Insert the number of items in your sequence using the **Items** text box at the bottom of the screen.
Grid		To teach the positions, select each item in the Program Tree: • CornerItem1 • CornerItem2 • CornerItem3 • CornerItem4 Insert the number of rows and columns in the appropriate text boxes to set the dimensions of the pattern.
Irregular		To teach the positions, select each item in the Program Tree: • Item1 • Item2 • Item3 Tap **Add Item** to add and identify a new item in the sequence.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Actions

Tool Action Point		The location and position you want the robot to be in when conducting an action for each item in a layer. The ToolActionPoint Waypoint is the ReferencePoint by default, but it can be edited in the Program Tree by tapping the ToolActionPoint Waypoint node. When using the wizard, the ReferencePoint is the first position in the first defined layer on the pallet. The ReferencePoint is used to teach the robot the Approach Waypoint, ToolActionPoint Waypoint, and Exit Waypoint for each item in a layer.
Approach		The collision-free position and direction you want the robot to take when approaching an item in a layer.
Tool Action		The action you want the robot attachment to perform for each item.
Exit Waypoint		The position and direction you want the robot to take when moving away from an item in a layer.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

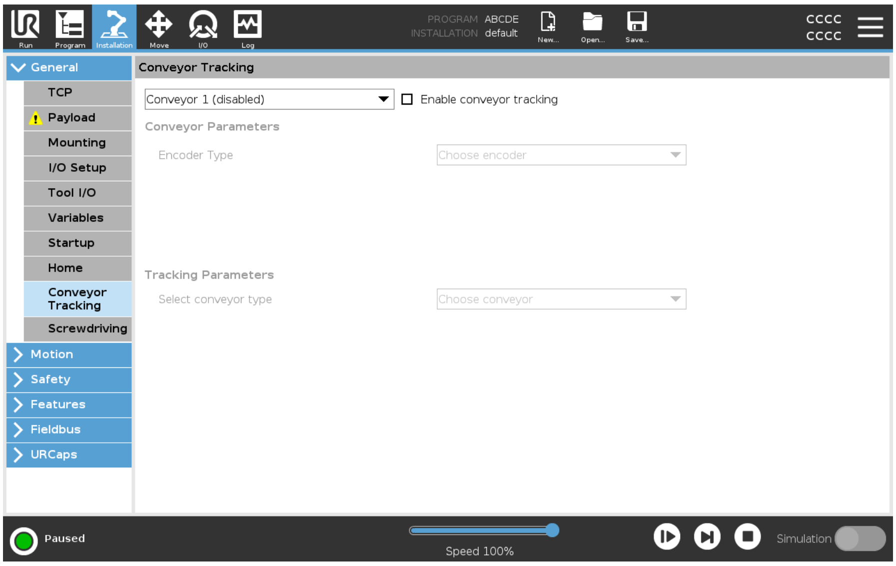
4.9.4. Conveyor Tracking

Description

Conveyor Tracking allows the Robot Arm to track the movement of up to two conveyors. Conveyor Tracking is defined in the Installation Tab.

> **NOTICE**
>
> Using this function at the same time as Force and/or Path Offset can lead to a program conflict.
>
> - Do not use this function together with Force or Path Offset.

The Conveyor Tracking program node is available in the Program Tab under Templates. All movements under this node are allowed while tracking the conveyor, but they are relative to the motion of the conveyor belt. Blends are not allowed when exiting Conveyor Tracking, so the robot stops completely before making the next motion.



Tracking a Conveyor

1. In the Header, tap **Program**.
2. Tap **Templates** and select **Conveyor Tracking** to add a Conveyor Tracking node to the Program Tree. Any movements listed under the Conveyor Tracking node tracks the movement of the conveyor.
3. Under Conveyor Tracking, in the Select Conveyor dropdown list, select **Conveyor 1** or **Conveyor 2** to define which conveyor must be tracked.

> **NOTICE**
>
> If there is a Conveyor Tracking node inside an `If`, `ElseIf` or `Loop`, and the `Check Expression Continuously` option is selected, you can add an `end_conveyor_tracking()` script at the end of the expression to exit conveyor tracking.

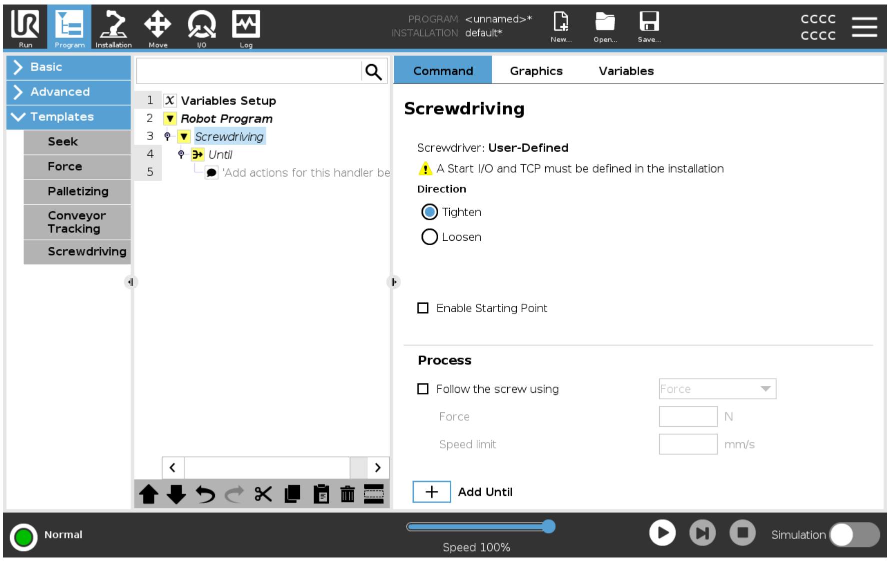
Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

4.9.5. Screwdriving

Description

The **Screwdriving** program node provides an easy way to add a screwdriving application for an attached screwdriver. Configuring the screwdriver and its connections to the robot is defined in the Installation Tab (see General).



Adding a Screwdriving Node

1. In the Header, tap **Program**.
2. Under Templates, tap **Screwdriving**.
3. Select **Tighten** to follow the screw in a tightening direction (in), or select **Loosen** to follow the screw in a loosening direction (out). This selection only impacts the motion of the robot to follow the screw and its measuring calculations.
4. In the **Program Selection** field, you can select a screwdriver program, depending on the **Program Selection** signals in the Installation.
5. Select **Enable Starting Point**, to add a MoveL to the Program Tree that is executed when the screwdriver is already running.
   Select **Enable Machine Error Handler**, to add a corrective measure, if required, to the Program Tree before the screwdriving actions begins.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Affect screwdriving action

Selecting **Follow the Screw**, under **Process**, to affect the screwdriving action in the following ways:

- **Force**: Select **Force** to define how much force is exerted on a screw. Then select **Speed Limit**, so the robot moves at this speed as long as it does not come into contact with the screw.

> **CAUTION**
> Place the screwdriver bit above the screw before starting a screwdriving program. Exerting any force on the screw can affect the screwdriving program performance.

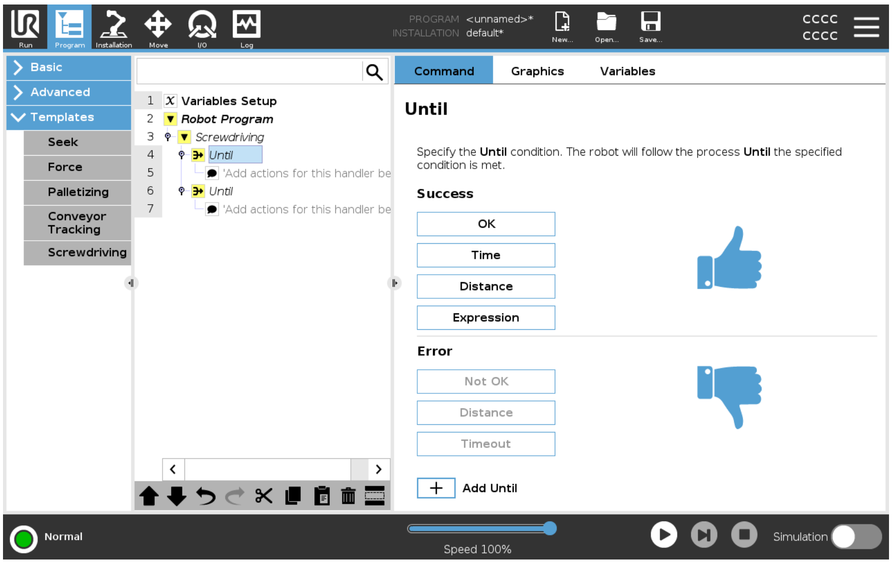
- **Speed**: Select a fixed **Tool Speed** and **Acceleration**for the robot to follow the screw.
- **Expression**: Similar to the **If** command (see 4.8.4 If on page 135), select **Expression** to describe the condition under which the robot follows the screw.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Screwdrive Until

Description

The Screwdriving program node includes a mandatory until success **Until** node that defines stop criteria for the screwdriving process.



You can define the following stop criteria:

- **Success**: Screwdriving continues until completion is detected using your selected option. You can only add one success condition.
- **Error**: Screwdriving continues until an error is detected using your selected option/s. You can add more than one error condition.

Criteria

Success

- **OK**: Screwdriving continues until an OK signal from the screwdriver is detected.
- **Time**: Screwdriving continues upto a defined time.
- **Distance**: Screwdriving continues upto a defined distance.
- **Expression**: Screwdriving continues until a custom expression condition is met.

Error

- **Not OK**: Screwdriving stops when a NOT OK signal from the screwdriver is detected.
- **Distance**: Screwdriving stops when the defined distance is exceeded.
- **Timeout**: Screwdriving stops when the defined time is exceeded.

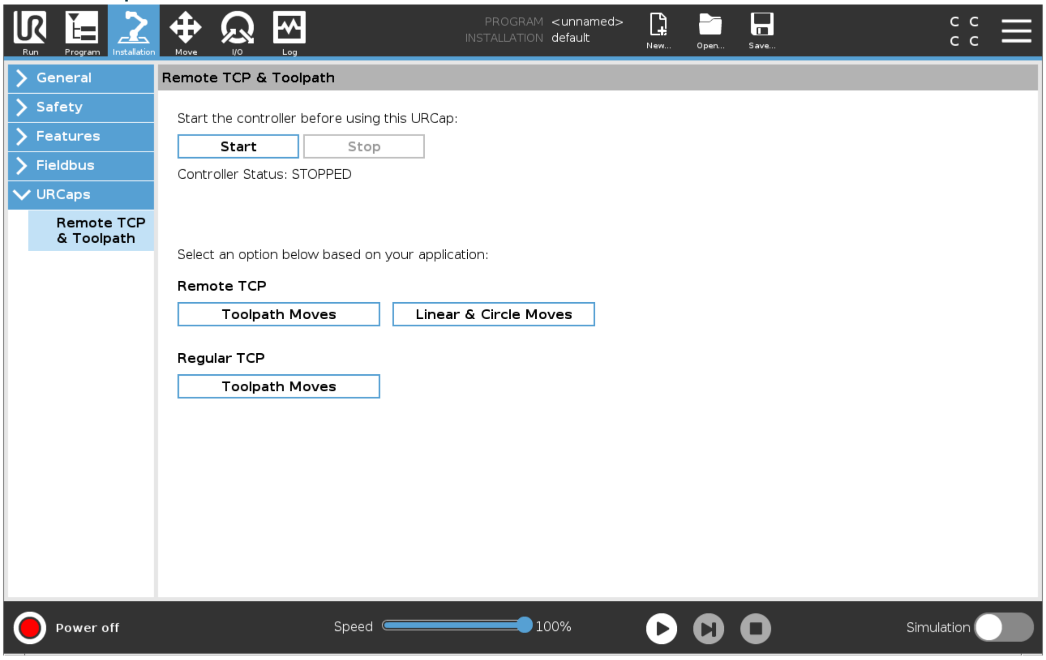
Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

4.10. URCaps

Description

The Remote TCP and Toolpath URCap allows you to set Remote Tool Center Points (RTCP), where the tool center point is fixed in space, relative to the base of the robot. The Remote TCP and Toolpath URCap also allows for programming waypoints and circle moves, and generating robot motion based on imported toolpath files defined in third-party CAD/CAM software packages.

The Remote TCP URCap requires your robot to be registered before use (see Robot Registration and License File on page 241). The RTCP works in applications that require the robot to grasp and move items, relative to a fixed tool. The RTCP is used together with the RTCP_MoveP and RTCP_CircleMove commands to move a grasped part with constant speed, relative to the fixed tool.



Similar to a regular TCP (see) you can define and name an RTCP in the Installation Tab's Setup.

You can also complete the following actions:

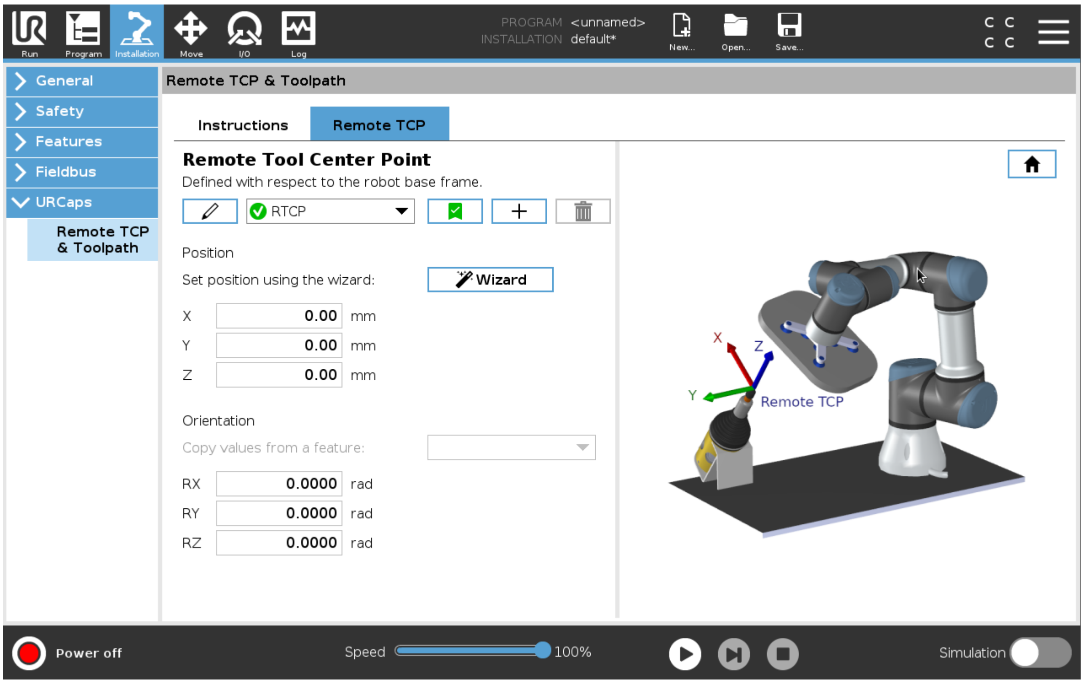
- Add, rename, modify, and remove RTCPs
- Understand the default and active RTCP
- Teach RTCP position
- Copy RTCP orientation

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Setting the Remote TCP from a Feature

Set an RTCP using a Feature to allow the robot to be jogged relative to the RTCP while creating RTCP Waypoints and RTCP Circle Moves.

1. Tap the plus icon to create a new RTCP **RTCP**. Or select an existing RTCP in the drop-down menu.
2. Tap the **Copy values from a point feature** drop-down menu and select a Feature. Verify the RTCP orientation values update to match that of the selected Feature.



Remote TCP Movement Types

RTCP_ MoveP

Similar to a regular MoveP, the RTCP_MoveP defines the tool speed and acceleration the Robot Arm moves relative to the Remote TCP. See .

RTCP Circle move

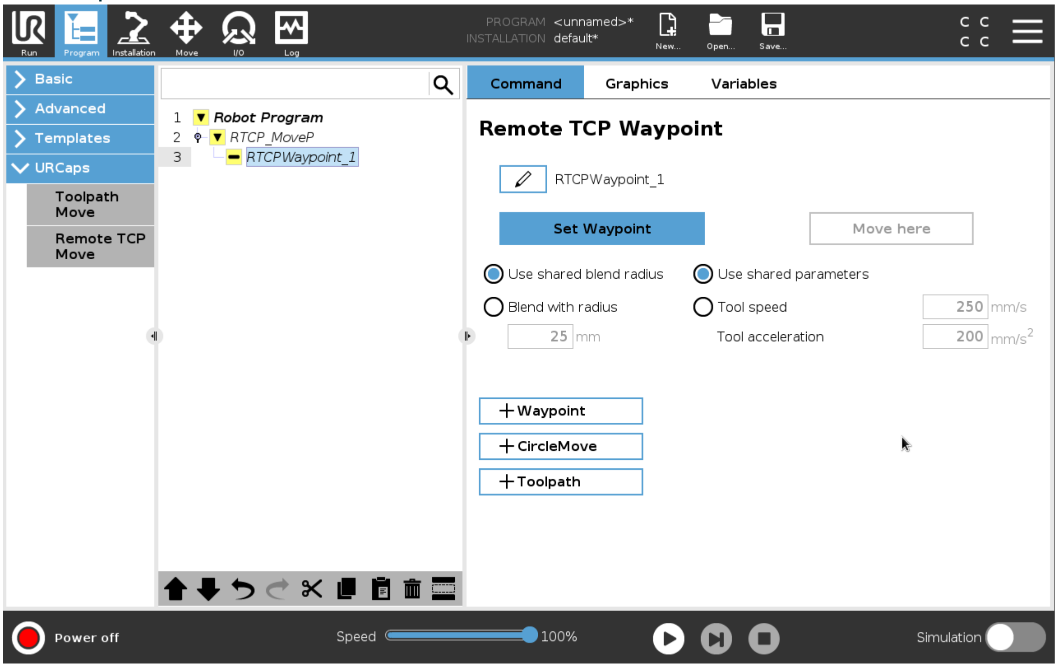
Similar to a regular Circle move, the RTCP Circle move can be added to an RTCP_MoveP to make circular movements. See .

> **NOTICE**
>
> The maximum speed of a Circle Move may be lower than the specified value. The circle radius is *r*, the maximum acceleration is *A*, and the maximum speed cannot exceed $\sqrt{Ar}$ due to centripetal acceleration.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Remote TCP Waypoint

Similar to regular waypoints, RTCP Waypoints allow a tool to move linearly using constant speed and circular blends. The default blend radius size is a shared value between all the waypoints. A smaller blend radius size sharpens the path turn. A larger blend radius size smoothens the path. RTCP Waypoints are taught by physically moving the Robot Arm to a desired position.



Teaching Remote TCP Waypoints

1. In the Program Tab, insert an **RTCP_MoveP** node.
2. On the RTCP_MoveP node, tap **Set** to bring up the Move screen.
3. On the Move screen, use **Teach Mode** or **Jog** to position the robot in a desired configuration.
4. Tap the green check mark to validate.

Configuring an RTCP Waypoint

Use blends to enable the robot to smoothly transition between two trajectories. Tap **Use Shared Blend Radius** or tap **Blend with radius** to set the blend radius for a waypoint from an RTCP_MoveP.

> **NOTICE**
> A physical time node (e.g. Move, Wait) cannot be used as a child of an RTCP_MoveP node. If an unsupported node is added as a child to an RTCP_MoveP node, the program fails to validate.

Remote TCP Toolpath

The Remote TCP and Toolpath URCap generates robots motions automatically, making it easier to follow complex trajectories accurately.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Configuring a Remote TCP Toolpath

1. Select **Remote TCP Toolpath Moves** on the Remote TCP & Toolpath URCap Home Page to enter the workflow.
2. Follow the instructions under the **Instructions Tab**.

A Remote TCP Toolpath Move requires the following major components:

- Toolpath file
- Remote TCP
- Remote TCP PCS

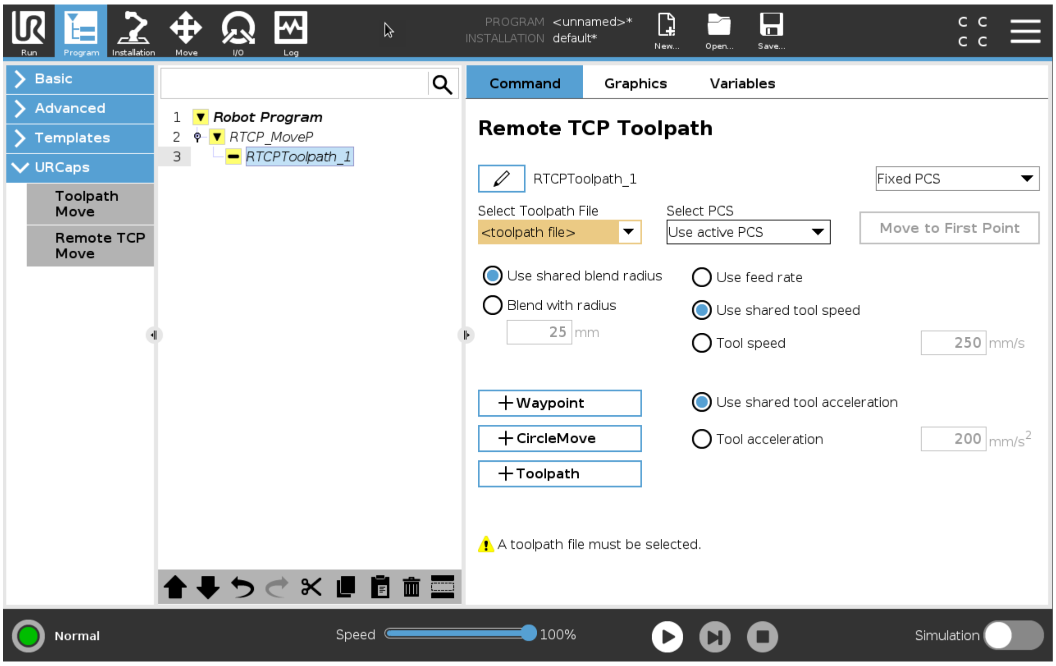
Configuring a Toolpath using CAD/CAM Software

A toolpath defines the orientation, trajectoy, speed or (feed rate) and travel direction of the tool.

1. Create or import a CAD model of a part.
2. Set up a Part Coordinate System (PCS) fixed to the part.
3. Create a toolpath relative to the PCS based on part features
4. Simulate the toolpath motion to verify it meets expectation.
5. Export the toolpath into a G-code file with .nc file extension.

Importing a G-code Toolpath into PolyScope

1. Load the toolpath files in the root directory of a USB stick. Toolpath files must have the .nc extension
2. Insert the USB stick into the Teach Pendant.
3. In the Header, tap Installation, tap URCaps, select Remote TCP & Toolpath, select Remote TCP - Toolpath Moves, and select Toolpath.
4. Select which toolpath files to import into Polyscope.



Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

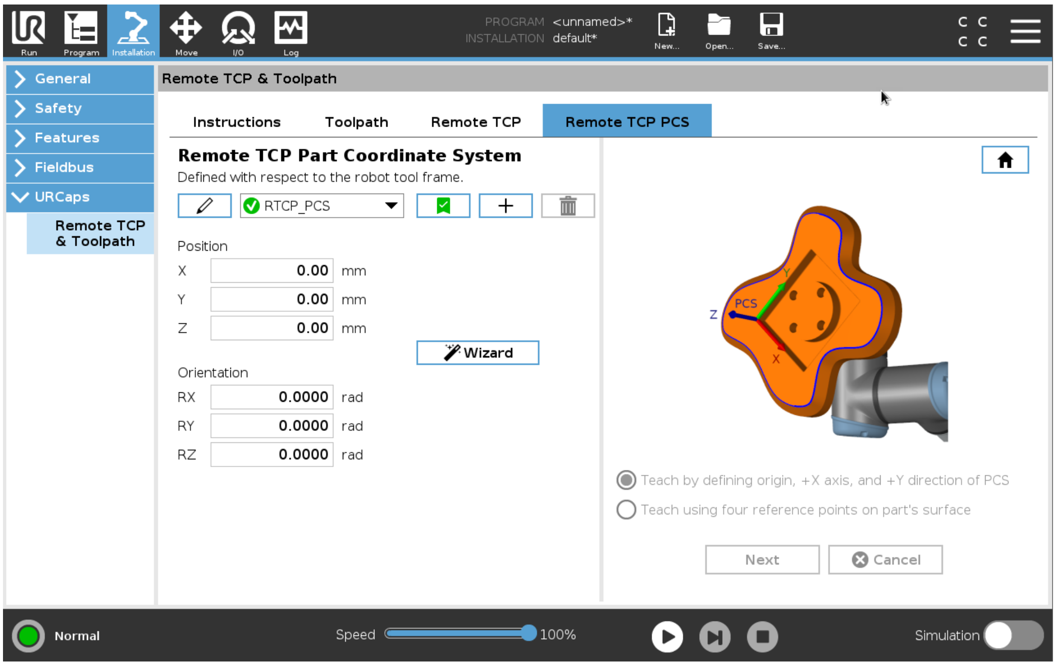
Remote TCP

Configuring a Remote TCP for Toolpath Moves

1. At the first waypoint in the CAM environment, determine the tool orientation.
2. Use Freedrive to manually grasp the part with the gripper.
3. Select the location of the Remote TCP
4. Use the position wizard to obtain the positive values.
5. Adjust the robot until the desired part pose for approaching the remote TCP is acheived.
6. Envision the tool orientation at the first waypoint on the physical part. The positive Z-axis direction should point away fromthe part surface.
7. Create a Plane Feature with the same orientation as envisioned in the previous step.
8. Set the Remote TCP orientation by copying values from the Plane Feature. The desired part pose is maintained while the toolpath is executed.

Remote TCP PCS

The Remote TCP Part Coordinate System (PCS) is defined as fixed relative to the robot tool flange. Tap the wand, on the PolyScope screen, to activate the wizard to teach the Remote TCP PCS. You can use either of the teaching methods described below.



Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Configuring a Remote TCP PCS

Use this method if the PCS can be set on the part surface.

1. Use freedrive to manually grasp the part with the gripper.
2. Select a Remote TCP to teach the reference points. For high accuracy, temporarily set up a sharp Remote TCP to complete this teaching process.
3. Jog the robot for the Remote TCP to touch the origin, positive X-axis and the positive Y-axis direction of the PCS on the part.
4. Tap Set to conclude the teaching process. The position and orientation values are populated automatically.

Otherwise, use this method.

1. Select three or four reference points on the part surface.
2. In the CAD/CAM software, record the X, Y, Z coordinates, relative to the PCS of the selected reference points.
3. Use freedrive to to manually grasp the part with the gripper.
4. Select a Remote TCP to teach the reference points. For high accuracy, temporarily set up a sharp Remote TCP to complete this teaching process.
5. Enter the coordinates for the first reference point.
6. Jog the robot for the Remote TCP to touch the first reference point on the part.
7. Repeat steps five and six for the other reference points.

Setting a Variable PCS

For advanced use cases, where the part is not grasped with high consistency, you can set a Variable PCS to adjust the toolpath moves according to the part location and orientation relative to the robot tool flange.
You can create a pose variable tied to an external sensor that can detect the PCS location and orientation.

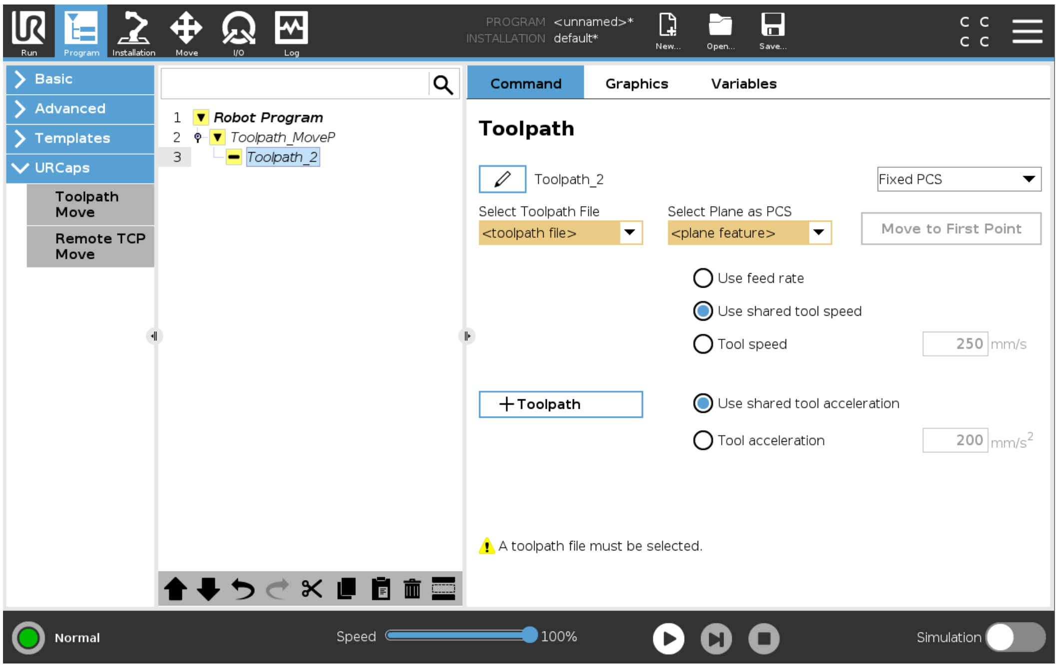
1. Set up an external sensor that detects the PCS location and orientation. You must convert the sensor output to the robot tool flange frame.
2. Verify the PCS is set up relative to the part and the location and orientation are detectable by the external sensor.
3. In PolyScope, create a pose variable tied to the external sensor output as a variable PCS. Give it a distinct name, for example, **variable_rtcp_pcs_1**.
4. Insert an **RTCP Toolpath Node**.
5. At the top right corner of the program page, in the drop-down menu, select **Variable PCS**.
6. In the **Select PCS** drop-down menu, select **variable_rtcp_pcs_1**.
7. Create an Assignment or Script node to update **variable_rtcp_pcs_1** before the RTCP Toolpath Node.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Configuring a Remote TCP Toolpath Node

The following section explains how to use a variable PCS in a Remote TCP Toolpath node.

1. Access the Program Tab and tap **URCaps**.
2. Select **Remote TCP Move** to insert an RTCP_MoveP node.
3. Select a TCP and set the motion parameters: tool speed, tool acceleration and blend radius.
4. Tap **+Toolpath** to insert an RTCPToolpath node. Delete the RTCPWaypoint node that was created by default, if it is not needed.
5. Select a toolpath file and the corresponding Remote TCP PCS fromthe drop-down menus.
6. Adjust the motion parameters if different values are to be applied to the RTCPToolpath node.
7. Tap **Move to First Point** to verify the grasped part approaches the Remote TCP as expected.
8. Test the program in the simulation mode at a low speed to confirm the configurations.



> **NOTICE**
> You can ensure the robot motion is identical, each time the toolpath is executed, by adding a MoveJ with a **Use Joint Angles** set to move to a fixed joint configuration before executing the toolpath. See 4.7.1 Move on page 87

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Regular TCP Toolpath Moves

Similar to configuring a Remote TCP Toolpath Move, a regular TCP Toolpath Move requires the following:

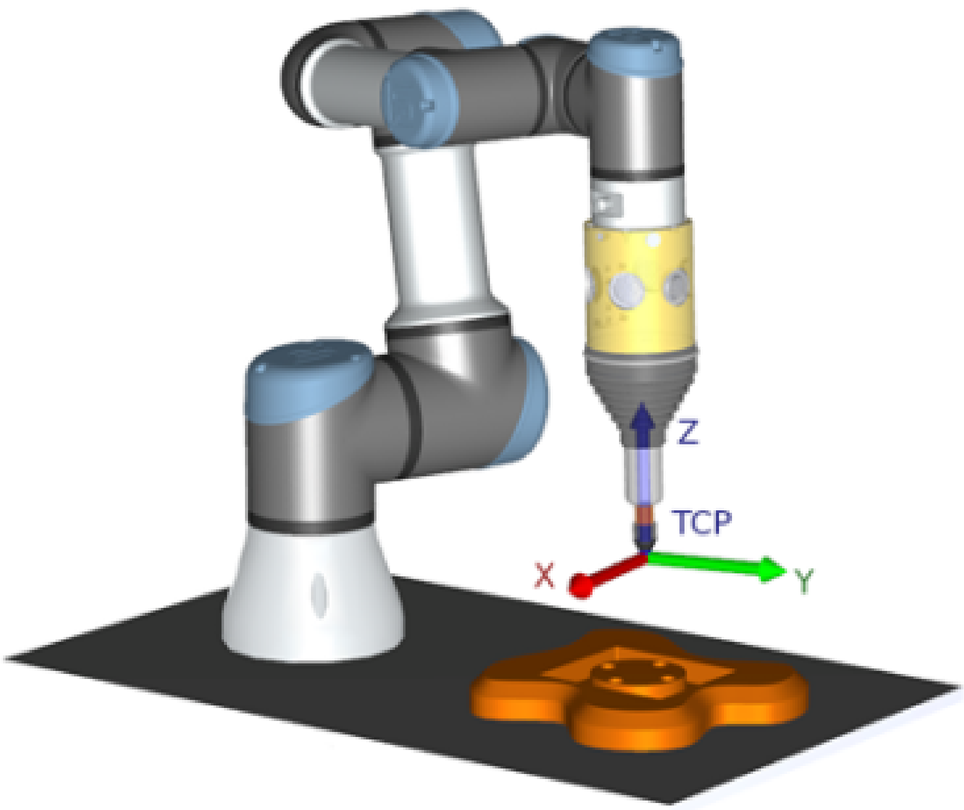
- Toolpath file
- Regular TCP
- Plane Feature as a PCS

Configuring and Importing a Toolpath File

This is similar to configuring a Toolpath (see Configuring a Toolpath using CAD/CAM Software on page 166) and importing Toolpath (see Importing a G-code Toolpath into PolyScope on page 166).

Configuring a Regular TCP

- Follow the instructions to configure a Regular TCP.
- Verify the positive Z-axis of the tool points away from the part surface.



Configuring a Plane Feature PCS

1. Create a plane feature by **Adding a plane** or **Teaching a plane**. See .
2. Fix the part relative to the robot base.
3. Verify the correct TCP is us to create the plane feature. For high accuracy, temporarily set up a sharp Remote TCP to complete this teaching process.
4. Jog the robot for the Remote TCP to touch the origin, positive X-axis and the positive Y-axis direction of the PCS on the part.
5. Finish the teaching process and confirm the PCS position and orientation.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Configuring a Toolpath Node

1. Access the Program Tab and tap **URCaps**.
2. Select a TCP and set the motion parameters: tool speed, tool acceleration and blend radius. Select **Spin tool freely around its Z-axis**. Do not select if the tool must follow the orientation around Z-axis defined in the toolpath file.
3. Tap **+Toolpath** to insert a Toolpath node.
4. In the drop-down menu, select a toolpath file and the corresponding PCS (Plane Feature).
5. Adjust the motion parameters if different values are to be applied to the Toolpath node.
6. Tap **Move to First Point** to verify the tool can move to the first point of the toolpath.
7. Run the program in the simulation mode, at a low speed, to confirm the configurations are correct.

> **NOTICE**
> You can ensure the robot motion is identical, each time the toolpath is executed, by adding a MoveJ with a **Use Joint Angles** set to move to a fixed joint configuration before executing the toolpath. See 4.7.1 Move on page 87

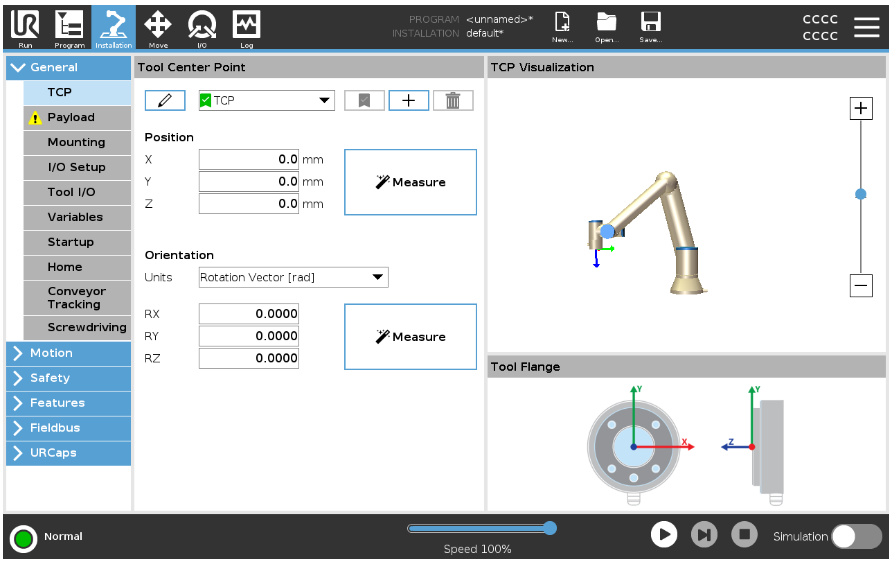
Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

5. Installation Tab

Description The Installation Tab allows you to configure the settings which affect the overall performance of the robot and PolyScope.

5.1. TCP Configuration

Description A **Tool Center Point** (TCP) is a point on the robot's tool. Each TCP contains a translation and a rotation relative to the center of the tool output flange.
When programmed to return to a previously stored waypoint, a robot moves the TCP to the position and orientation saved within the waypoint. When programmed for linear motion, the TCP moves linearly.



Position The X, Y, Z coordinates specify the TCP position. When all values (including orientation) are zero, the TCP coincides with the center point of the tool output flange and adopts the coordinate system depicted on the screen.

Orientation The RX, RY, RZ coordinate boxes specify the TCP orientation. Similar to the Move Tab, use the Units drop down menu above the RX, RY, RZ boxes to select the orientation coordinates

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Adding, Renaming, Modifying and Removing TCPs

You can start configuring a new TCP with the following actions:

- Tap the [+] to define a new TCP with a unique name. The new TCP is available in the dropdown menu.
- Tap the [✎] to rename a TCP.
- Tap the [🗑] to remove a selected TCP. You cannot remove the last TCP.

Active TCP

When moving linearly, the robot always uses the active TCP to determine the TCP offset. The active TCP can be changed using a Move command (see 4.2.2 Command Tab on page 73) or a Set command. The motion of the active TCP is visualised on the Graphics Tab (see 4.2.3 Graphics Tab on page 76).

Default TCP

The Default TCP must be set as the active TCP before running a program.

- Select the desired TCP and tap **Set as default** to set a TCP as the default.

The green icon in the available drop-down menu indicates the default configured TCP.

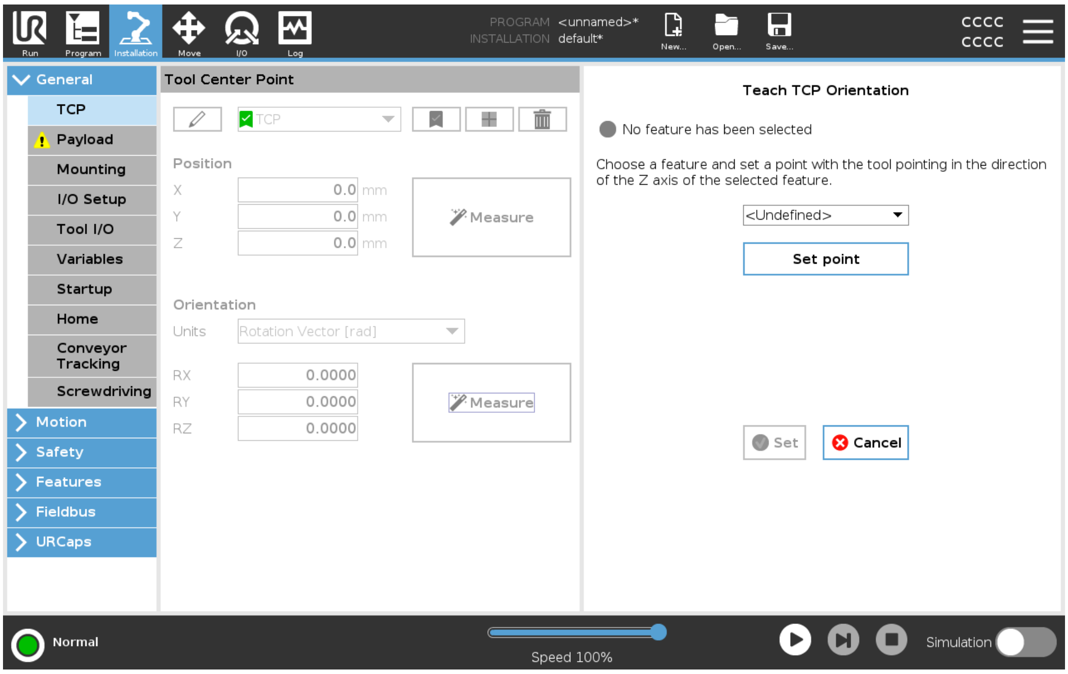
Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

5.1.1. Teaching TCP Position

Teaching TCP orientation

TCP position coordinates can be calculated automatically as follows:

1. Tap **Measure**.
2. Choose a fixed point in the workspace of the robot.
3. Use the position arrows on the right side of the screen to move the TCP from at least three different angles and to save the corresponding positions of the tool output flange.
4. Use the **Set** button to apply the verified coordinates to the appropriate TCP. The positions must be sufficiently diverse for the calculation to work correctly. If they are not sufficiently diverse, the status LED above the buttons turns red.



Though three positions are sufficient to determine the TCP, a fourth position can be used to further verify the calculation is correct. The quality of each saved point, with respect to the calculated TCP, is indicated using a green, yellow, or red LED on the corresponding button.

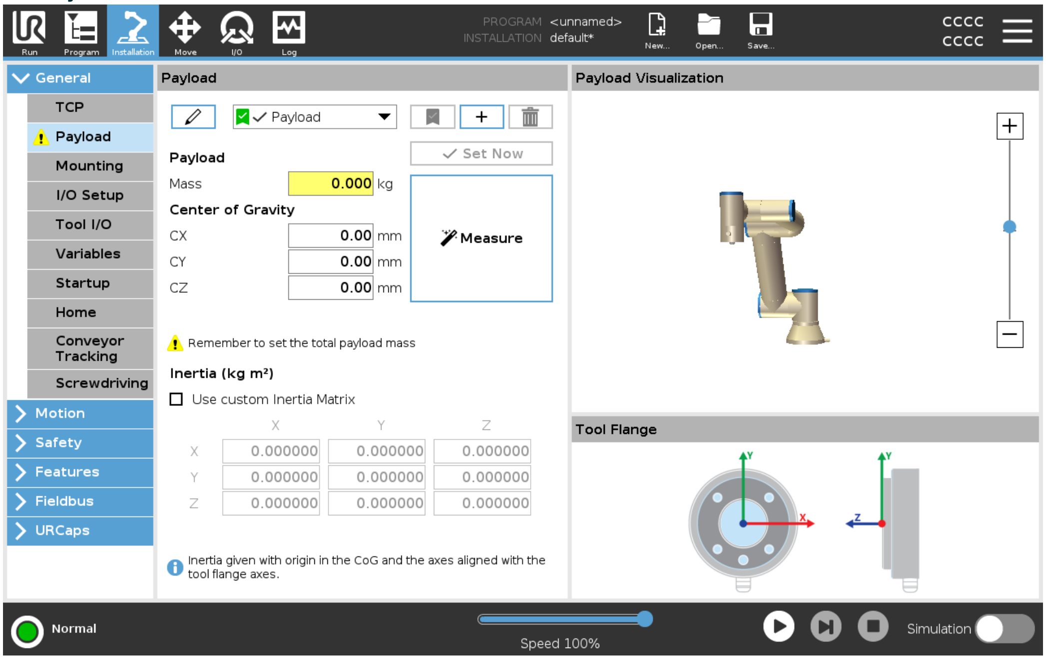
1. Tap **Measure**.
2. Select a feature from the drop-down list. (See 5.16 Features on page 197) for additional information on defining new features
3. Tap **Set point** and use **Move tool arrows** to a position where the tool's orientation and the corresponding TCP coincide with the selected features's coordinate system.
4. Verify the calculated TCP orientation and apply it to the selected TCP by tapping **Set**.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

5.2. Payload

Description

You must set the Payload, the CoG and the inertia for the robot to perform optimally. You can define multiple Payloads, and switch between them in your program. This is useful in Pick and Place applications, for example, where the robot picks up and releases an object.



Adding, Renaming, Modfying and Removing Payloads

You can start configuring a new Payload with the following actions:

- Tap the + to define a new Payload with a unique name. The new payload is available in the drop-down menu.
- Tap the to rename a Payload.
- Tap the to remove a selected Payload. You cannot remove the last Payload.

Active Payload

The checkmark in the drop-down indicates which payload is active ✓ Payload. The active Payload can be changed using the ✓ Set Now.

Default Payload

The default Payload is set as the active Payload before the program starts.

- Select the desired Payload and tap **Set as default** to set a Payload as the default.

The green icon in the drop-down menu indicates the default configured Payload ✓ Payload.

Setting the Center of Gravity

Tap the fields `CX`, `CY` and `CZ` to set the center of gravity. The settings apply to the selected Payload.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Payload Estimation

This feature allows the robot to help set the correct Payload and Center of Gravity (CoG).

Using the Payload Estimation Wizard

1. In the Installation Tab, under General, select **Payload**.
2. On the Payload screen, tap **Measure**.
3. In the Payload Estimation Wizard tap **Next**.
4. Follow the steps in the Payload Estimation Wizard to set the four positions.
   Setting the four positions requires moving the robot arm into four different positions.
   The load of the payload is measured at each position.
5. Once all measurements are complete, you can verify the result and tap **Finish**.

NOTICE

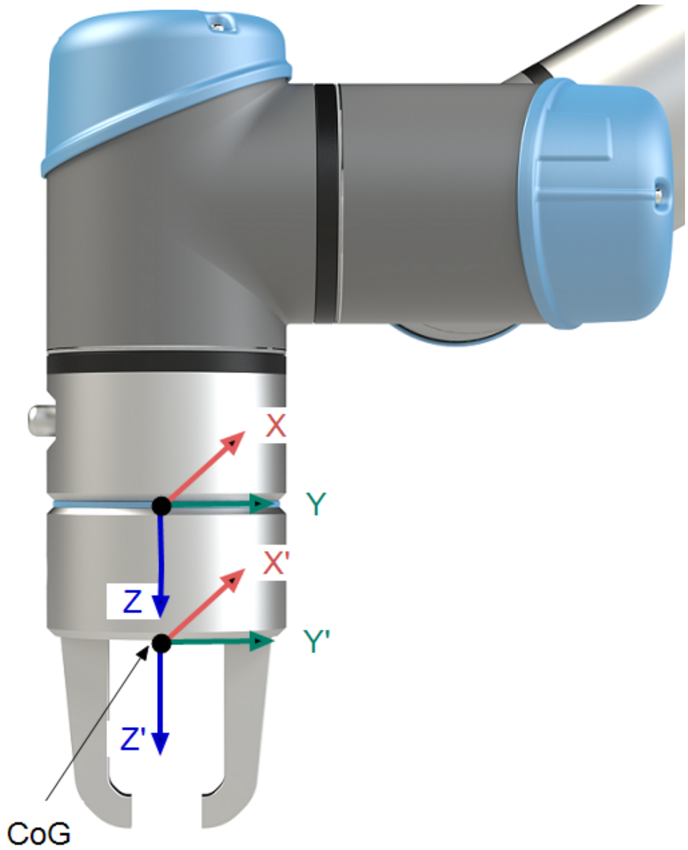
Follow the these guidelines for best Payload Estimation results:

- Ensure the TCP positions are as different as possible from each other
- Perform the measurements within a short timespan
- Avoid pulling on the tool and/or attached payload before and during estimation
- Robot mounting and angle must be correctly defined in the installation

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Setting Inertia Values

You can select **Use custom Inertia Matrix** to set inertia values.
Tap the fields: `IXX`, `IYY`, `IZZ`, `IXY`, `IXZ` and `IYZ` to set the inertia for the selected Payload.
The inertia is specified in a coordinate system with the origin at the Center of Gravity (CoG) of the payload and the axes aligned with the tool flange axes.
The default inertia is calculated as the inertia of a sphere with the user specified mass, and a mass density of $1 g/cm^3$



Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

5.3. Mounting

Description

Specifying the mounting of the robot arm serves two purposes:

1. To make the robot arm appear correctly on the PolyScope screen.
2. To tell the controller about the direction of gravity.

WARNING
Failure to mount the robot arm correctly can result in frequent stops.

WARNING
Verify and use the correct installation settings. Save and load the installation files with the program.

If the robot arm is mounted in one of the ways listed below, adjustment is required.

- ceiling mounted
- wall mounted
- mounted at an angle

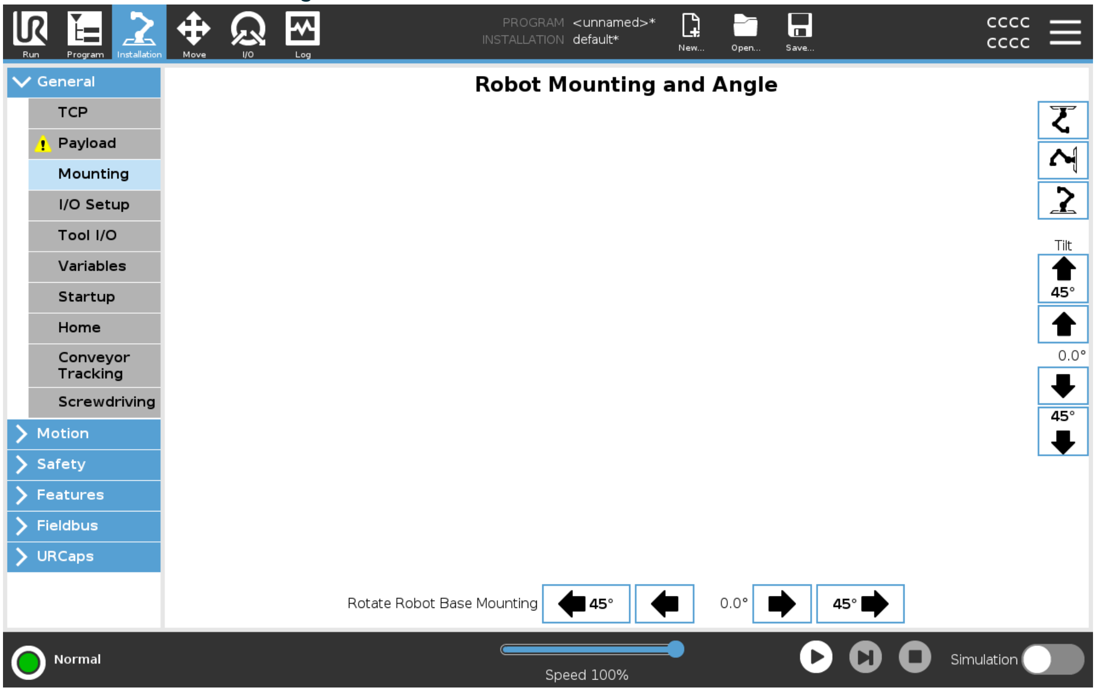
On the Robot Mounting and Angle screen, use the buttons on the right to set the angle of the robot arm mounting. The first three buttons set the angle as follows:

- ceiling (180°)
- wall (90°)
- floor (0°)

The **Tilt** buttons set an arbitrary angle.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Use the buttons on the lower part of the screen to rotate the mounting of the Robot arm to match the actual mounting.



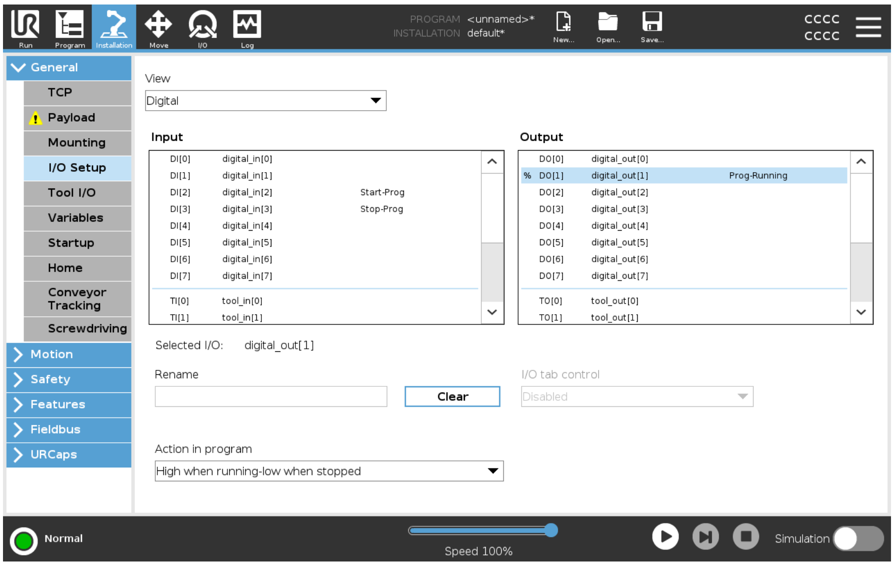
An advanced dynamics model gives the robot arm smooth and precise motions and allows the robot arm to hold itself in Freedrive. For this reason, it is important to mount the robot arm correctly.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

5.4. I/O Setup

Description

Use the I/O Setup screen to define I/O signals and configure actions with the I/O tab control. The types of I/O signals are listed under **Input** and **Output**.
You can use a fieldbus, for example, Profinet and EtherNet/IP, to access the general purpose registers.
If you enable the Tool Communication Interface (TCI), the tool analog input becomes unavailable.



> **NOTICE**
>
> When starting programs from an I/O or fieldbus input, the robot can begin movement from the position it has, there will not be any manual movement to the first waypoint via PolyScope required.

I/O Signal Type

To limit the number of signals listed under **Input** and **Output**, use the **View** drop-down menu to change the displayed content based on signal type.

Assigning User-defined Names

You can name the Input and Output signals to easily identify the ones being used.

1. Select the desired signal.
2. Tap the text field to type a name for the signal.
3. To reset the name to default, tap **Clear**.

You must provide a user-defined name for a general purpose register to make it available in the program (i.e., for a **Wait** command or the conditional expression of an **If** command). The **Wait** and **If** commands are described in (4.7.4 Wait on page 119) and (4.8.4 If on page 135), respectively. You can find named general purpose registers in the **Input** or **Output** selector on the **Expression Editor** screen.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

I/O Actions and I/O Tab Control

You can use Physical and Fieldbus digital I/Os to trigger actions or react to the status of a program.

I/O Tab Control

Use I/O Tab Control to specify whether an output is controlled on the I/O tab (by either programmers, or both operators and programmers), or if it is controlled by the robot programs.

Available Input Actions

Command	Action
Start	Starts or resumes the current program on a rising edge (only enabled in Remote Control)
Stop	Stops the current program on a rising edge
Pause	Pauses the current program on a rising edge
Freedrive	When the input is high, the robot goes into freedrive (similar to the freedrive button). The input is ignored if other conditions disallow freedrive.

WARNING

If the robot is stopped while using the Start input action, the robot slowly moves to the first waypoint of the program before executing that program.
If the robot is paused while using the Start input action, the robot slowly moves to the position from where it was paused before resuming that program.

Available Output Actions

Action	Output state	Program state
Low when not running	Low	Stopped or paused
High when not running	High	Stopped or paused
High when running, low when stopped	Low High	Running, Stopped or paused
Low on unscheduled stop	Low	Program terminated unscheduled
Low on unscheduled stop, otherwise High	Low High	Program terminated unscheduled Running, stopped or paused
Continuous Pulse	Alternates between high and low	Running (pause or stop the program to maintain the pulse state)

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Program Termination Cause

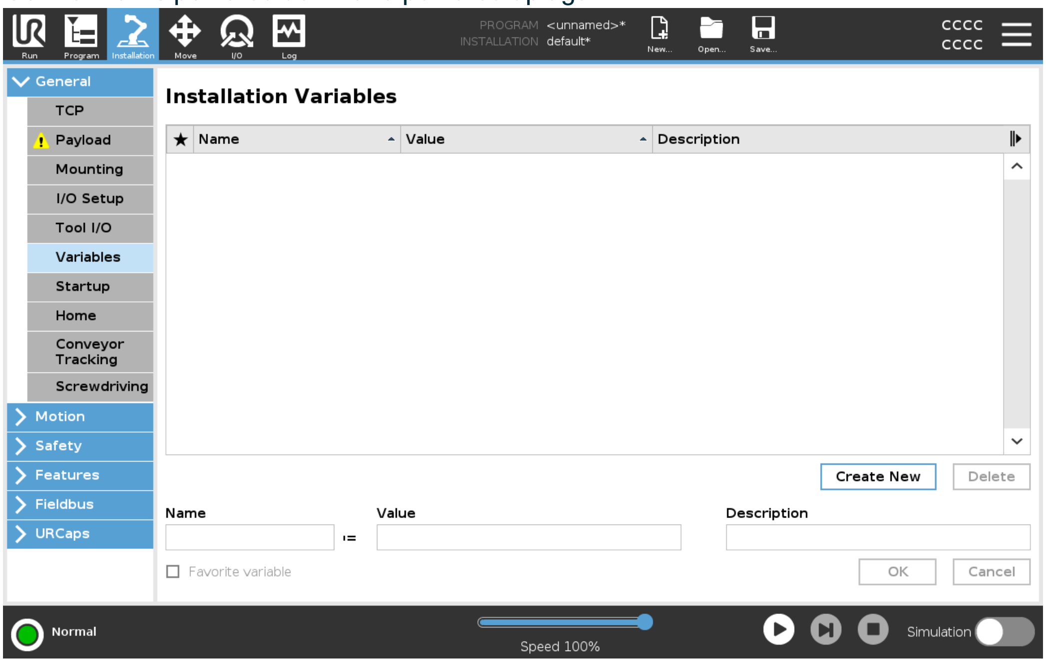
An unscheduled program termination can occur for any of the reasons listed below:

- Robot stop
- Fault
- Violation
- Runtime exception

5.5. Installation Variables

Description

Variables created on the **Installation Variables** pane are called installation variables and are used like normal program variables. Installation variables are distinct because they keep their value even if a program stops then starts again, and when the robot arm and/or Control Box is powered down and powered up again.



Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Creating installation variables

Installation variable names and values are stored with the installation, so you can use the same variable in multiple programs.
Installation variables and their values are saved automatically every 10 minutes during program execution, also when the program is paused and when it is stopped.

To create an installation variable

1. Tap **Create New** and a new variable name is suggested in the **Name** field.

   You can edit the variable name as desired.
2. In the **Value** field, set a value for the new variable.

   You cannot save a variable without first setting the Value.
3. You can describe the new installation variable in the **Description** field.
4. You can set the new variable as favorite by checking the **Favorite variable** box.
5. Tap **OK** to add the new variable to the Installation Variables list.

Designating favorite installation variables

Designating a favorite installation variable allows the installation variable to be part of the set of favorite variables displayed when you select it to only show favorite variables on the Variables tab in the Program Tab screen and on the Run Tab screen.

To designate an installation variable as favorite

1. In the Header, tap **Installation**.
2. Under General, select **Variables**.

   The variables are listed under **Installation Variables**.
3. Select the desired variables.
4. Check the **Favorite variable** box.
5. Tap **Run** to return to your variable display.

Managing installation variables

To edit installation variables

1. Select the desired variable in the Installation Variables list.
2. You can edit the **Value**, the **Description**, or the **Favorite variable**.

   You cannot edit the variable name in this step.

Changes to edited installation variables take immediate effect.
To delete installation variables

1. Select the desired variable and tap **Delete**.
2. Select **Delete Variable** in the confirmation pop-up

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Variable descriptions

You can add information to your variables by adding variable descriptions in the Description column. You can use the variable descriptions to convey the purpose of the variable and/or the meaning of its value to operators using the Run tab screen and/or other programmers.
Variable descriptions (if used) can be up to 120 characters, displayed in the Description column of the variables list on the Run tab screen and the Variables tab screen.
A variable description spans multiple lines to fit the width of the Description column if necessary. You can also collapse and expand the Description column by using the buttons shown below.

To collapse/expand the Description column

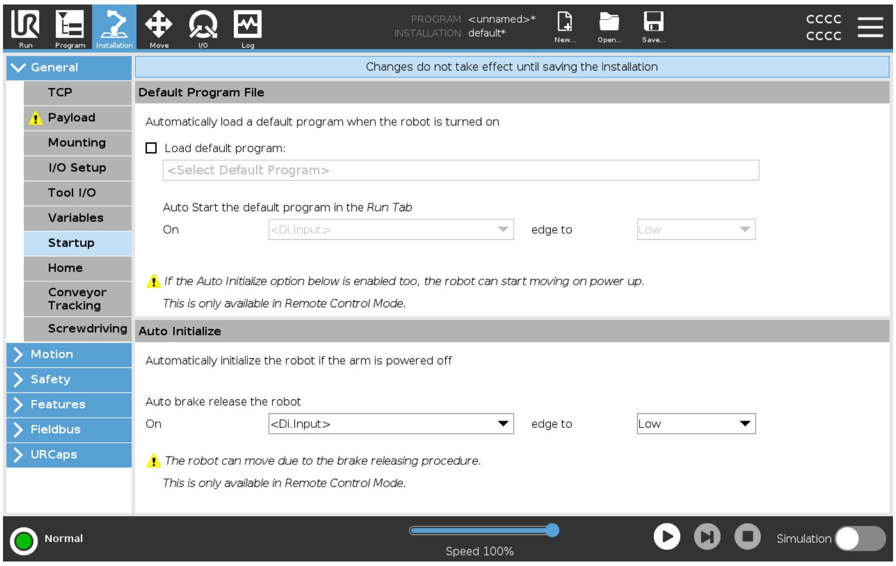
1. Tap **Collapse** to collapse the Description column.
2. Tap **Expand** to expand the Description column.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

5.6. Startup

Description

The Startup screen contains settings for automatically loading and starting a default program, and for auto-initializing the Robot arm during power up.



WARNING

1. When autoload, auto start and auto initialize are enabled, the robot runs the program as soon as the Control Box is powered up as long as the input signal matches the selected signal level. For example, the edge transition to the selected signal level will not be required in this case.
2. Use caution when the signal level is set to LOW. Input signals are low by default, leading the program to automatically run without being triggered by an external signal.
3. You must be in **Remote Control** Mode before running a program where auto start and auto initialize are enabled.

Loading a Startup Program

A default program is loaded after the Control Box is powered up. Furthermore, the default program is auto loaded when the **Run Program** screen (see 3 Run Tab on page 61) is entered and no program is loaded.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

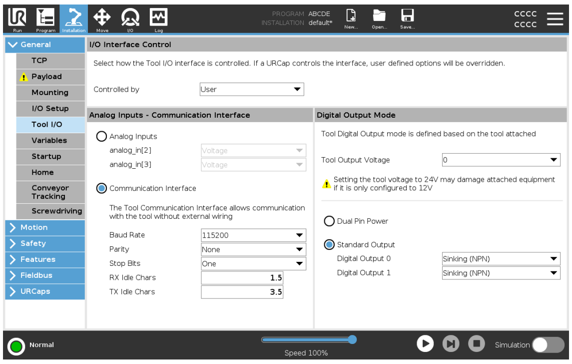
Starting a Startup Program

The default program is auto started in the **Run Program** screen. When the default program is loaded and the specified external input signal edge transition is detected, the program is started automatically.
On Startup, the current input signal level is undefined. Choosing a transition that matches the signal level on startup starts the program immediately. Furthermore, leaving the **Run Program** screen or tapping the Stop button in the Dashboard disables the auto start feature until the Run button is pressed again.

5.7. I/O Interface Control

Description

The I/O Interface Control allows you to switch between user control and URcap control.



to use the I/O Interface Control

1. Tap the Installation tab and under General, tap Tool I/O
2. Under I/O Interface Control, select User to access the Tool Analog Inputs and/or Digital Output Mode settings. Selecting a URCap removes access to the Tool Analog Inputs and the Digital Output Mode settings.

> **NOTICE**
> If a URCap controls an end-effector, such as a gripper, then the URCap requires control of the Tool IO Interface. Select the URCap in the list, to allow it to control the Tool IO Interface.

5.7.1. Analog Input - Communication Interface

Description

The Tool Communication Interface (TCI) enables the robot to communicate with an attached tool via the robot tool analog input. This removes the need for external cabling. Once the Tool Communication Interface is enabled, all tool analog inputs are unavailable

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Tool Communication Interface

1. Tap the Installation tab and under General tap Tool I/O.
2. Select Communication Interface to edit TCI settings.
   Once the TCI is enabled, the tool analog input is unavailable for the I/O Setup of the Installation and does not appear in the input list. Tool analog input is also unavailable for programs as Wait For options and expressions.
3. In the drop-down menus under Communication Interface, select required values.
   Any changes in values are immediately sent to the tool. If any installation values differ from what the tool is using, a warning appears.

5.7.2. Digital Output Mode

Digital Output The tool communication interface allows two digital outputs to be independently configured. In PolyScope, each pin has a drop-down menu that allows the output mode to be set. The following options are available:

- Sinking: This allows the pin to be configured in an NPN or Sinking configuration. When the output is off, the pin allows a current to flow to the ground. This can be used in conjunction with the PWR pin to create a full circuit.
- Sourcing: This allows the pin to be configured in a PNP or Sourcing configuration. When the output is on, the pin provides a positive voltage source (configurable in the IO Tab). This can be used in conjunction with the GND pin to create a full circuit.
- Push / Pull: This allows the pin to be configured in a Push / Pull configuration. When the output is on, the pin provides a positive voltage source (configurable in IO Tab). This can be used in conjunction with the GND pin to create a full circuit When the output is off, the pin allows a current to flow to the ground.

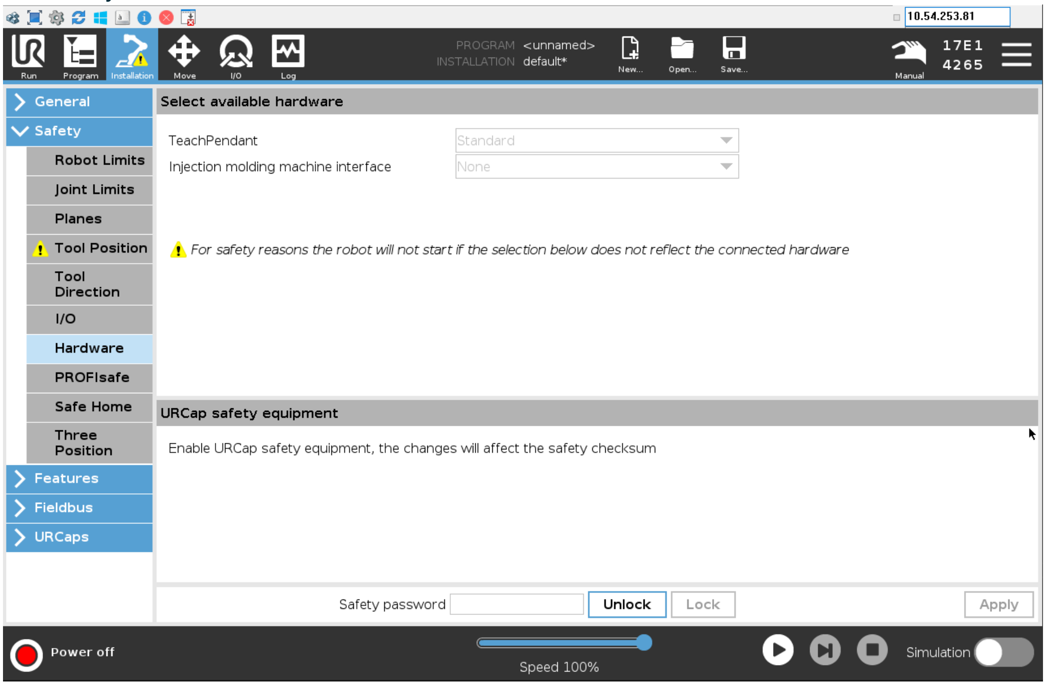
After selecting a new output configuration, the changes take effect. The currently loaded installation is modified to reflect the new configuration. After verifying the tool outputs are working as intended, make sure to save the installation to prevent losing changes.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

5.8. Hardware

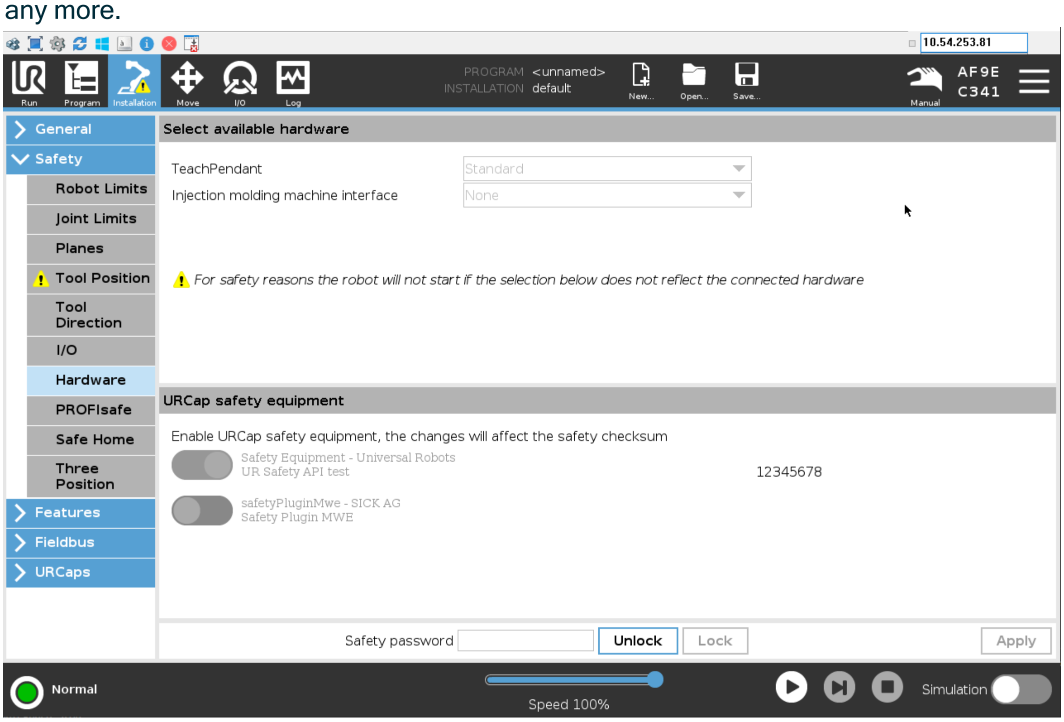
Description

In the hardware tab, you can see and select the hardware that is available to you. Some URCap may be enabled or disabled here. Any URCap safety equipment will affect the safety checksum.



URCap safety equipment

A deactivation of a Safety URCap disconnects that Safety URCap and its related hardware (if given) from all safety and other functional procedures of the robot. Program nodes of a deactivated Safety URCap will remain within the robot program but may not be functional any more.



Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

5.9. Smooth Transition

Description

When switching between safety modes during events (i.e., Reduced Mode Input, Reduced Mode Trigger Planes, Safeguard Stops, and Three-Position Enabling Device), the Robot Arm aims to use 0.4s to create a "soft" transition. Existing applications have unchanged behavior which corresponds to the "hard" setting. New installation files default to the "soft" setting.

To Adjust Transition

1. In the Header, tap **Installation**.
2. In the menu on the left, under Motion, select **Smooth Transition**.
3. Select **Hard** to have a higher acceleration/deceleration or select **Soft** for the smoother default transition setting.

5.10. Motion Version

Description

Selecting Motion Version allows MoveL and MoveJ to have smoother motion during changes in acceleration when the robot is starting and stopping.
This smooth motion results in the following:

- Reduced vibrations on the robot and equipment.
- Automatically clamped move command velocity and acceleration which avoids exceeding robot limits.
- Dynamically reduced blend radii which avoids overlapping blends.

To Adjust Motion Version

1. In the Header, tap **Installation**.
2. In the menu on the left, under Motion, select **Motion Version**.
3. At the bottom of the screen, switch on the **Motion Version 2** button.

5.11. Motion Limits

Description

When selected, Motion Limits restricts payload acceleration, resulting in the following:

- Avoiding items being dropped when using suction type grippers.
- Reduced vibrations when moving heavier items.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

To Limit Payload Acceleration

1. In the Header, tap **Installation**.
2. In the menu on the left, under Motion, select **Motion Limits**.
3. Now you can select:

   Limit acceleration force (F) on payload to set the linear limit.

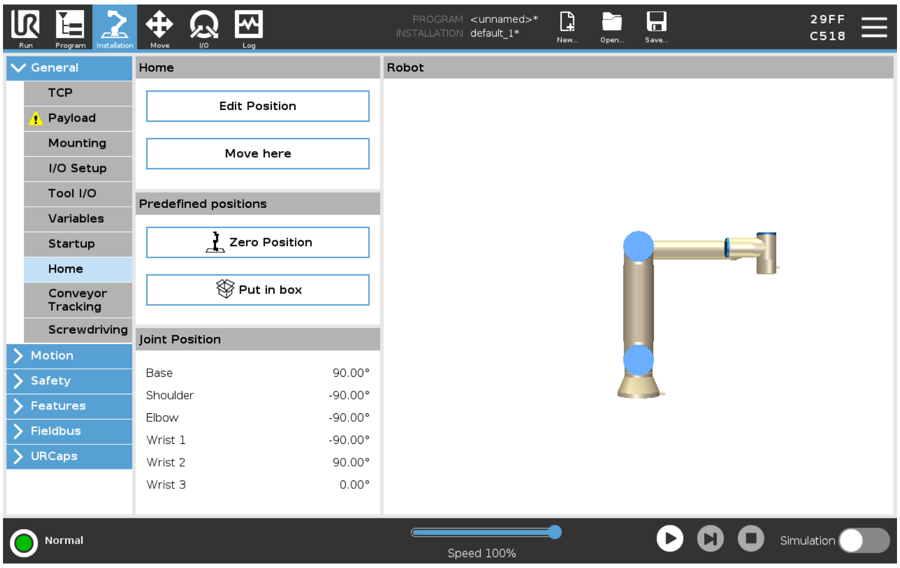
   Limit acceleration moment (M) on payload to set the rotational limit.

5.12. Home

Description

Home is a user-defined return position for the Robot Arm. Once defined, the Home Position is available when creating a robot program. You can use the Home Position to define a Safe Home Position. (See 2.9 Safe Home Position on page 59) Use the Home screen buttons for the following:

- **Edit Position** modifies a Home Position.
- **Move here** moves the Robot Arm to the defined Home Position.
- **Zero Position** returns the Robot Arm to an upright position.



To Define Home

1. In the Header, tap **Installation**.
2. Under **General**, select **Home**.
3. Tap **Set Position**.
4. Teach robot using either **Freedrive** or **Transition buttons**.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

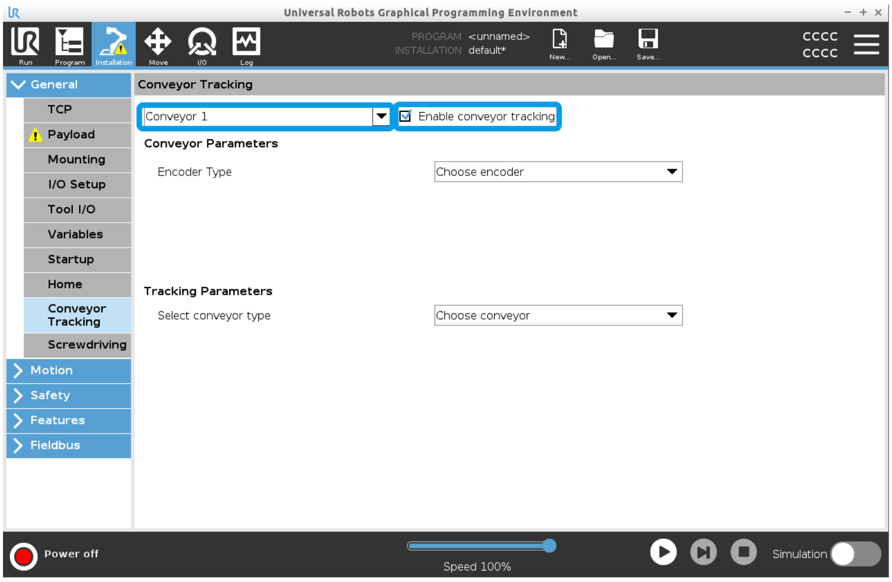
5.13. Conveyor Tracking Setup

Description

The Conveyor Tracking Setup allows the movement of up to two separate conveyors to be configured. The Conveyor Tracking Setup provides options for configuring the robot to work with absolute or incremental encoders, as well as linear or circular conveyors.

Defining a Conveyor

1. In the Header, tap Installation.
2. Under General, select **Conveyor Tracking**.
3. Under Conveyor Tracking Setup, in the dropdown list select **Conveyor 1** or **Conveyor 2**.
   You can only define one conveyor at a time.
4. Select **Enable Conveyor Tracking**
5. Configure **Conveyor Parameters**and **Tracking Parameters**.



Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

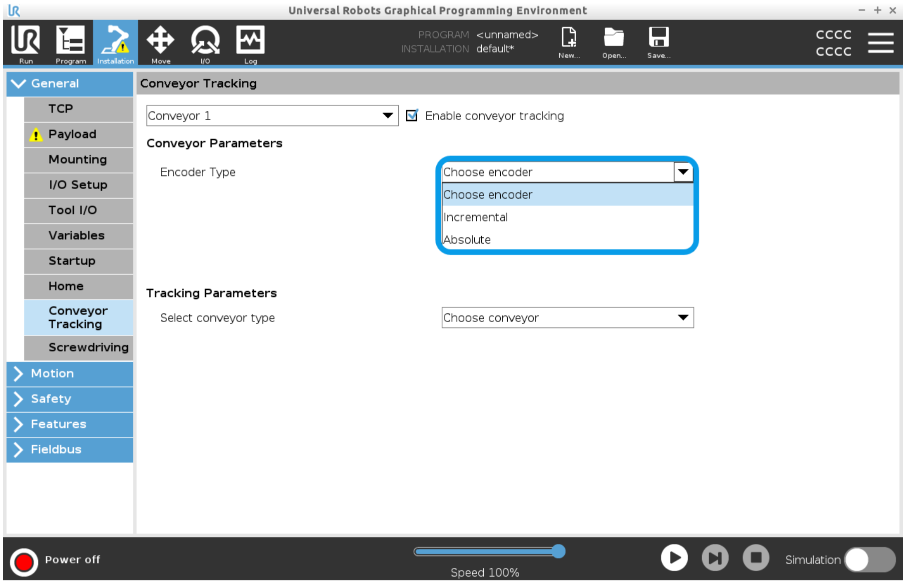
Conveyor Parameters

Incremental Encoders

These can be connected to Digital Inputs 8 to 11. Decoding of digital signals runs at 40 kHz. Using a **Quadrature** encoder (requiring two inputs), the robot can determine the speed and direction of the conveyor. If the direction of the conveyor is constant, a single input can be used to detect *Rising*, *Falling*, or *Rise and Fall* edges which determine conveyor speed.

Absolute Encoders

These can be connected through a MODBUS signal. This requires a Digital MODBUS Output register is preconfigured under the Fieldbus section



Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Tracking Parameters

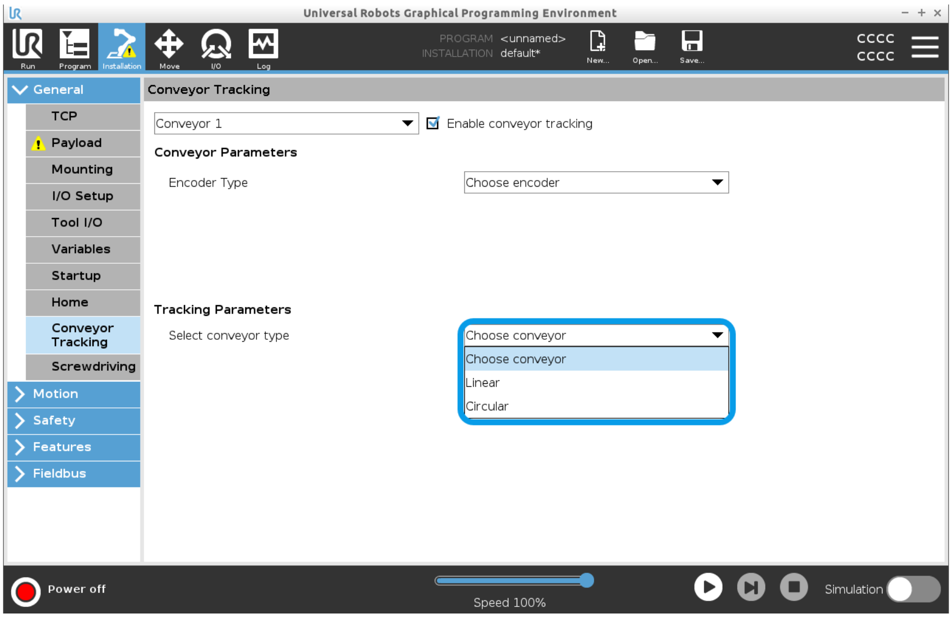
Linear Conveyors

When a linear conveyor is selected, a line feature must be configured in the **Features** part of the installation to determine the direction of the conveyor. Ensure accuracy by placing the line feature parallel to the direction of the conveyor, with a large distance between the two points that define the line feature. Configure the line feature by placing the tool firmly against the side of the conveyor when teaching the two points. If the line feature's direction is opposite to the conveyor's movement, use the **Reverse direction** button. The **Ticks per meter** field displays the number of ticks the encoder generates when the conveyor moves one meter.

Circular Conveyors

When tracking a circular conveyor, the conveyor center point must be defined.

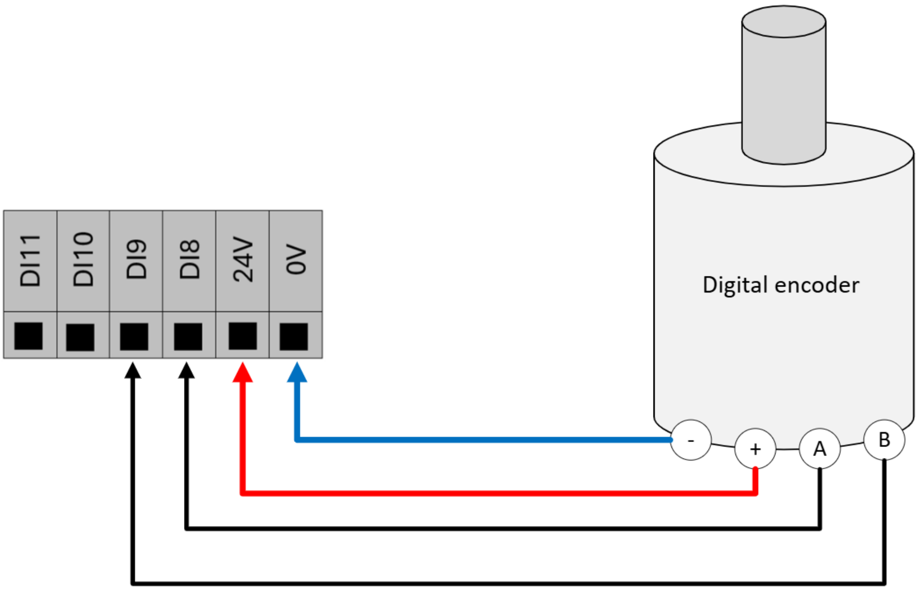
1. Define the center point in the **Features** part of the installation. The value of **Ticks per revolution** must be the number of ticks the encoder generates when the conveyor rotates one full revolution.
2. Select the **Rotate tool with conveyor** checkbox for the tool orientation to track the conveyor rotation.



> **NOTICE**
>
> When using a conveyor, the robot can be configured to track its movement. A program node, Conveyor Tracking, is available for tracking a conveyor. When the Conveyor Tracking defined in the installation is configured correctly, a linear or circular conveyor can be tracked. The node can be added from the Wizard Program node under the Structure tab. While the program is executing under the Conveyor Tracking node, the robot will adjust its movements to follow the conveyor. Other movements are allowed while tracking the conveyor but are relative to the motion of the conveyor belt.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

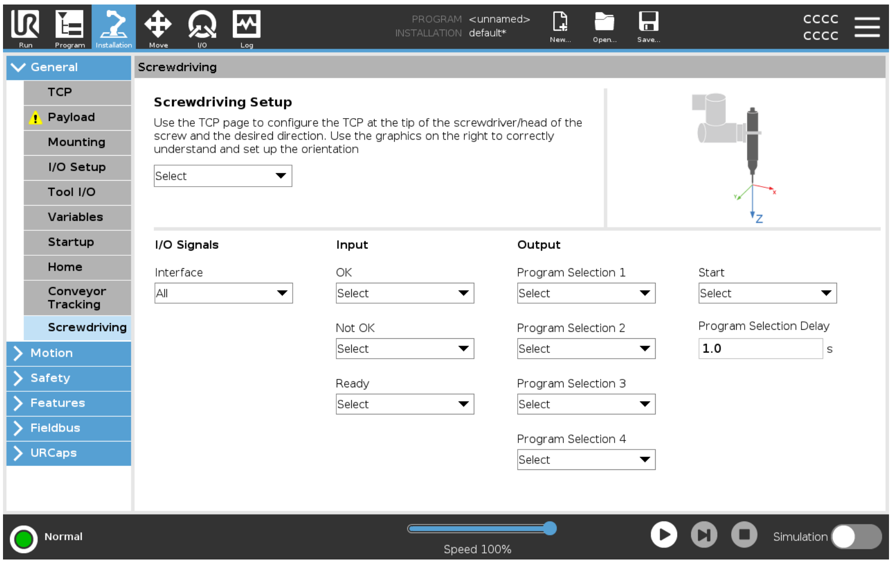
Quadrature Input schematic



5.14. Screwdriving Setup

Description

The Screwdriving Setup provides options for configuring the robot to work with an industrial screwdriver or an industrial nutrunner. You can setup the screwdriver's position with respect to the robot's tool flange and electrical interface.



Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Configuring a Screwdriver

In any output Program Selection list under Output, you can select an integer output to switch Program Selection to a number field.

1. In the Header, tap **Installation**.
2. Under General, select **Screwdriving**, or create your own TCP for screwdriving by tapping **TCP** under General.
3. Under **Input** and **Output**, configure the I/Os for your screwdriver. You can use the **Interface** list to filter the type of I/Os displayed under Input and Output.
4. Under **Start**, select the I/O that starts the screwdriving action.

Configuring the Screwdriver Position

1. Under **Screwdriving Setup**, use the drop-down menu to select a previously defined TCP (see 5.1 TCP Configuration on page 172) where Position and Orientation are set up as follows:
   - Configure the Position to be the tip of the screwdriver tool where it contacts the screw.
   - Configure the Orientation so that the positive Z direction is aligned to the length of the screws to be tightened.

You can visualize the X, Y and Z coordinates of the selected TCP to confirm it matches the tool's bit or socket.
The Screwdriving program node uses the positive Z direction of the selected TCP to follow the screw and calculate distances.

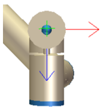
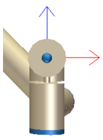
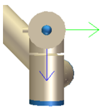
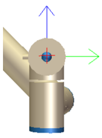
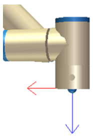
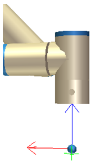
Configuring the Screwdriver Interface

1. Use the **Interface** drop-down menu at the top of the screen to change the displayed content based on signal type.
2. Under **Input**, configure the signals that the robot receives from the screwdriver:
   - OK: High when tightening ends successfully, if not selected this condition is not available in the Screwdriving program node
   - Not OK: High when tightening ends with errors, if not selected this condition is not available in the Screwdriving program node
   - Ready: High when the screwdriver is ready to be started, if not selected this condition is not checked
3. Under **Output** configure the signals that the robot sends to the screwdriver:
   - Start: starts the tool tightening or loosening a screw depending only on wiring.
   - Program Selection: one integer, or up to four binary signals, can be selected to activate different tightening configurations stored in the screwdriver
   - Program Selection Delay: wait time to be used after changing the screwdriver's program to make sure it is active

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Typical Orientation values

Values (in Rotation Vector [rad] notation) are illustrated in the table below.

		Orientation
Screwdriving axis parallel to the negative Y direction of the robot's tool flange1		• **RX**: 1.5708 rad • **RY**: 0.0000 rad • **RZ**: 0.0000 rad
Screwdriving axis parallel to the positive Y direction of the robot's tool flange2		• **RX**: -1.5708 rad • **RY**: 0.0000 rad • **RZ**: 0.0000 rad
Screwdriving axis parallel to the positive X direction of the robot's tool flange3		• **RX**: 0.0000 rad • **RY**: 1.5708 rad • **RZ**: 0.0000 rad
Screwdriving axis parallel to the negative X direction of the robot's tool flange4		• **RX**: 0.0000 rad • **RY**: -1.5708 rad • **RZ**: 0.0000 rad
Screwdriving axis parallel to the positive Z direction of the robot's tool flange5		• **RX**: 0.0000 rad • **RY**: 0.0000 rad • **RZ**: 0.0000 rad
Screwdriving axis parallel to the negative Z direction of the robot's tool flange6		• **RX**: 3.1416 rad • **RY**: 0.0000 rad • **RZ**: 0.0000 rad

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

5.15. Safety

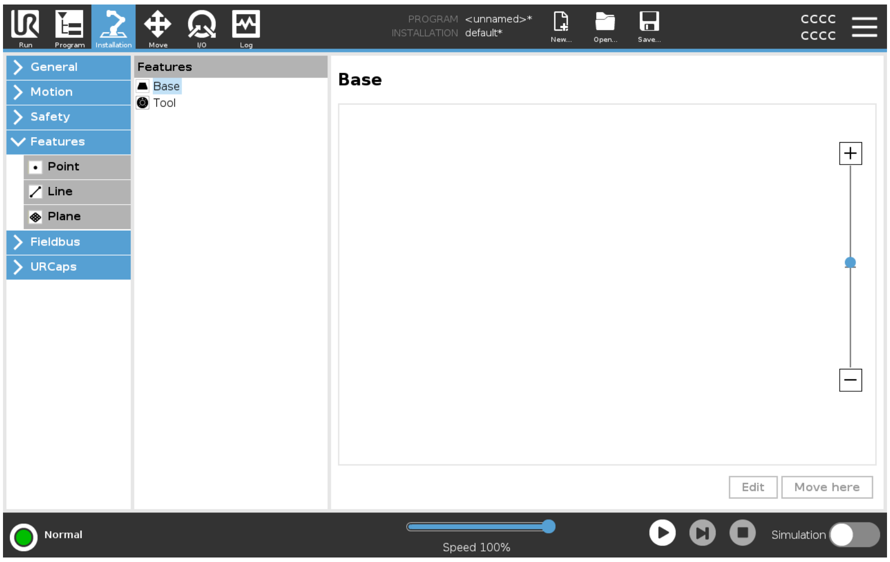
Description

See chapter: 2 Software Safety Configuration on page 32.

5.16. Features

Description

A **Feature** represents an object defined by a six dimensional pose (position and orientation) relative to the robot base. You can name a feature for future reference. Some subparts of a robot program consist of movements executed relative to specific objects other than the base of the robot arm. These objects can be tables, other machines, workpieces, vision systems, blanks, or boundaries existing around the robot arm.

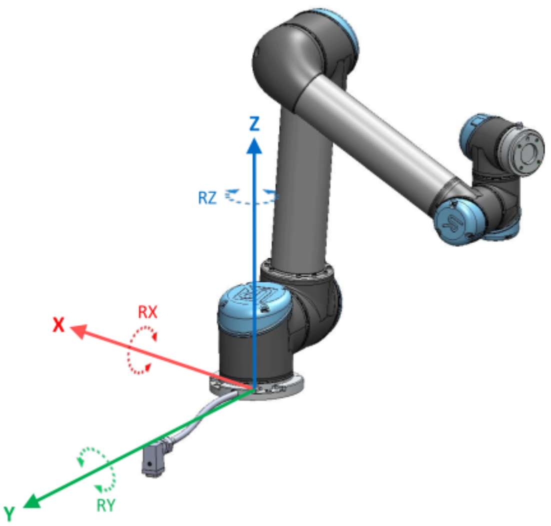


The robot includes two predefined features, listed below, with poses defined by the configuration of the robot arm itself:

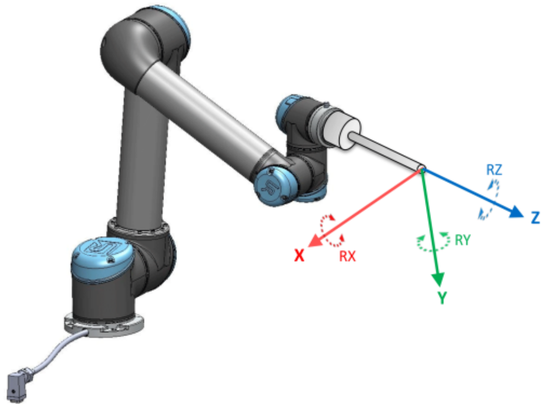
- The Base feature is located with origin in the centre of the robot base.
- The Tool feature is located with origin in the centre of the current TCP.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Base feature



Tool feature



Detail

Use the Point feature, Line feature and/or Plane feature to define a feature pose.
These features are positioned through a method that uses the current pose of the TCP in the work area. So you can teach feature locations using Freedrive , or "jogging" to move the robot to the desired pose.
Selecting a feature depends on the type of object being used and the precision requirements. Use the Line feature and Plane feature where possible as they are based on more input points. More input points mean higher precision.

For example, you can accurately define the direction of a linear conveyor, by defining two points of a Line feature with as much physical separation as possible. You can also use the Point feature to define a linear conveyor, however, you must point the TCP in the direction of the conveyor's movement.
Using more points to define the pose of a table means that the orientation is based on the positions rather than the orientation of a single TCP. A single TCP orientation is harder to configure with high precision.
For more on adding features, see (sections: Adding a Point on page 200) and (Plane Feature on page 201).

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Using a Feature

You can refer to a feature defined the installation from the robot program, to relate robot movements (e.g. **MoveJ**, **MoveL** and **MoveP** commands) to the feature (see section 4.7.1 Move on page 87).
This allows for easy adaptation of a robot program when for example: there are multiple robot stations, when an object is moved during program runtime, or when an object is permanently moved in the scene. Adjusting the feature of an object, adjusts all program movements relative to the object accordingly.
For further examples, see sections: (Example: Manually Updating a Feature to Adjust a Program on page 203) and (Example: Dynamically Updating a Feature Pose on page 204).
When a feature is chosen as a reference, the Move Tool buttons for translation and rotation operate in the selected feature space (see 6 Move Tab on page 214) and (To use the Move Tool arrows on page 214), reading of the TCP coordinates. For example, if a table is defined as a feature and is chosen as a reference in the Move Tab, the translation arrows (i.e., up/down, left/right, forward/backward) move the robot in these directions relative to the table. Additionally, the TCP coordinates will be in the frame of the table.

- In the Features tree you can rename a Point, Line or Plane by tapping the pencil button.
- In the Features tree you can delete a Point, Line or Plane by tapping the Delete button.

Using Move here

Tap **Move here** to move the robot arm towards the selected feature. At the end of this movement, the coordinate systems of the feature and the TCP will coincide.
Move here is disabled if the robot arm cannot reach the feature.

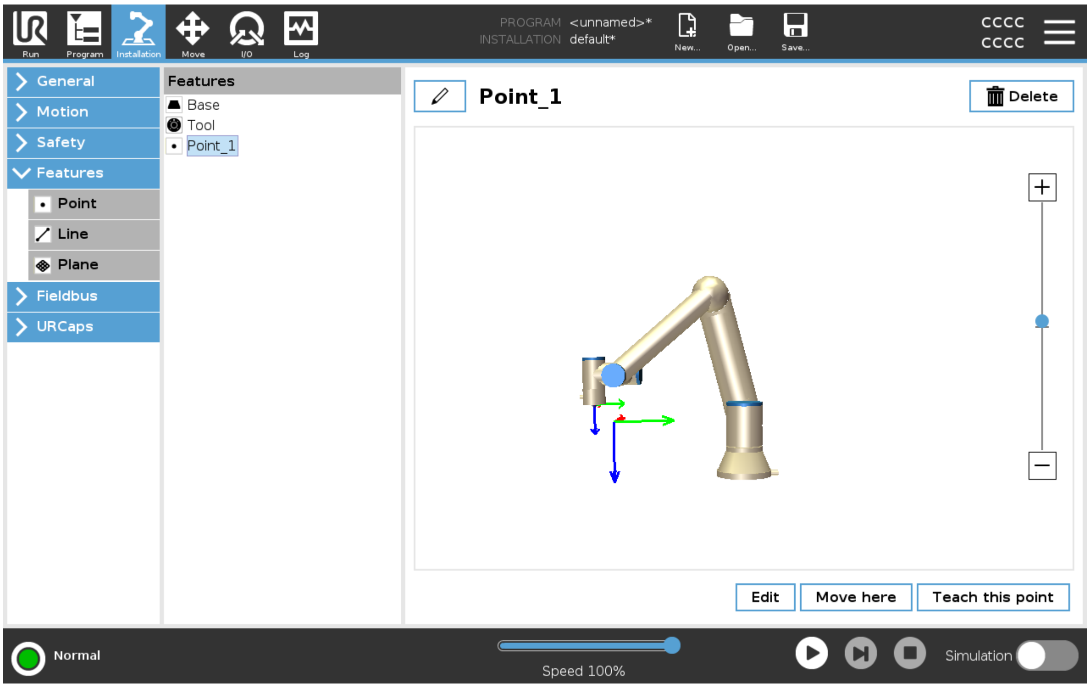
Point feature

The point feature defines a safety boundary or a global home configuration of the Robot arm. The point feature pose is defined as the position and orientation of the TCP.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Adding a Point

1. In Installation, select **Features**.
2. Under Features select **Point**.

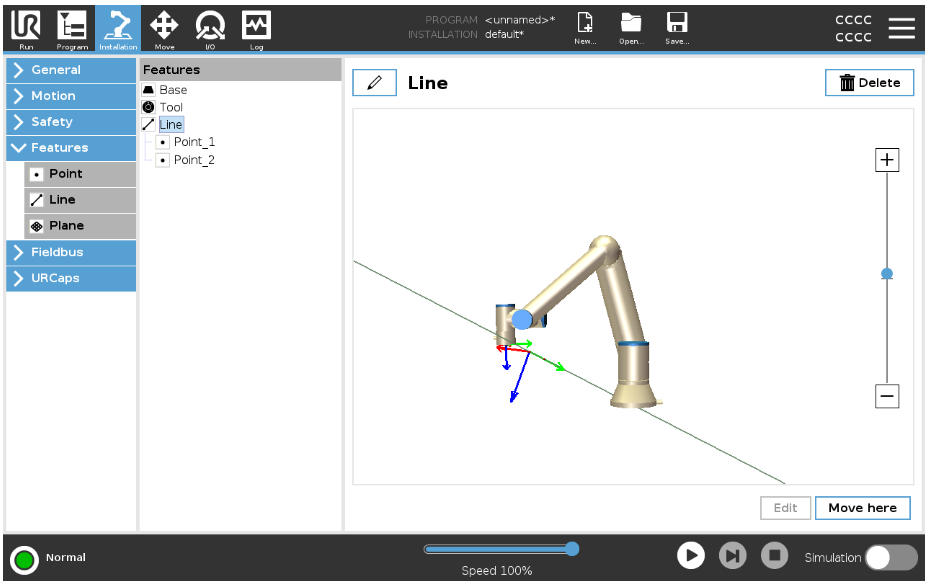


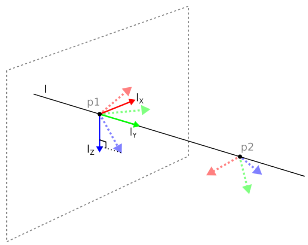
Line feature The line feature defines lines that the robot needs to follow. (e.g., when using conveyor tracking). A line *l* is defined as an axis between two point features *p1* and *p2* as shown in figure 5.16.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Adding a Line

1. In Installation, select **Features**.
2. Under Features select **Line**.





Here you can see the axis directed from the first point towards the second point, constitutes the y-axis of the line coordinate system. The z-axis is defined by the projection of the z-axis of *p1* onto the plane perpendicular to the line. The position of the line coordinate system is the same as the position of *p1*.

Plane Feature

Select the plane feature when you need a frame with high precision: e.g., when working with a vision system or doing movements relative to a table.

Adding a plane

1. In Installation, select **Features**.
2. Under Features select **Plane**.

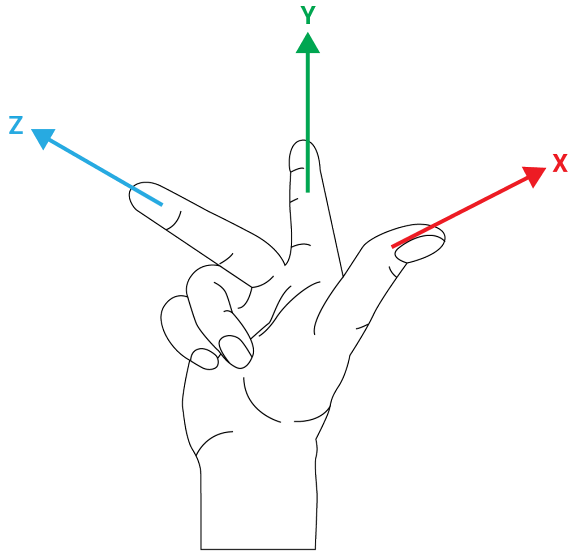
Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Teaching a plane

When you press the plane button to create a new plane, the on-screen guide assists you creating a plane.

1. Select Origo
2. Move robot to define the direction of the positive x-axis of the plane
3. Move robot to define the direction of the positive y-axis of the plane

The plane is defined using the right hand rule so the z- axis is the cross product of the x-axis and the y-axis, as illustrated below.



NOTICE

You can re-teach the plane in the opposite direction of the x-axis, if you want that plane to be normal in the opposite direction.

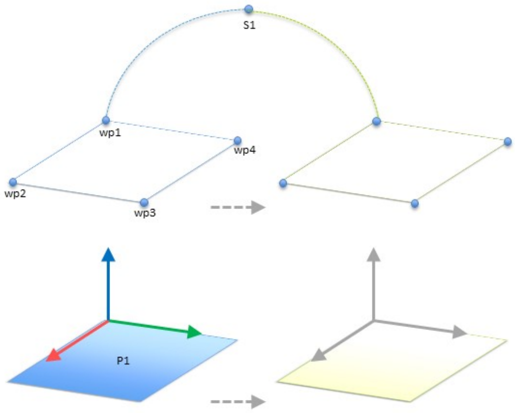
Modify an existing plane by selecting Plane and pressing Modify Plane. You will then use the same guide as for teaching a new plane.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Example: Manually Updating a Feature to Adjust a Program

Consider an application where multiple parts of a robot program is relative to a table. The figure below illustrates the movement through waypoints from `wp1` to `wp4`.

```
Robot Program
    MoveJ
        S1
    MoveL # Feature: P1_var
        wp1
        wp2
        wp3
        wp4
```



14.1: *Simple program with four waypoints relative to a feature plane manually updated by changing the feature*

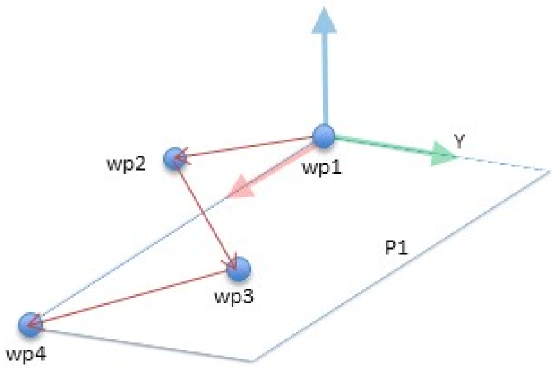
14.2:

The application requires the program to be reused for multiple robot installations where the position of the table varies slightly. The movement relative to the table is identical. By defining the table position as a feature *P1* in the installation, the program with a *MoveL* command configured relative to the plane can be easily applied on additional robots by just updating the installation with the actual position of the table.
The concept applies to a number of Features in an application to achieve a flexible program can solve the same task on many robots even though if other places in the work space varies between installations.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

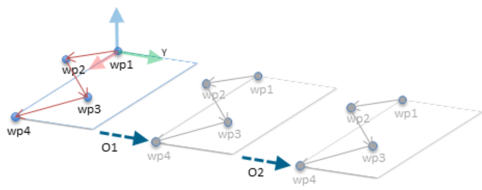
Example: Dynamically Updating a Feature Pose

Consider a similar application where the robot must move in a specific pattern on top of a table to solve a particular task.



A *MoveL* command with four waypoints relative to a plane feature

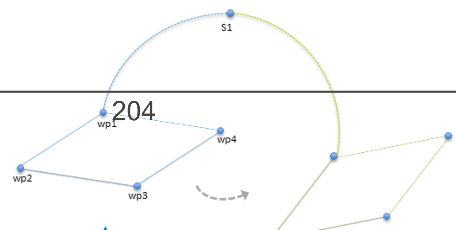
```
Robot Program
    MoveJ
        wp1
    y = 0.01
    o = p[0,y,0,0,0,0]
    P1_var = pose_trans(P1_var, o)
    MoveL # Feature: P1_var
        wp1
        wp2
        wp3
        wp4
```



14.3: *Applying an offset to the plane feature*

14.4:

```
Robot Program
    MoveJ
        S1
    if (digital_input[0]) then
        P1_var = P1
    else
        P1_var = P2
    MoveL # Feature: P1_var
        wp1
        wp2
        wp3
        wp4
```

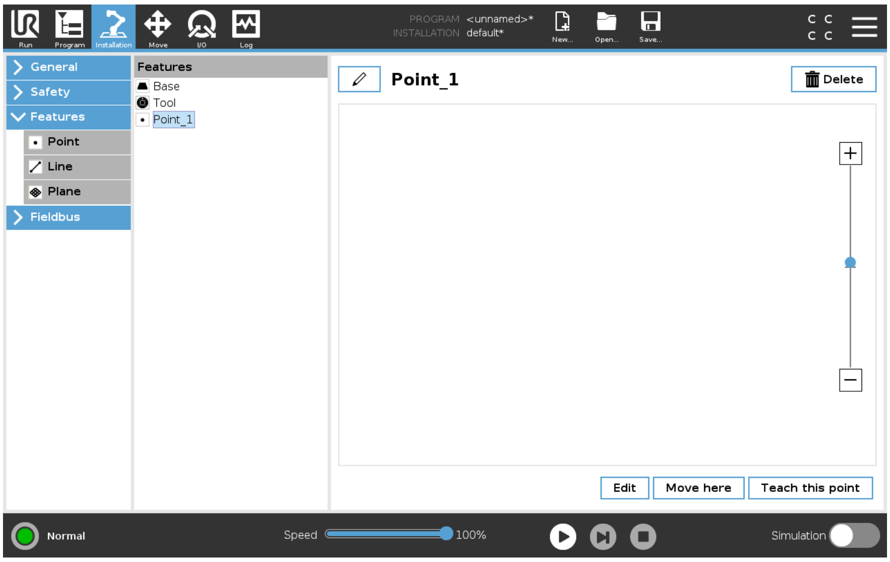


Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

5.16.1. Feature Edit

Description

The Feature Edit is an alternative way to add features to your installation and/or edit existing features.

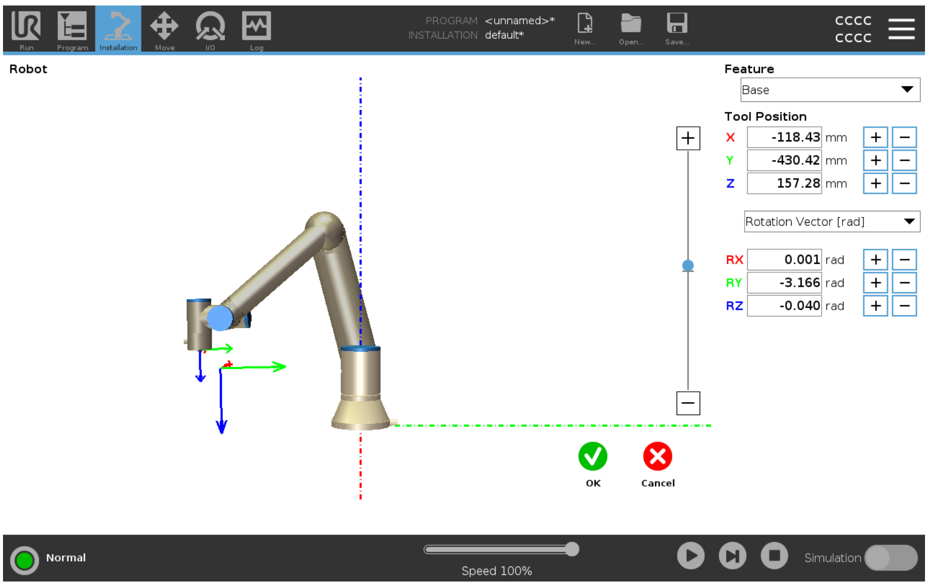


Use Edit to place and move features without moving the robot arm, so the feature can be placed outside of the robot arm's reach.

Editing a Point

You can edit a defined point or an undefined point. Editing an undefined point defines it.

1. In Installation, tap **Features**.
2. Under Features, select **Point** to add a point to your program tree.
3. Tap **Edit** to access the Edit screen to make changes to the position and rotation of the point



Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Editing a Line

The line appears as two points in your program tree. You must define each point.

1. In Installation, tap **Features**.
2. Under Feature, select **Line** to add a line to your program tree.
3. The line is made up of two points:
    - Tap one point to edit those coordinates, then tap the other line point to edit those coordinates.

Editing a Plane

1. In Installation, tap **Features**.
2. Under Feature, select **Plane** to add a plane to your program tree.
3. Tap **Edit** to access the Edit screen to make changes to the positon and rotation of the plane

5.17. Fieldbus

Fieldbus

You can use the Fieldbus options to define and configure the family of industrial computer network protocols used for real-time distributed control accepted by PolyScope:

- MODBUS
- Ethernet/IP
- PROFINET
- PROFIsafe
- UR Connect

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

5.17.1. MODBUS Client I/O Setup

5.17.2. EtherNet/IP Adapter

Description

EtherNet/IP is a network protocol that enables the connection of the robot to an industrial EtherNet/IP Scanner Device.
If the connection is enabled, you can select the action that occurs when a program loses EtherNet/IP Scanner Device connection.
Those actions are:

None	PolyScope ignores the loss of EtherNet/IP connection and the program continues to run.
Pause	PolyScope pauses the current program. The program resumes from where it stopped.
Stop	PolyScope stops the current program.

5.17.3. PROFINET Device

Description

The PROFINET network protocol enables or disables the connection of the robot to an industrial PROFINET IO-Controller.
If the connection is enabled, you can select the action that occurs when a program loses PROFINET IO-Controller connection.
Those actions are:

None	PolyScope ignores the loss of PROFINET connection and the program continues to run.
Pause	PolyScope pauses the current program. The program resumes from where it stopped.
Stop	PolyScope stops the current program.

If the PROFINET engineering tool (e.g. TIA portal) emits a DCP Flash signal to the robot's PROFINET or PROFIsafe device, a popup in PolyScope is displayed.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

5.17.4. PROFIsafe

Description

The PROFIsafe network protocol (implemented as version 2.6.1) allows the robot to communicate with a safety PLC according to ISO 13849, Cat 3 PLd requirements. The robot transmits safety state information to a safety PLC and receives commands to reduce safety limits or trigger a safety-related function such as an emergency stop.
The PROFIsafe interface provides a safe, network-based alternative to connecting wires to the safety IO pins of the robot control box.
PROFIsafe is only available on robots that have an enabling license, which you can obtain by contacting your local sales representative. After you obtain the license, download it from myUR.
For information about robot registration and license activation, see Robot Registration and License File on page 241.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Advanced Options

A control message that the safety PLC sends to the robot contains the information shown in the following table.

Signal	Description
E-Stop by system	0: Asserts the system e-stop. 1: Clear system e-stop.
Safeguard stop	0: Asserts the safeguard stop. 1: Normal operation state. **Note**: Also refer to the “Reset safeguard stop” signal description.
Reset safeguard stop	Resets the safeguard stop state on a 0-to-1 transition when the “safeguard stop” signal is already set to 1.
Safeguard stop auto	0: Asserts safeguard stop if the robot is operating in Automatic mode. 1: Normal operation state. Safeguard stop auto shall only be used when a 3-Position Enabling (3PE) Device is configured. If no 3PE Device is configured, the safeguard stop auto acts as a normal safeguard stop input. **Note**: Also refer to the “Reset safeguard stop auto” signal description.
Reset safeguard stop auto	Resets the safeguard stop auto state on a 0-to-1 transition when the “safeguard stop auto” signal is already set to 1.
Reduced	0: Activates the Reduced safety limits. 1: Activates the “Normal mode” safety limits. The safety system guarantees the robot is within reduced limits less than 0.5s after the input is activated. If the robot arm continues to violate any of the reduced limits, a Stop Category 0 is triggered.
Operational mode	0: Activates the manual operational mode. 1: Activates the automatic operational mode. If the safety configuration "Operational mode selection via PROFIsafe" is disabled, this field shall be omitted from the PROFIsafe control message.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Advanced Options

A status message that the robot sends to the safety PLC contains the information shown in the following table.

Signal	Description
Stop, cat. 0	0: Robot is performing, or has completed, a safety stop of category 0; a hard stop by immediate removal of power to the arm and the motors. 1: Normal operation state.
Stop, cat. 1	0: Robot is performing, or has completed, a safety stop of category 1; a controlled stop after which the motors are left in a power-off state with brakes engaged. 1: Normal operation state.
Stop, cat. 2	0: Robot is performing, or has completed, a safety stop of category 2; a controlled stop after which the motors are left in a power-on state. 1: Normal operation state.
Violation	0: Robot is stopped because the safety system has failed to comply with the active safety limits defined. 1: Normal operation state.
Fault	0: Robot is stopped because of an unexpected exceptional error in the safety system. 1: Robot is not experiencing an unexpected exceptional error in the safety system.
E-stop by system	0: Robot is stopped because of one of the following conditions: • A safety PLC connected via PROFIsafe has asserted a system level e-stop. • An IMMI module connected to the control box has asserted a system level e-stop. • A unit connected to the system e-stop configurable safety input of the control box has asserted a system level e-stop. 1: Robot is not in system e-stop.
E-stop by robot	0: The robot is stopped because of one of the following conditions: • The e-stop button of the teach pendant is pressed. • An e-stop button connected to the robot e-stop non-configurable safety input of the control box is pressed. 1: Robot is not in e-stop by robot.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Advanced Options

Signal	Description
Active limit set	The active set of safety limits. 0: Normal 1: Reduced 2: Recovery
Robot moving	0: Robot is moving. If any joint moves at a velocity of 0.02 rad/s or higher, the robot is considered in motion. 1: Robot is at standstill.
Safe home position	0: Robot is at rest (robot not moving), and in the position defined as the Safe Home Position. 1: Robot is not at rest, or not in the position defined as the Safe Home Position.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Advanced Options

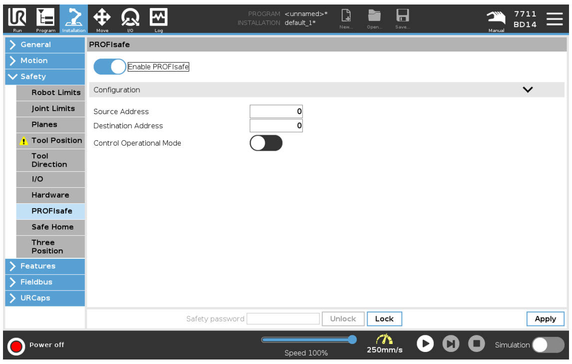
Signal	Description
Safeguard stop	0: The robot is stopped due to one of the following conditions: • A safety PLC connected via PROFIsafe has asserted the safeguard stop. • A unit connected to the safeguard stop nonconfigurable input of the control box has asserted the safeguard stop. • A unit connected to the safeguard stop configurable safety input of the control box has asserted the safeguard stop. 1: The robot is not stopped due to a safeguard stop. **Note**: Also refer to the “Reset safeguard stop” signal description. PROFIsafe enforces the use of the safeguard reset functionality.
Safeguard stop auto	0: The robot is stopped because it is operating in Automatic mode and one of the following conditions applies: • A safety PLC connected via PROFIsafe has asserted safeguard stop auto. • A unit connected to a safeguard stop auto configurable safety input of the control box has asserted safeguard stop auto. 1: The robot is not stopped due to safeguard stop auto. **Note**: Also refer to the “Reset safeguard stop auto” signal description. PROFIsafe enforces the use of the safeguard reset functionality.
3PE stop	0: The robot is stopped because it is operating in Manual mode and one of the following conditions applies: • Any 3PE is pressed to the middle position, and Freedrive input is active. • Not all 3PE devices are pressed to the middle position. 1: Robot is not stopped because of a 3-position enabling device.
Operational mode	Indication of the active operational mode of the robot. 0: Disabled 1: Automatic 2: Manual
Reduced	0: Reduced safety limits are active. 1: Normal safety limits are active.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Configuring PROFIsafe

Configuring PROFIsafe relates to programming the safety PLC, but requires minimal robot setup.

1. Connect the robot to a trusted network that accesses a safety compliant PLC.
2. On PolyScope, in the Header, tap **Installation**.
3. Tap Safety, select **PROFIsafe** and configure as needed.



Enabling PROFIsafe

1. Enter the robot safety password and tap **Unlock**.
2. Use the switch button to enable PROFIsafe.
3. Enter a source address and destination address into the corresponding boxes.

   These addresses are arbitrary numbers used by the robot and the safety PLC to identify each other.
4. You can switch the Control Operational Mode to the ON position if you want PROFIsafe to control the robot operational mode.

   Only one source can control the operational mode of the robot. Therefore other sources of mode selection are disabled when operational mode selection via PROFIsafe is enabled.

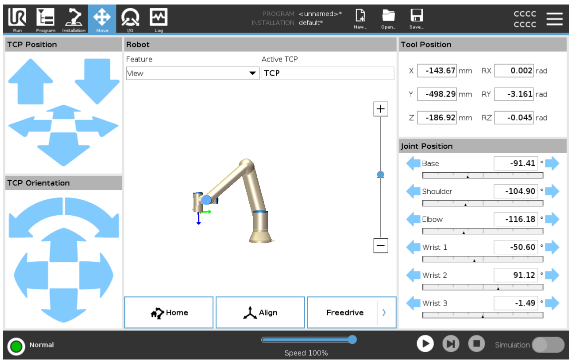
The robot is now setup to communicate with a safety PLC.
You cannot release the robot's brakes if the PLC is not responding or if it is misconfigured.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

6. Move Tab

Description

Use the Move Tab screen to move (jog) the robot arm directly, either by translating/rotating the robot tool, or by moving robot joints individually.



To use the Move Tool arrows

Hold down any of the **Move Tool** arrows to move the robot arm in the corresponding direction.

- The **Translate arrows** (upper) move the tool flange in the direction indicated.
- The **Rotate arrows** (lower) change the orientation of the tool in the indicated direction. The rotation point is the Tool Center Point (TCP), i.e.the point at the end of the robot arm that gives a characteristic point on the tool. The TCP is shown as a small blue ball.

Robot

If the current position of the TCP approaches a safety plane, a trigger plane, or the orientation of robot tool is near the tool orientation boundary limit , a 3D representation of the proximate boundary limit is shown. The visualization of boundary limits is disabled during program execution.

Safety planes display in yellow and black with an arrow indicating which side of the plane, the robot TCP is allowed to be positioned.

Trigger planes display in blue and green with an arrow indicating the side of the plane, where the **Normal** mode limits are active.

The tool orientation boundary limit is visualized with a spherical cone together with a vector indicating the current orientation of the robot tool. The inside of the cone represents the allowed area for the tool orientation (vector).
When the robot TCP is no longer in proximity of the limit, the 3D representation disappears. If the TCP is in violation or very close to violating a boundary limit, the visualization of the limit turns red.

Feature

Under **Feature**, you can define how to control the robot arm relative to **View**, **Base** or **Tool** features. For the best feel for controlling the robot arm you can select the **View** feature, then use **Rotate arrows** to change the viewing angle of the 3D image to match your view of the real robot arm.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Active TCP In the **Robot** field, under **Active TCP**, the name of the current active Tool Center Point (TCP) is displayed.

Home The **Home** button accesses the **Move Robot into Position** screen, where you can hold down the **Auto** button to move robot into position previously defined under Installation. The Home button's default setting returns the Robo Arm to an upright position .

Freedrive The on-screen **Freedrive** button allows the Robot Arm to be pulled into desired positions/poses.

Align The **Align** button allows the Z axis of the active TCP to align to a selected feature.

Tool Position The text boxes display the full coordinate values of the TCP relative to the selected feature. You can configure several named TCPs. You can also tap **Edit pose** to access the **Pose Editor** screen.

Joint Position The **Joint Position** field allows you to directly control individual joints. Each joint moves along a default joint limit range from $-360°$ to $+360°$, defined by a horizontal bar. Once the limit is reached you cannot move a joint any further. You can configure joints with a position range different from the default, this new range is indicated with red zone inside the horizontal bar.

Using Freedrive in the Move tab The **Freedrive** button shall only be used in applications if allowed by the risk assessment.

> **WARNING**
>
> Failure to correctly configure the mounting setting can result in unwanted robot arm movement when you use the **Freedrive** button.
>
> - Payload settings and robot mounting settings shall be set correctly before using Freedrive.
> - All personnel shall remain outside the reach of the robot arm, when **Freedrive** is in use.

> **WARNING**
>
> Failure to correctly configure the installation settings, can increase the risk of the robot arm falling during **Freedrive**, due to payload errors.
>
> - Verify the installation settings are correct (e.g. Robot mounting angle, payload mass and payload center of gravity offset) . Save and load the installation files along with the program.
> - Save and load the installation files along with the program.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

6.1. Pose Editor Screen

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

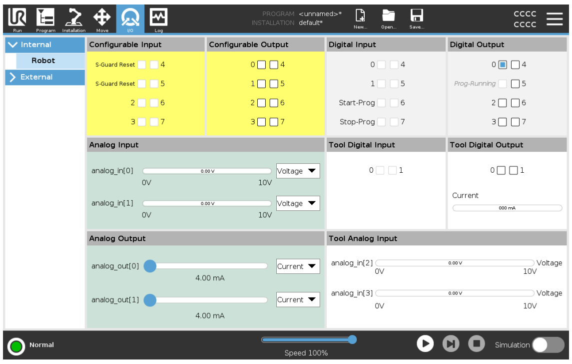
7. I/O Tab

Description

Use the I/O Tab screen to monitor and set the live I/O signals from/to the Control Box.

The screen displays the current state of the I/O, including during program execution. The program stops if anything is changed during execution. At program stop, all output signals retain their states. The screen updates at 10Hz, so a very fast signal might not display properly.

Configurable I/Os can be reserved for special safety settings defined in the safety I/O configuration section of the installation (see I/O); those which are reserved will have the name of the safety function in place of the default or user defined name.
Configurable outputs that are reserved for safety settings are not togglable and will be displayed as LED's only.



Voltage

When the Tool Output is controlled by the user, you can configure Voltage. Selecting a URCap removes access to Voltage.

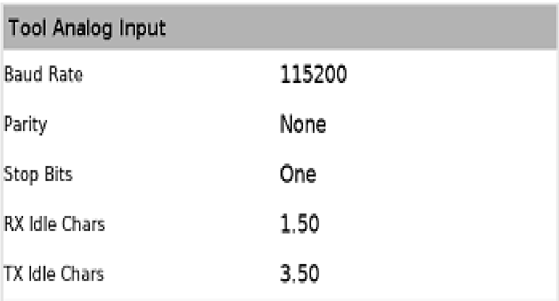
Analog Domain Settings

The analog I/O's can be set to either current [4-20mA] or voltage [0-10V] output. These settings are persistent over restarts of the robot controller and saved in the installation. Control over the tool I/Os could be assigned to a URCap in **Tool I/O** of the **Installation** tab. Selecting a URCap removes user's control over tool's analog I/O.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Tool Communication Interface

When the **Tool Communication Interface TCI** is enabled, the tool analog input becomes unavailable. On the **I/O** screen, the **Tool Input** field appears as shown.

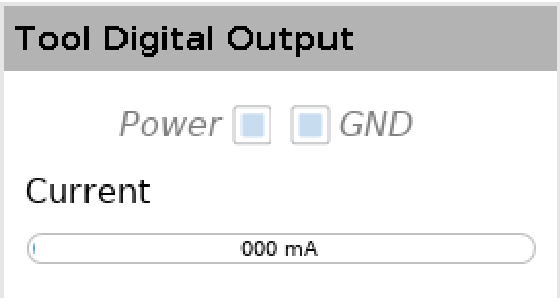


Tool Analog Input	
Baud Rate	115200
Parity	None
Stop Bits	One
RX Idle Chars	1.50
TX Idle Chars	3.50

Dual Pin power

Dual Pin Power is used as a source of power for the tool. Enabling Dual Pin Power disables the default tool digital outputs. When Dual Pin Power is enabled, the tool digital outputs shall be named as follows:

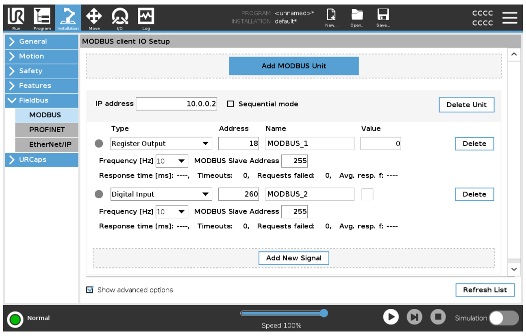
- tool_out[0] (Power)
- tool_out[1] (GND)



Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

7.1. MODBUS

Description Here, the MODBUS client (master) signals can be set up. Connections to MODBUS servers (or slaves) on specified IP addresses can be created with input/output signals (registers or digital). Each signal has a unique name so it can be used in programs.



Refresh Push this button to refresh all MODBUS connections. Refreshing disconnects all modbus units, and connects them back again. All statistics are cleared.

Add unit Push this button to add a new MODBUS unit.

Delete unit Push this button to delete the MODBUS unit and all signals on that unit.

Set unit IP Here the IP address of the MODBUS unit is shown. Press the button to change it.

Sequential mode *Available only when Show Advanced Options selected.* Selecting this checkbox forces the modbus client to wait for a response before sending the next request. This mode is required by some fieldbus units. Turning this option on may help when there are multiple signals, and increasing request frequency results in signal disconnects.
The actual signal frequency may be lower than requested when multiple signals are defined in sequential mode. Actual signal frequency can be observed in signal statistics. The signal indicator turns yellow if the actual signal frequency is less than half of the value selected from the **Frequency** drop-down list.

Add signal Push this button to add a signal to the corresponding MODBUS unit.

Delete signal Push this button to delete a MODBUS signal from the corresponding MODBUS unit.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Set signal type Use this drop down menu to choose the signal type.
Available types are:

Digital input	A digital input (coil) is a one-bit quantity which is read from the MODBUS unit on the coil specified in the address field of the signal. Function code 0x02 (Read Discrete Inputs) is used.
Digital output	A digital output (coil) is a one-bit quantity which can be set to either high or low. Before the value of this output has been set by the user, the value is read from the remote MODBUS unit. This means that function code 0x01 (Read Coils) is used. When the output has been set by a robot program or by pressing the **set signal value** button, the function code 0x05 (Write Single Coil) is used onwards.
Register input	A register input is a 16-bit quantity read from the address specified in the address field. The function code 0x04 (Read Input Registers) is used.
Register output	A register output is a 16-bit quantity which can be set by the user. Before the value of the register has been set, the value of it is read from the remote MODBUS unit. This means that function code 0x03 (Read Holding Registers) is used. When the output has been set by a robot program or by specifying a signal value in the **set signal value** field, function code 0x06 (Write Single Register) is used to set the value on the remote MODBUS unit.

Set signal address This field shows the address on the remote MODBUS server. Use the on-screen keypad to choose a different address. Valid addresses depends on the manufacturer and configuration of the remote MODBUS unit.

Set signal name Using the on-screen keyboard, the user can give the signal a name. This name is used when the signal is used in programs.

Signal value Here, the current value of the signal is shown. For register signals, the value is expressed as an unsigned integer. For output signals, the desired signal value can be set using the button. Again, for a register output, the value to write to the unit must be supplied as an unsigned integer.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Signal connectivity status

This icon shows whether the signal can be properly read/written (green), or if the unit responds unexpected or is not reachable (gray). If a MODBUS exception response is received, the response code is displayed. The MODBUS-TCP Exception responses are:

E1	ILLEGAL FUNCTION (0x01) The function code received in the query is not an allowable action for the server (or slave).
E2	ILLEGAL DATA ADDRESS (0x02) The function code received in the query is not an allowable action for the server (or slave), check that the entered signal address corresponds to the setup of the remote MODBUS server.
E3	ILLEGAL DATA VALUE (0x03) A value contained in the query data field is not an allowable value for server (or slave), check that the entered signal value is valid for the specified address on the remote MODBUS server.
E4	SLAVE DEVICE FAILURE (0x04) An unrecoverable error occurred while the server (or slave) was attempting to perform the requested action.
E5	ACKNOWLEDGE (0x05) Specialized use in conjunction with programming commands sent to the remote MODBUS unit.
E6	SLAVE DEVICE BUSY (0x06) Specialized use in conjunction with programming commands sent to the remote MODBUS unit, the slave (server) is not able to respond now.

Show Advanced Options

This check box shows/hides the advanced options for each signal.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Advanced Options

Update Frequency	This menu can be used to change the update frequency of the signal. This means the frequency with which requests are sent to the remote MODBUS unit for either reading or writing the signal value. When the frequency is set to 0, then modbus requests are initiated on demand using a *modbus_get_signal_status*, *modbus_set_output_register*, and *modbus_set_output_signal* script functions.
Slave Address	This text field can be used to set a specific slave address for the requests corresponding to a specific signal. The value must be in the range 0-255 both included, and the default is 255. If you change this value, it is recommended to consult the manual of the remote MODBUS device to verify its functionality when changing slave address.
Reconnect count	Number of times TCP connection was closed, and connected again.
Connection status	TCP connection status.
Response time [ms]	Time between modbus request sent, and response received - this is updated only when communication is active.
Modbus packet errors	Number of received packets that contained errors (i.e. invalid length, missing data, TCP socket error).
Timeouts	Number of modbus requests that didn't get response.
Requests failed	Number of packets that could not be sent due to invalid socket status.
Actual freq.	The average frequency of client (master) signal status updates. This value is recalculated each time the signal receives a response from the server (or slave).

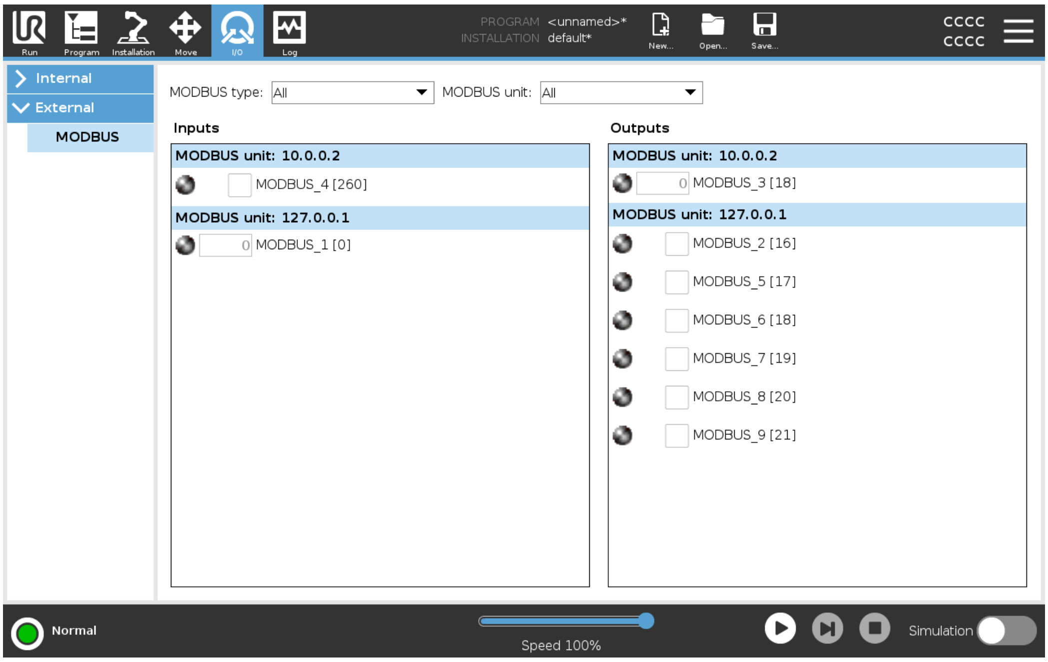
All counters count up to 65535, and then wrap back to 0.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

7.1.1. MODBUS Client I/O

Description

The MODBUS Client I/O signals as they are set up in the installation. Using the drop-down menus at the top of the screen, you can change the displayed content based on signal type and MODBUS unit if more than one is configured.
Each signal in the lists contains its connections status, value, name, and signal address. The output signals can be toggled if the connection status and the choice for I/O tab control allows it.

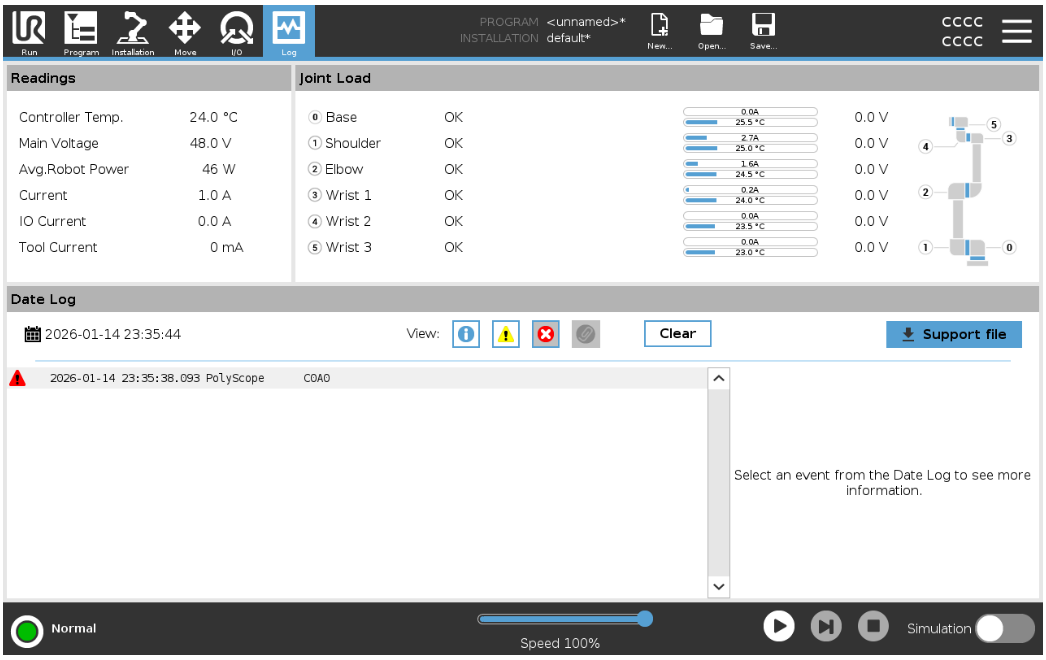


Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

8. Log Tab

Description

The **Log** tab displays information about the robot arm and Control Box.



Readings and Joint Load

The Readings pane displays Control Box information. The Joint Load pane displays information for each robot arm joint.
Each joint displays:

- Temperature
- Load
- Status
- Voltage

Date Log

The first column displays log entries, categorized by the severity. The second column shows a paperclip if there is an Error Report associated with the log entry. The next two columns display the messages' time of arrival and the source of the message. The last column shows a short description of the message itself.
Some log messages are designed to provide more information that is displayed on the right side, after selecting the log entry.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Message Severity

You can filter messages by selecting the toggle buttons that correspond to the severity of the log entry or by whether an attachment is present. The following table describes message severity.

	Provides general information, such as status of a program, changes of the controller and controller version.
	Issues that may have occurred but the system was able to recover.
	A violation occurs if the safety limit is exceeded. This causes the robot to perform a safety rated stop.
	A fault occurs if there is an unrecoverable error in the system. This causes the robot to perform a safety rated stop.

When you select a log entry, additional information appears on the right side of the screen. Selecting the attachments filter either displays entry attachments exclusively or, displays all entries.

Saving Error Reports

A detailed status report is available when a paper clip icon appears on the log line.

NOTICE

The oldest report is deleted when a new one is generated. Only the five most recent reports are stored.

1. Select a log line and tap the Save Report button to save the report to a USB drive.

   You can save the report while a program is running.

You can track and export the following list of errors:

- Emergency stop
- Fault
- Internal PolyScope exceptions
- [1]Robot Stop
- Unhandled exception in URCap
- Violation

The exported report contains: a user program, a history log, an installation and a list of running services.

[1]Robot stop was previously known as "Protective Stop" for Universal Robots robots.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Technical Support File

The report file contains information that is helpful to diagnose and reproduce issues. The file contains records of previous robot failures, as well as current robot configurations, programs and installations. The report file can be saved to external USB drive. On the Log screen, tap **Support file** and follow the on-screen instructions to access the function.

NOTICE

The export process can take up to 10 minutes depending on USB drive speed and the size of files collected from robot file system. The report is saved as a regular zip file, that is not password protected, and can be edited before sending to technical support.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

9. Program and Installation Manager

Description

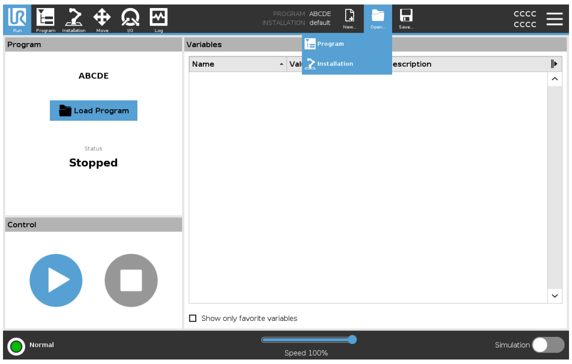
The Program and Installation Manager refers to three icons that allow you to create, load and configure Programs and Installations:

- **New...** Allows you to create a new Program and/or Installation.
- **Open...** Allows you to load a Program and/or Installation.
- **Save...** Offers saving options for a Program and/or Installation.

The File Path displays your current loaded Program name and the type of Installation. File Path changes when you create or load a new Program or Installation.
You can have several installation files for a robot. Programs created load and use the active installation automatically.

To load a program

1. In the Program and Installation Manager, tap **Open...** and select Program.
2. On the Load Program screen, select an existing program and tap Open.
3. In the File Path, verify that the desired program name is displayed.



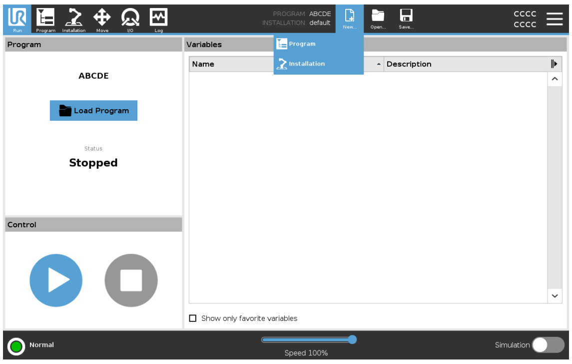
To load an installation

1. In the Program and Installation Manager, tap **Open...** and select Installation.
2. On the Load Robot Installation screen, select an existing installation and tap Open.
3. In the Safety Configuration box, select Apply and restart to prompt robot reboot.
4. Select Set Installation to set installation for the current Program.
5. In the File Path, verify that the desired installation name is displayed.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

To create a new program

1. In the Program and Installation Manager, tap **New...** and select Program.
2. On the Program screen, configure your new program as desired.
3. In the Program and Installation Manager, tap **Save...** and select Save All or Save Program As...
4. On the Save Program As screen, assign a file name and tap Save.
5. In the File Path, verify that the new program name is displayed.



To create a new installation

Save your installation for use after powering down the robot.

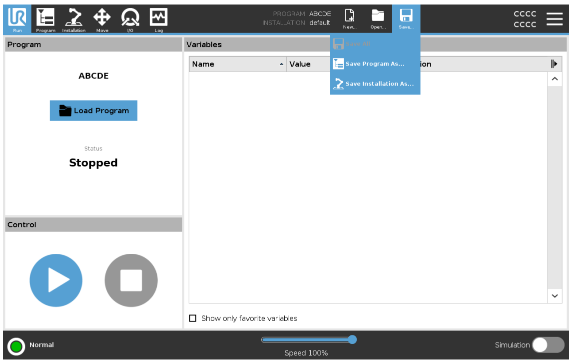
1. In the Program and Installation Manager, tap **New...** and select Installation.
2. Tap Confirm Safety Configuration.
3. On the Installation screen, configure your new installation as desired.
4. In the Program and Installation Manager, tap **Save...** and select Save Installation As...
5. On the Save Robot Installation screen, assign a file name and tap Save.
6. Select Set Installation to set installation for the current Program.
7. In File Path, verify that the new installation name is displayed.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

To use the save options

Save... Depending on the program/installation you load-create, you can:

- **Save All** to save the current Program and Installation immediately, without the system prompting to save to a different location or different name. If no changes are made to the Program or Installation, the Save All... button appears deactivated.
- **Save Program As...** to change the new Program name and location. The current Installation is also saved, with the existing name and location.
- **Save Installation As...** to change the new Installation name and location. The current Program is saved, with the existing name and location.



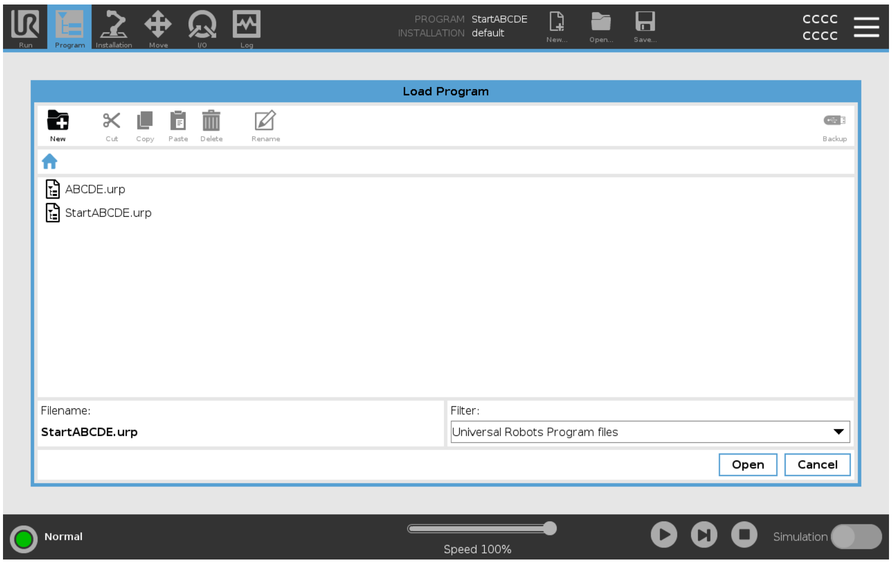
Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

9.1. File Manager

Description

This image shows the load screen which consists of the following buttons:

- Breadcrumb Path
  The breadcrumb path shows a list of directories leading to the present location. By selecting a directory name in the breadcrumb, the location changes to that directory and displays it in the file selection area.
- File Selection Area
  Tap the name of a file to open it. Directories are selected by pressing their name for half a second.
- File Filter
  You can specify the file types shown. After selecting Backup Files, this area displays the 10 most recently saved program versions, where '.old0' is the newest and '.old9' is the oldest.
- Filename
  The selected file is shown here. When saving a file, use the text field to manually enter the file name.
- Action buttons
  The action bar consists of a series of buttons that enable you to manage files.



The 'Backup' action to the right of the action bar supports backing up the currently selected files and directories to the location and to a USB. The 'Backup' action is only enabled when an external media is attached to the USB port.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

10. Hamburger menu

Description

The hamburger menu contains the general settings for PolyScope among these password, system and security settings.

10.1. About

Description

Use the About option to access and display different types of data about the robot. You can display the following types of robot data:

- General
- Version
- Legal

To display data about the robot

1. In the Header, tap the **Hamburger** menu.
2. Select **About**.
3. Tap **General** to access the robot's software version, network settings and serial number.

   For the other data types you can:

   - Tap **Version** to display more detailed data about the robot's software version.
   - Tap **Legal** to display data about the robot's software license/s.
4. Tap **Close** to return to your screen.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

10.1.1. Help

Description

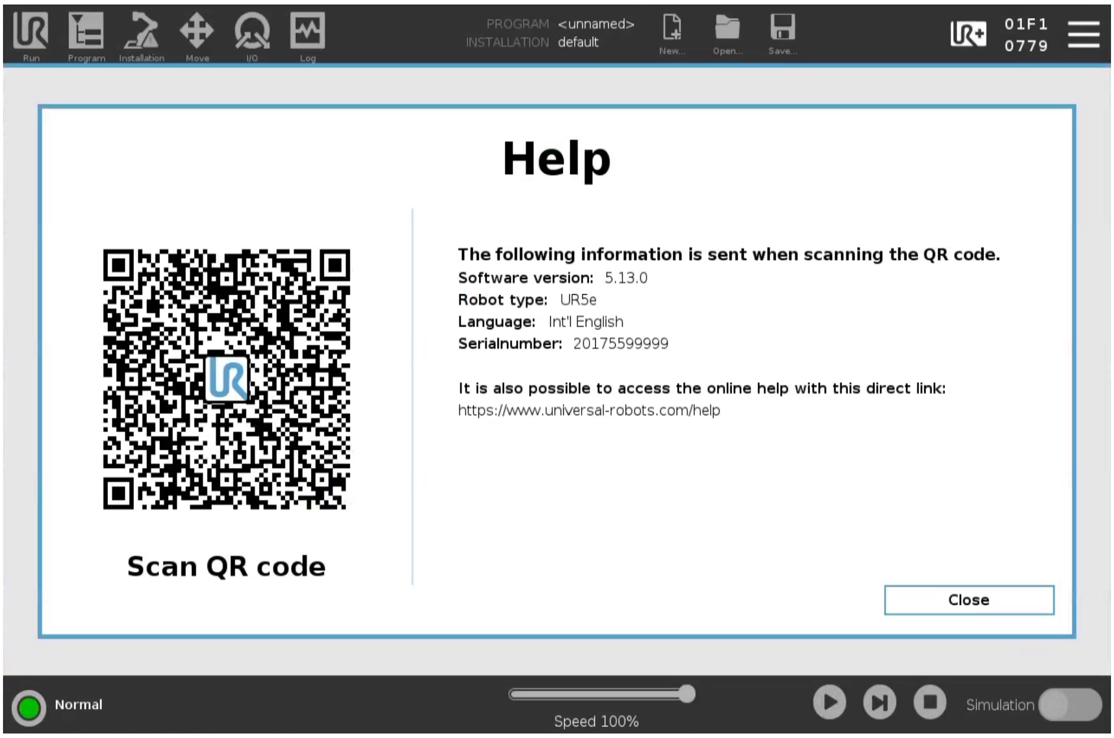
You can access the online help description for PolyScope, the robot arm, control box and other documents that might be helpful. You can access the help via a QR code, or type the following URL in a browser: https://myur.universal-robots.com/manuals.

You can find documentation about the following:

- Robot Arm
- Control Box
- PolyScope
- Other (Service Script and Error Codes)

To find the QR code and URL

1. In the top right corner of PolyScope, tap the Hamburger dropdown menu.
2. In the dropdown menu, select **Help**.
3. Now you can scan the QR code to access help.universal-robots.com.



NOTICE

When you scan the QR code the following information will be sent with the QR code, and can be used in customer analytics at Universal Robots:

- PolyScope software version installed
- Robot type and size
- Language in Polyscope
- Serial number of the robot arm

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

10.2. Settings

To personalize PolyScope settings	1. In the Header, tap the Hamburger menu and select **Settings**. 2. In the Side Menu on the left, select an item to personalize. If an operational mode password was set, in the Side Menu, **System** is only available to the programmer. 3. On the bottom right, tap **Apply and Restart** to implement your changes. 4. On the bottom left, tap **Exit** to close Settings screen without changes.

10.2.1. Preferences

Description	Preferences contain the most basic settings, and would probably only be set once at first start up.

Languages

Description	You can change the PolyScope language and measurement unit (Metric or Imperial).

Run Screen

Description	Located at the base of the Run tab screen, the Speed Slider allows the operator to change the speed of a running Program.
To hide the Speed Slider	1. In the Header, tap the Hamburger menu icon and select **Settings**. 2. Under Preferences, tap **Run Screen**. 3. Select check box to show or hide **Speed Slider**.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Time

Description

You can access and/or adjust the current time and date displayed on the PolyScope.

Time

1. In the Header, tap the Hamburger menu icon and select **Settings**.
2. Under Preferences, select **Time**.
3. Verify and/or adjust **Time** and/or **Date** as desired.
4. Tap **Apply and Restart** to apply your changes.

Date and Time are displayed in the Log tab (see Date Log on page 225) under **Date Log**.

10.2.2. Password

Description

You can create and manage different types of password in PolyScope. An initial password must be set to access the full safety settings. The following password types are described below:

- Administrator
- Operational

Password Settings

To set a Password

You must set a password to Unlock all safety settings that make up your Safety Configuration. If no safety password is applied, you are prompted to set it up.

1. In your PolyScope header right corner, press the **Hamburger** menu and select **Settings**.
2. On the left of the screen, in the blue menu, press **Password** and select **Safety**.
3. In **New password**, type a password.
4. Now, in **Confirm new password**, type the same password and press **Apply**.
5. In the bottom left of the blue menu, press **Exit** to return to previous screen.

You can press the **Lock** tab to lock all Safety settings again or simply navigate to a screen outside of the Safety menu.

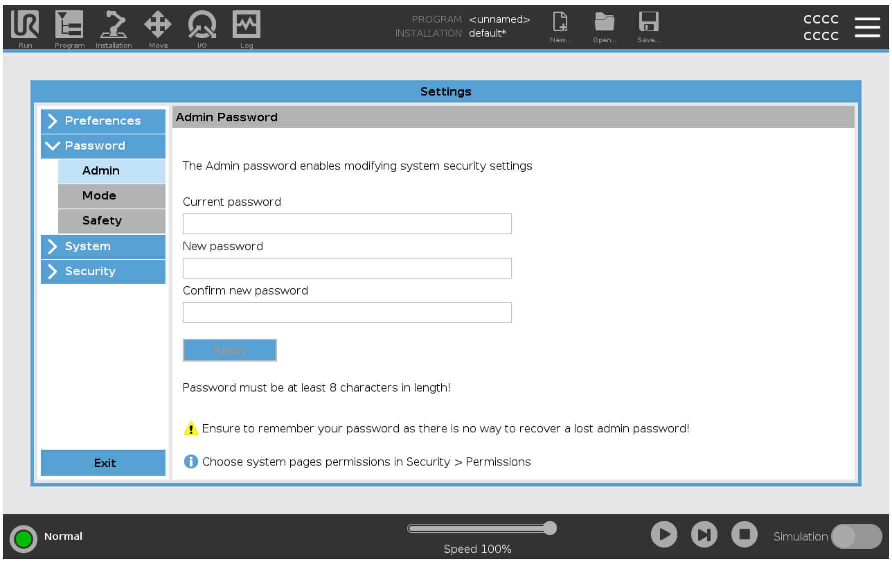
Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

10.2.3. Accessing System and Security

Description

The Admin password is also required to access System and Security, see 1 Administrator Password on page 1.

When you change the default password to your own password, you can use it to access the settings under System (see 10.2.4 System on page 240) and to access the settings under Security (see 10.2.5 Security on page 248).



The default admin password for a Universal Robots robot arm controller is “easybot”. It is a factory setting, configured on all new robots.

> **WARNING**
>
> It is critically important that you change this default admin password to your own password, to ensure cyber security of your robot.

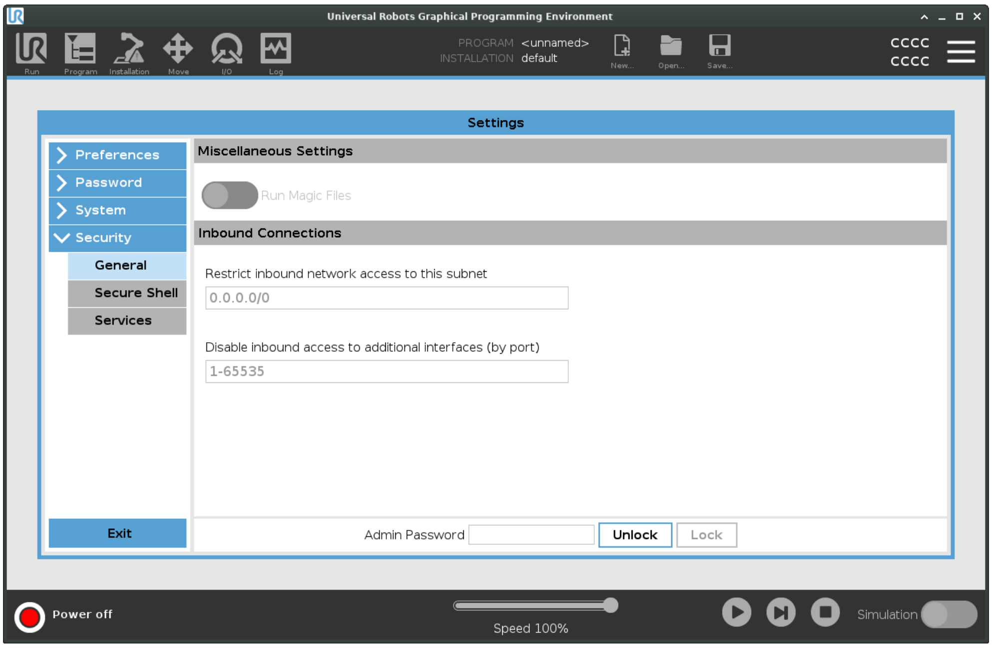
As of PolyScope update 5.17 all security settings are set by default to restrictive (disabled or blocked). (This only applies to new robots and newly created SD-cards. Otherwise, refer to a guide “Secure setup of UR cobots”).

If any of the settings are needed to be enabled for your application, you can easily enable them in the Security screen.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

To use the Admin password

1. In the Header, tap the Hamburger menu icon and select either **System** or **Security** as desired.
2. Locate the dialog box at the base of the screen and enter your Admin password.
3. Tap **Unlock**.



Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

10.2.4. System

Description The system settings control backup of the system, URcaps and networks settings among other.

System Backup

Description

> **NOTICE**
> Use one of the USB ports inside the Control Box (CB) when doing a Backup and Restore operation. Using a CB USB port is more stable and running a Backup requires less time.

> **NOTICE**
> If you restore a system with a new SD card, you must match the Serial Number in the new SD card image when you start up Polyscope. Failure to match the Serial Number can result in an incomplete restore process. A restore error appears for not finding the matching serial number

Backup and Restore Save a full copy of your system to a USB drive and use it to restore your system back to a previous state. This may be necessary after disk corruption or accidental deletion.

To backup the system

1. In the Header, tap the Hamburger menu icon and select **Settings**.
2. Under System, tap **Backup Restore**.
3. Select Location to store the back-up and press **Backup**.
4. Tap **OK** for full system reboot.

To restore the system

1. In the Header, tap the Hamburger menu icon and select **Settings**.
2. Under System, tap **Backup Restore**.
3. Select your Backup file and press **Restore**.
4. Tap **OK** to confirm.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Robot Registration and License File

Description	It is necessary to register the robot and download and install the License File, because the license file will include all available software licenses.
Activate Remote TCP & Toolpath URCap via web	It is possible to activate the Remote TCP & Toolpath URCap directly from www.universal-robots.com/activate. This is only possible for Remote TCP & Toolpath URCap. If you plan to obtain additional licenses via myUR, please activate Remote TCP & Toolpath URCap first.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

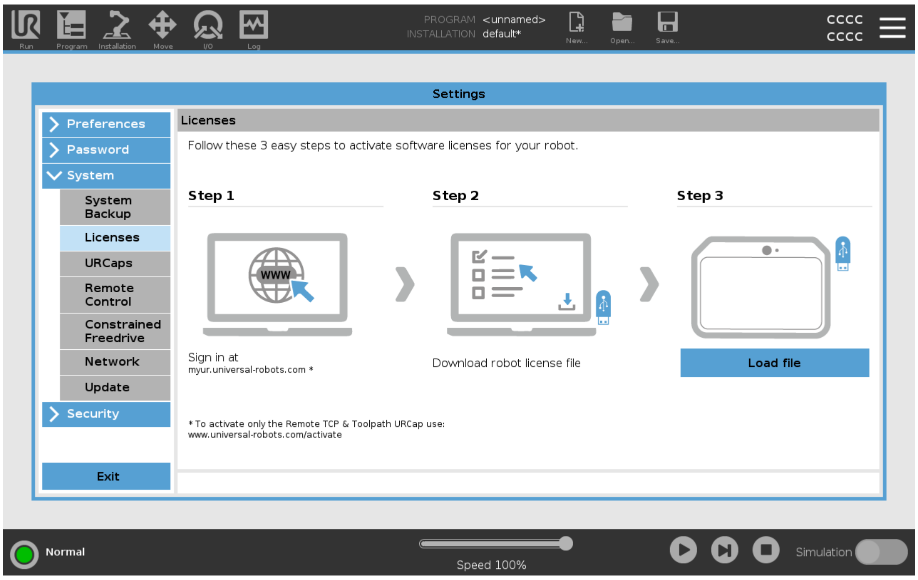
Activate the Software Licenses via myUR

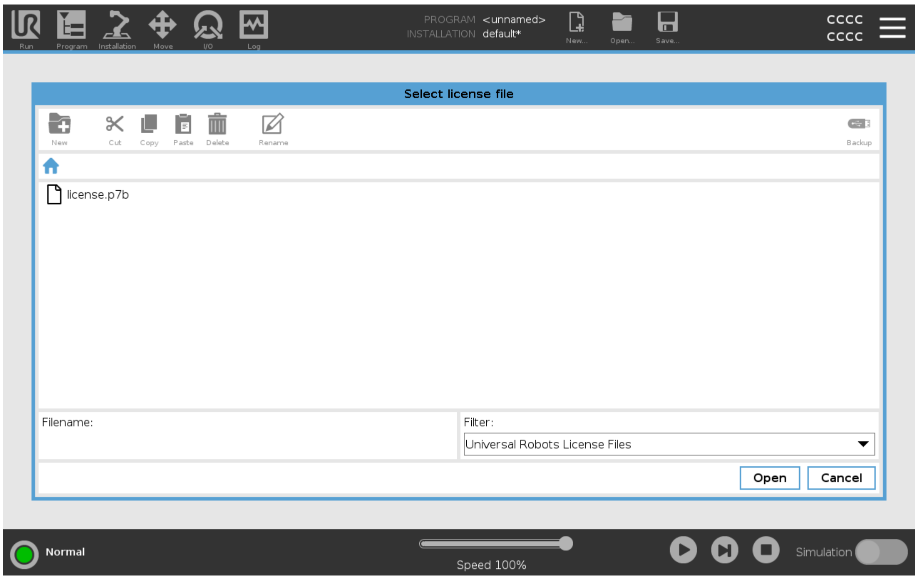
NOTICE

If you have more than one active license, all licenses will be included in the downloaded license file.

If you have not registered your robot, please go to the URL on the screen at step 1 and register your robot.

1. Download the license file to your PC.
2. Copy the license file to the USB and connect it to the Teach Pendant.
3. On the Settings screen, in Step 3, tap **Load file** to open the **Select license file** screen.
4. In the list, select the USB to display content and navigate to the license file.
5. Select **license.p7b** and tap **Open** to confirm robot registration.
6. On the bottom left, tap **Exit**.





Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Deactivate Software Licenses

- A new license file is required if the robot changes owners. In this case, the license file must be deactivated.
- If you purchase a new software license for your robot, it is necessary to deactivate and reactivate the license file in order to add the new software license.

1. In the Header, tap the **Hamburger Menu** and select **Settings**.
2. In the menu on the left, tap **System** and select **Licenses**.
3. On the bottom right of the Settings screen, tap **Deactivate**.

   See Activate the Software Licenses via myUR on the previous page

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

URCaps

Description

Managing URCaps
You can manage your existing **URCaps** or install a new one in your robot.

To manage URCaps

1. In the Header, press the Hamburger menu and select **Settings**.
2. Under System, select **URCaps**.
3. Tap the **+** button, select the **.urcap** file and press **Open**.
4. If you wish to proceed with the installation of that URCap, press **Restart**. After that step, the URCaps is installed and ready to be used.
5. To remove an installed URCaps, select it from Active URCaps, press the **-** button and press **Restart** so changes can take effect.

Active URCaps

Details about the new URCap appear in the **Active URCaps** field.
A status icon indicates the state of the URCap, as listed below:

- URCap ok: The URCap is installed and running normally.
- URCap fault: The URCap is installed but unable to start. Contact the URCap developer.
- URCap restart needed: The URCap has just been installed and a restart is required.

Example

Error messages and information about the URCap appear in the **URCaps Information** field.
Different error messages appear depending on the type of error/s detected.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Remote Control

Description

A robot can either be in Local Control (controlled from the Teach Pendant) or Remote Control (controlled externally).
Remote Control allows you to control the robots via external sources, such as controller sockets, I/Os and the Dashboard Server.
This can be used to send simple commands to PolyScope such as: Starting or loading programs as well as sending UR Script commands directly to the controller.

> **CAUTION**
> To ensure safe usage, the robot can either be in "Remote Control" mode or "Local Control" mode.
> "Local Control" mode will ensure that nay commands, sent to the controller from an external source, will be rejected while the robot is controlled in person.

To enable Remote Control

1. In the Header, tap the Hamburger menu and select **Settings**.
2. Under System, select **Remote Control**.
3. Tap **Enable** to make the Remote Control feature available. PolyScope remains active. Enabling Remote Control does not immediately start the feature. It allows you to switch from Local Control to Remote Control.
4. In the profile menu, select **Remote Control** to alter PolyScope. You can return to Local Control by switching back in the profile menu.

> **NOTICE**
> - Although Remote Control limits your actions in PolyScope, you can still monitor robot state.
> - When a robot system is powered off in Remote Control, it starts up in Remote Control.

Settings Requirements

Control of the robot via network or digital input is, by default, restricted.

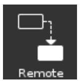
- Enable and select the Remote Control feature removes this restriction.
- Enable Remote Control by switching to the Local Control profile (PolyScope control) of the robot, allowing all control of running programs and executing scripts to be performed remotely.
- Enable the Remote Control feature in Settings to access Remote mode and Local mode in the profile.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Local Control does not allow

- Power on and brake release sent to the robot over network
- Receiving and executing robot programs and installation sent to the robot over network
- Autostart of programs at boot, controlled from digital inputs
- Auto brake release at boot, controlled from digital inputs
- Start of programs, controlled from digital inputs

Remote Control does not allow



- Moving the robot from Move Tab
- Starting from Teach Pendant
- Load programs and installations from the Teach Pendant
- Freedrive

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Network

Description

You can configure robot connection to a network by selecting one of three available network methods:

- DHCP
- Static Address
- Disabled network (if you don't wish to connect your robot to a network)

Depending on the network method you select, configure your network settings:

- IP Address
- Subnet Mask
- Default Gateway
- Preferred DNS Server
- Alternative DNS Server

Update

Description

Install updates from a USB to ensure the robot software is up-to-date.
Please use this link to find the latest software update: Download Latest Software

To update software

1. In the Header, tap the Hamburger menu icon and select **Settings**.
2. Under System, tap **Update**.
3. Insert an USB and tap **Search** to list valid update files.
4. In the list of valid update files, select desired version and tap **Update** to install.

WARNING
Always check your program/s after a software upgrade. The upgrade might change the trajectories in your program.

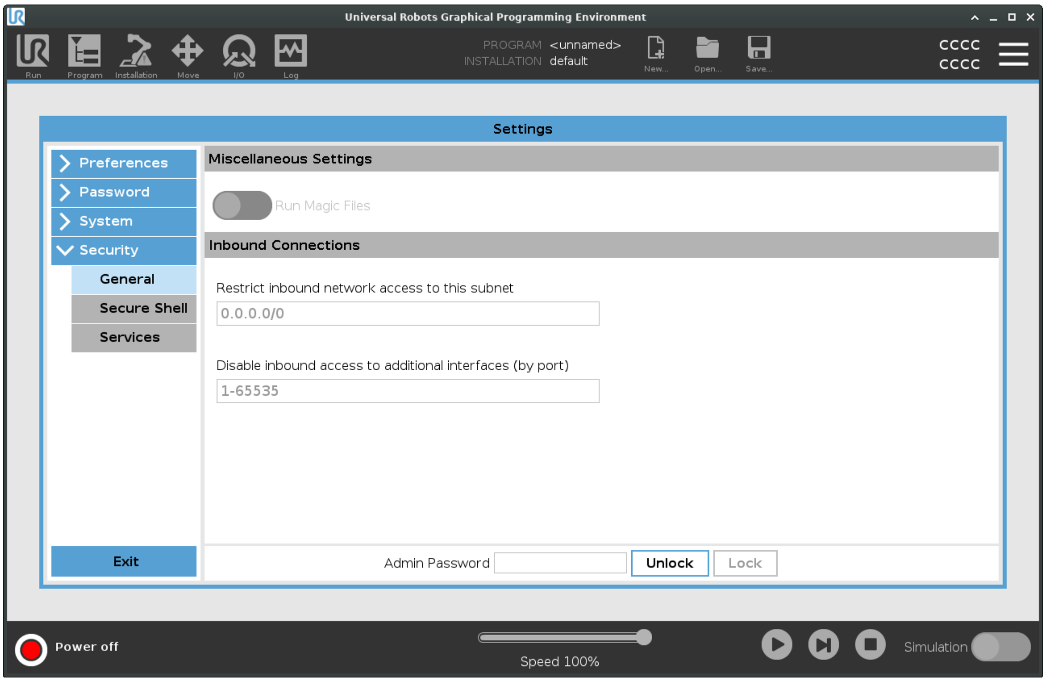
Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

10.2.5. Security

Description You can find information on how to manage and protect your system.

General

Description In the General settings, you can enable magic files and configure inbound connections. A Magic File is a script on a USB drive that executes once it is inserted into the system. This function is disabled by default to make sure that magic files are not unknowingly executed on PolyScope.



Magic Files Magic files have unrestricted privileges to make system changes, thus they must be considered as a security liability.

To enable Magic Files on PolyScope

1. In the Header, tap the Hamburger menu and select **Settings**.
2. Under Security, select **General**.
3. Enable **Run Magic Files**.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Restrict Inbound Connections

The network access has been set to 0.0.0.0/0 has a security setting to ensure that there is no access to the subnet in PolyScope.

> **WARNING**
> URCaps may require particular network interfaces to be open in order to function.
>
> - Consult your URCaps vendor/s, if any of your URCaps require particular network interfaces (ports/services) to be open.

Configuring Inbound Connections

Use **Restrict inbound network access to a specific subnet** to make sure network connections originating from an IP-address outside the indicated subnet will be refused. For example:

- Use 192.168.1.0/24 to only allow access from hosts in the range of 192.168.1.0 - 192.168.1.255.
- Use 192.168.1.96 to allow inbound access only from this host.

1. In the Header, tap the Hamburger menu and select **Settings**.
2. Under Security, select **General**.
3. Enter your Admin password.
4. Enter subnet restrictions under **Restrict inbound network access to a specific subnet**.

Disable Inbound Access

Use **Disable inbound access to additional interfaces (by port)** to make sure any inbound connection to the designated ports will be refused.

1. In the Header, tap the Hamburger menu and select **Settings**.
2. Enter the interfaces to be closed in **Disable inbound access to additional interfaces (by port)**.

Leave the field blank to avoid blocking ports. Any enabled service Services on page 252 will take precedence over port blocking. Even if a port if blocked in the general security settings, it will be open by an enabled service.

Example

- You can block all ports
  - Use 1-65535 to block all ports.
- You can block one specfic port
  - You can use port number 564 to block port 564.
- You can block a range of ports
  - Use ranges to block a specfic range of ports. 2318-3412, 22, 56-67 to block specific ports and specific ranges of ports.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

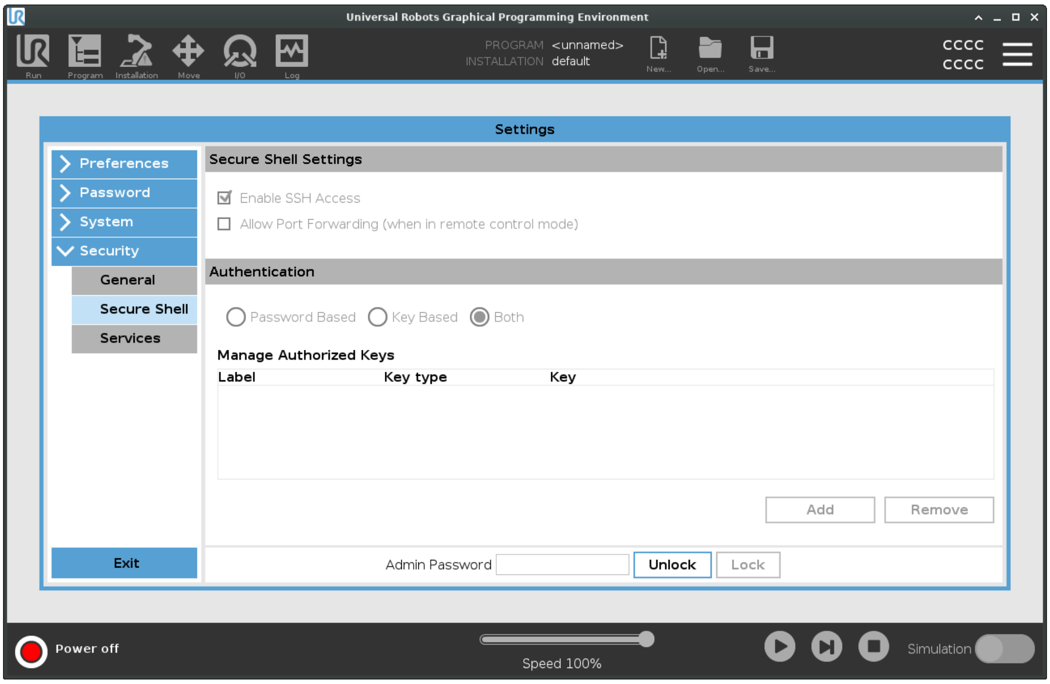
Secure Shell

Description

Secure Shell (SSH) provides a private (encrypted) and authenticated connection to the robot allowing:

- operating system access
- file copying
- tunneling of network interfaces

> **NOTICE**
> SSH is a powerful tool if used as intended. Make sure you understand how to use SSH technology securely before you enable it on your robot.



To enable SSH Access

1. In the Header, tap the Hamburger menu and select **Settings**.
2. Under Security, select **Secure Shell**.
3. Configure the Secure Shell Settings:
   - Select **Enable SSH Access**.
   - Select to enable/disable **Allow Port Forwarding (when in remote control mode)**.

     Port forwarding is only available in remote control mode.

     Port forwarding is a recommended technique for wrapping open interfaces (eg. the Dashboard service) in a secure and encrypted tunnel requiring authentication.
4. Select the Authentication type.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Authentication

Any SSH connection requires the connecting user to authenticate when the connection is established. You can set up authentication with a password and/or with a pre-shared, authorized key.
Key based authentication relies on pre-shared keys.

To use Authentication

Available keys are listed here together with buttons for removing a selected key from the list and for adding new keys.

1. Tap **Add** to open a file selection dialog.
2. Select a key from the file.

   The file is read line by line adding only lines which are not blank and not identified as comments (starting with #). No validation of lines added is done.
3. They must comply with the format used for authorized_keys.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Services

Description

Services lists the standard services running on the robot. You can enable or disable each service.

> **NOTICE**
> All services are disabled as a security feature. When you start or configure your robot, you have to enable relevant services.

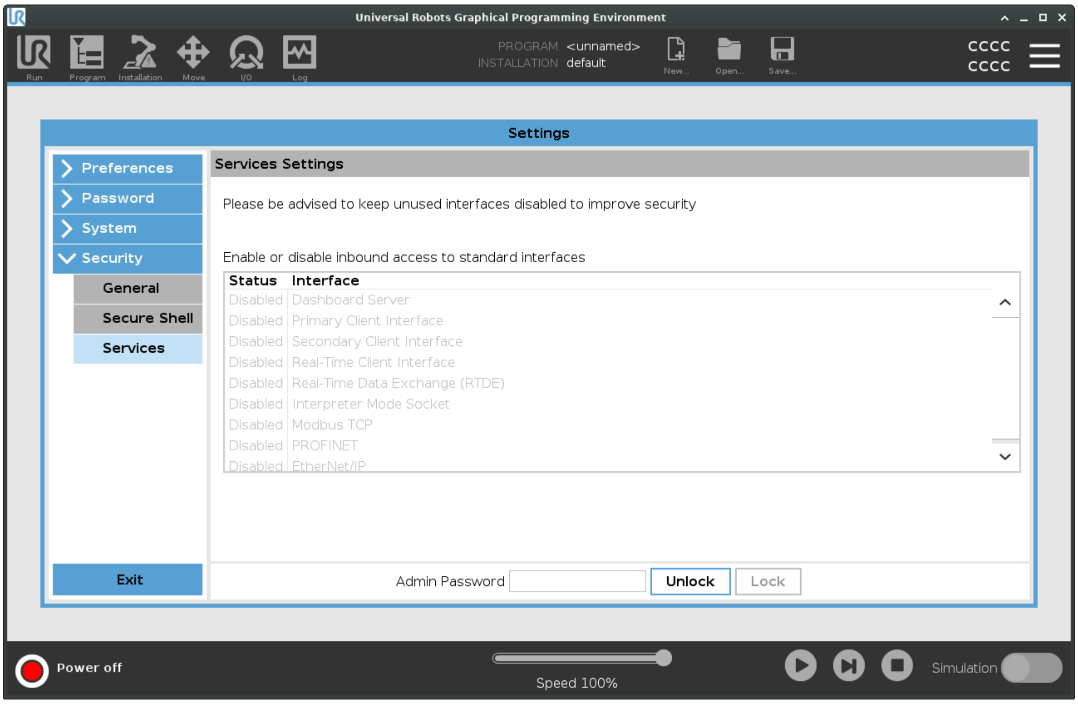
An enabled service remains available even if the ports associated with that service are blocked. So, simply blocking a port is not enough to prevent access to the listed services if they are enabled.

Enabling Services

You have to enable the relevant service for the function that you are using.

To enable Services

1. In the Header, tap the Hamburger menu and select **Settings**.
2. Under Security, select **Services**.
3. In the list, select an option and tap **Enable**, or tap **Disable**.



Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Permissions

Description

In Permissions you can restrict the settings in the System section with Administrative password protection.

Using Permissions

You have to enable a permission to be restricted and disable the permission for the restriction to be removed.
Before you can enable/disable Permissions, you have to unlock the Security Settings.

To unlock Security Settings

1. In the Header, tap the Hamburger menu.
2. Select **Settings**.
3. Under Security, select **Permissions**.
4. At the bottom of the screen tap **Admin Password**.
5. Use the on-screen keyboard to enter your password and tap **Submit**.

   You can now access and configure the all Settings including Permissions

Navigating away from the Security Settings locks the screen again.

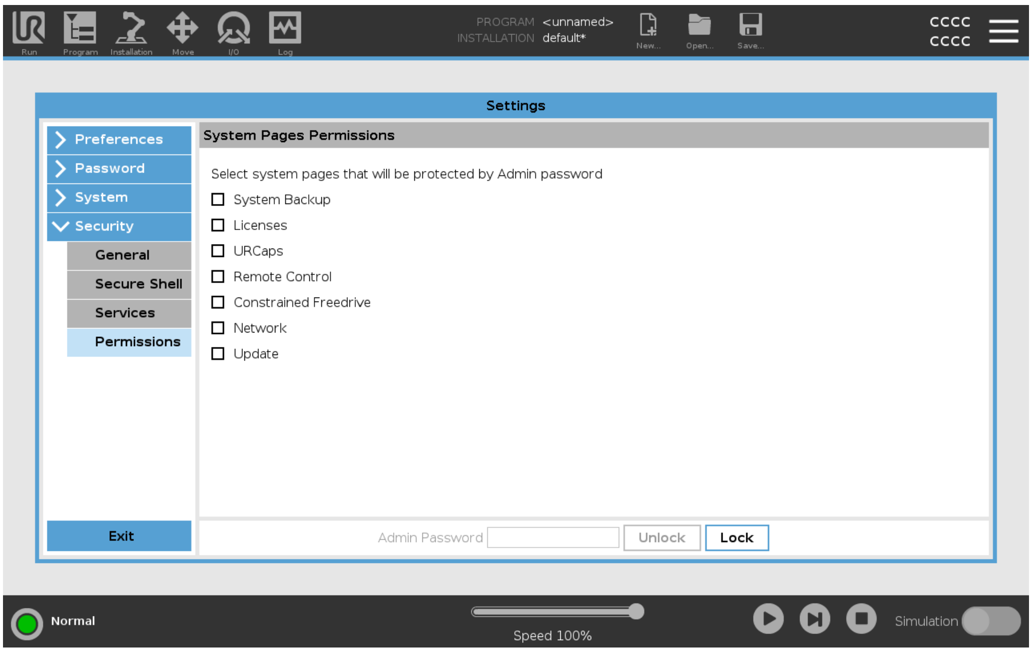
To enable/disable Permissions

Once you unlock the Settings, you can select any of the options in the list. Selecting an option, removes the checkmark and allows access to that selection.

1. On the Permissions screen, select an option from the list to remove the checkmark.

   The selection is unlocked.
2. To lock the option again, select it again to add the checkmark.

   The selection is locked.



Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

10.3. Shutdown Robot

Description	The **Shutdown Robot** button allows the robot to be powered off or restarted.
To shut down the robot	1. In the Header, tap the Hamburger menu and select **Shutdown Robot**. 2. When the Shutdown Robot dialog box appears, tap **Power Off**.

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.

Software Name: PolyScope 5
Software Version: 5.25
Document Version: 10.16.242

Copyright © 2009-2025 by Universal Robots A/S. All rights reserved.