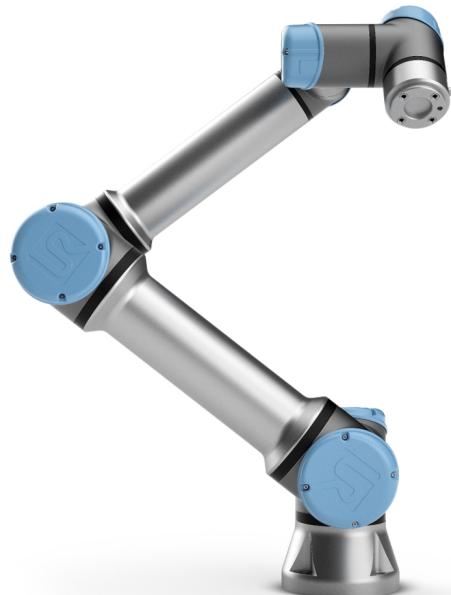




# UNIVERSAL ROBOTS

## Universal Robots e-Series User Manual



UR5e  
Original instructions (en)



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 © 2009-2021 by Universal Robots A/S.

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



# Contents

<b>1. Preface</b>	1
1.1. What Do the Boxes Contain	2
1.2. Important Safety Notice	2
1.3. How to Read This Manual	3
1.4. Where to Find More Information	3
1.4.1. UR+	3
<b>Part I Hardware Installation Manual</b>	5
<b>2. Safety</b>	7
2.1. Introduction	7
2.2. Validity and Responsibility	7
2.3. Limitation of Liability	8
2.4. Warning Symbols in this Manual	8
2.5. General Warnings and Cautions	9
2.6. Intended Use	12
2.7. Risk Assessment	13
2.8. Pre-Use Assessment	14
2.9. Emergency Stop	15
2.10. Movement Without Drive Power	15
<b>3. Safety-related Functions and Interfaces</b>	17
3.1. Introduction	17
3.2. Stop Categories	17
3.3. Configurable Safety Functions	18
3.4. Safety Function	21
3.5. Modes	22
<b>4. Transportation</b>	25
<b>5. Mechanical Interface</b>	27
5.1. Introduction	27
5.2. Workspace of the Robot	27
5.3. Mounting	27
5.4. Maximum Payload	30
<b>6. Electrical Interface</b>	32
6.1. Introduction	32



6.1.1. Control Box Bracket .....	32
6.2. Ethernet .....	32
6.3. Electrical Warnings and Cautions .....	33
6.4. Controller I/O .....	34
6.4.1. Common specifications for all digital I/O .....	35
6.4.2. Safety I/O .....	36
6.4.3. General purpose digital I/O .....	41
6.4.4. Digital Inputs from a button .....	41
6.4.5. Communication with other machines or PLCs .....	42
6.4.6. General purpose analog I/O .....	42
6.4.7. Remote ON/OFF control .....	43
6.5. Mains Connection .....	44
6.6. Robot Connection: Robot Cable .....	46
6.6.1. Robot Cable connector .....	46
6.7. Robot Connection: Base Flange Cable .....	47
6.7.1. Base Flange Cable connector .....	47
6.8. Tool I/O .....	47
6.8.1. Tool Power Supply .....	49
6.8.2. Power Supply .....	49
6.8.3. Dual Pin Power Supply .....	49
6.8.4. Tool Digital Outputs .....	49
6.8.5. Tool Digital Inputs .....	51
6.8.6. Tool Analog Input .....	51
6.8.7. Tool Communication I/O .....	52
<b>7. Maintenance and Repair .....</b>	<b>53</b>
7.1. Safety Instructions .....	53
7.2. Cleaning .....	54
7.3. Inspection .....	54
7.3.1. Robot Arm Inspection Plan .....	54
7.3.2. Robot Arm Visual Inspection .....	55
7.3.3. Control Box Inspection Plan .....	55
7.3.4. Control Box Visual Inspection .....	56
7.3.5. Freedrive Inspection .....	56
<b>8. Disposal and Environment .....</b>	<b>58</b>
<b>9. Certifications .....</b>	<b>59</b>
<b>10. Warranties .....</b>	<b>61</b>

10.1. Product Warranty .....	61
10.2. Disclaimer .....	61
<b>11. Stopping Time and Stopping Distance .....</b>	<b>62</b>
<b>12. Declarations and Certificates .....</b>	<b>65</b>
<b>13. Certificates .....</b>	<b>67</b>
<b>14. Applied Standards .....</b>	<b>74</b>
<b>15. Technical Specifications .....</b>	<b>80</b>
<b>16. Safety Functions Tables .....</b>	<b>82</b>
16.1. Table 1 .....	82
16.2. Table 2 .....	88
<b>Part II PolyScope Manual .....</b>	<b>93</b>
<b>17. Introduction .....</b>	<b>95</b>
17.1. Robot Arm Basics .....	95
17.2. PolyScope Basics .....	95
17.2.1. Touch Screen .....	96
17.2.2. Header Icons/Tabs .....	96
17.2.3. Footer Buttons .....	98
17.3. Installation .....	98
17.3.1. Installing the Robot Arm and Control Box .....	98
17.3.2. Turning the Control Box On/Off .....	99
17.4. Initialization .....	99
17.4.1. Starting up the Robot Arm .....	100
17.5. Quick System Start-up .....	101
17.6. The First Program .....	101
17.7. Robot Registration and URCap License files .....	102
17.7.1. Registering the Robot from Your Current Screen .....	103
17.7.2. Downloading the URCAP License File .....	103
17.7.3. Unregistering Robot .....	104
17.8. Robot Cyber Security .....	104
17.8.1. Cyber Security Pre-requisites .....	104
17.8.2. Hardening Cyber Security .....	105
<b>18. Freedrive .....</b>	<b>107</b>
18.1. Enabling Freedrive: Standard Teach Pendant .....	108
18.1.1. Using the Freedrive Button .....	109
18.1.2. Using the Freedrive Button on the Move Tab Screen .....	109



18.2. Enabling Freedrive: 3PE Teach Pendant .....	109
<b>19. Backdrive .....</b>	<b>110</b>
19.1. Enabling Backdrive: Standard Teach Pendant .....	110
19.2. Enabling Backdrive: 3PE Teach Pendant .....	110
<b>20. Operational Mode Selection .....</b>	<b>111</b>
20.1. Operational Modes .....	111
20.2. Three-Position Enabling Device .....	113
20.2.1. Manual High Speed .....	113
<b>21. Safety Configuration .....</b>	<b>115</b>
21.1. Safety Settings Basics .....	115
21.1.1. Accessing Safety Configuration .....	115
21.2. Setting a Safety Password .....	116
21.3. Changing the Safety Configuration .....	116
21.4. Applying New Safety Configuration .....	117
21.5. Safety Checksum .....	117
21.6. Safety Menu Settings .....	117
21.7. Robot Limits .....	117
21.8. Safety Modes .....	119
21.9. Tolerances .....	120
21.10. Joint Limits .....	120
21.11. Planes .....	121
21.11.1. Modes .....	121
21.11.2. Configuring Safety Planes .....	122
21.11.3. Elbow .....	123
21.11.4. Color Codes .....	123
21.12. Freedrive .....	124
21.12.1. Using the Freedrive Button .....	124
21.13. Backdrive .....	124
21.13.1. Enabling Backdrive .....	124
21.14. Tool Position .....	125
21.15. Tool Direction .....	126
21.15.1. Limit Properties .....	127
21.15.2. Tool Properties .....	128
21.16. I/O .....	128
21.16.1. Input Signals .....	128

21.16.2. Output Signals .....	129
21.16.3. OSSD Safety Signals .....	130
21.17. Hardware .....	130
21.17.1. Selecting Available Hardware .....	131
21.18. Safe Home Position .....	131
21.18.1. Syncing from Home .....	132
21.19. Safe Home Output .....	132
21.19.1. Defining Safe Home Output .....	132
21.20. Edit Safe Home .....	132
21.20.1. Editing Safe Home .....	132
<b>22. Run Tab .....</b>	<b>133</b>
22.1. Program .....	133
22.2. Variables .....	133
22.3. Robot Age .....	134
22.4. Move Robot into Position .....	134
22.4.1. Accessing the Move Robot into Position Screen .....	134
22.4.2. Move robot to: .....	135
22.4.3. Manual .....	135
<b>23. Program Tab .....</b>	<b>137</b>
23.1. Program Tree .....	137
23.1.1. Program Execution Indication .....	138
23.1.2. Search Button .....	138
23.2. Program Tree Toolbar .....	138
23.2.1. Undo/Redo Buttons .....	138
23.2.2. Move Up & Down .....	138
23.2.3. Cut .....	138
23.2.4. Copy .....	139
23.2.5. Paste .....	139
23.2.6. Delete .....	139
23.2.7. Suppress .....	139
23.3. Expression Editor .....	139
23.4. Starting a Program from a Selected Node .....	140
23.4.1. Using Play from Selection .....	140
23.5. Using Breakpoints in a Program .....	141
23.6. Single Step in a Program .....	142



23.7. Command Tab .....	142
23.8. Graphics Tab .....	143
23.9. Variables Tab .....	145
23.10. Basic program nodes .....	146
23.10.1. Move .....	146
23.10.2. Fixed Waypoint .....	149
23.10.3. Relative Waypoint .....	154
23.10.4. Variable Waypoint .....	155
23.10.5. Direction .....	156
23.10.6. Until .....	157
23.10.7. Until-Tool Contact .....	157
23.10.8. Wait .....	159
23.10.9. Set .....	160
23.10.10. Popup .....	160
23.10.11. Halt .....	161
23.10.12. Comment .....	162
23.10.13. Folder .....	162
23.10.14. Set Payload .....	163
23.11. Advanced program nodes .....	164
23.11.1. Loop .....	164
23.11.2. If .....	164
23.11.3. SubProgram .....	166
23.11.4. Assignment .....	167
23.11.5. Script .....	167
23.11.6. Event .....	168
23.11.7. Thread .....	169
23.11.8. Switch .....	169
23.11.9. Timer .....	170
23.11.10. Home .....	171
23.12. Templates .....	171
23.12.1. Palletizing .....	171
23.12.2. Seek .....	176
23.12.3. Force .....	180
23.12.4. Force value selection .....	182
23.12.5. Speed limits .....	182
23.12.6. Test force settings .....	182
23.12.7. Conveyor Tracking .....	183
23.12.8. Screwdriving .....	184

23.12.9. Screwdrive Until .....	185
<b>23.13. URCaps .....</b>	<b>186</b>
23.13.1. Remote TCP and Toolpath URCap .....	186
23.13.2. Remote TCP Movement Types .....	188
23.13.3. Remote TCP Waypoint .....	188
23.13.4. Remote TCP Toolpath .....	189
23.13.5. Remote TCP .....	191
23.13.6. Remote TCP PCS .....	191
23.13.7. Regular TCP Toolpath Moves .....	194
<b>24. Installation Tab .....</b>	<b>197</b>
24.1. General .....	197
24.2. TCP Configuration .....	197
24.2.1. Position .....	197
24.2.2. Orientation .....	197
24.2.3. Adding, Renaming, Modifying and Removing TCPs .....	198
24.2.4. Teaching TCP position .....	198
24.2.5. Teaching TCP orientation .....	199
24.3. Payload .....	199
24.3.1. Adding, Renaming, Modifying and Removing Payloads .....	200
24.3.2. Setting the Center of Gravity .....	200
24.3.3. Payload Estimation .....	201
24.3.4. Inertia .....	201
24.4. Mounting .....	202
24.5. I/O Setup .....	203
24.5.1. I/O Signal Type .....	203
24.5.2. Assigning User-defined Names .....	204
24.5.3. I/O Actions and I/O Tab Control .....	204
24.6. Installation Variables .....	206
24.7. Startup .....	207
24.7.1. Loading a Startup Program .....	207
24.7.2. Starting a Startup Program .....	208
24.8. Tool I/O .....	208
24.9. I/O Interface Control .....	208
24.10. Tool Analog Inputs .....	209
24.10.1. Tool Communication Interface .....	209
24.10.2. Configuring the Tool Communication Interface (TCI) .....	209



---

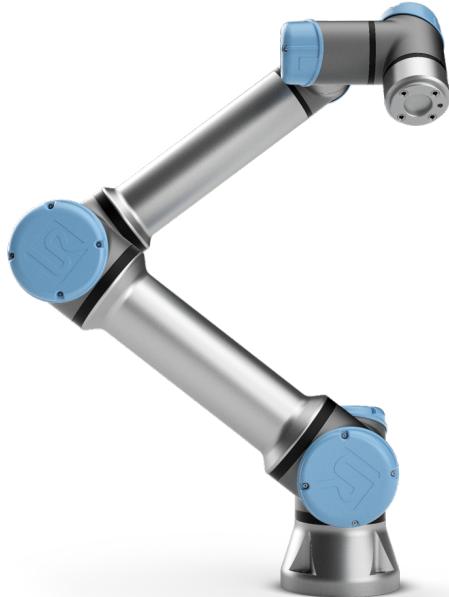
24.11. Digital Output Mode .....	209
24.11.1. Dual Pin Power .....	210
24.12. Smooth Transition Between Safety Modes .....	210
24.12.1. Adjusting Acceleration/Deceleration Settings .....	210
24.13. Home .....	210
24.13.1. Defining Home .....	211
24.14. Conveyor Tracking Setup .....	211
24.14.1. Defining a Conveyor .....	211
24.14.2. Conveyor Parameters .....	211
24.14.3. Tracking Parameters .....	211
24.15. Screwdriving Setup .....	212
24.15.1. Configuring a Screwdriver .....	213
24.15.2. Configuring the Screwdriver Position .....	213
24.15.3. Configuring the Screwdriver Interface .....	214
24.16. Safety .....	214
24.17. Features .....	215
24.17.1. Using a feature .....	216
24.17.2. Using Move here .....	217
24.17.3. Point feature .....	217
24.17.4. Line feature .....	217
24.17.5. Plane Feature .....	218
24.17.6. Example: Manually Updating a Feature to Adjust a Program .....	219
24.17.7. Example: Dynamically Updating a Feature Pose .....	220
24.17.8. Feature Edit .....	222
24.18. Fieldbus .....	224
24.19. MODBUS Client I/O Setup .....	224
24.19.1. Refresh .....	224
24.19.2. Add unit .....	224
24.19.3. Delete unit .....	224
24.19.4. Set unit IP .....	224
24.19.5. Sequential mode .....	225
24.19.6. Add signal .....	225
24.19.7. Delete signal .....	225
24.19.8. Set signal type .....	225
24.19.9. Set signal address .....	226
24.19.10. Set signal name .....	226
24.19.11. Signal value .....	226

24.19.12. Signal connectivity status .....	226
24.19.13. Show Advanced Options .....	227
24.19.14. Advanced Options .....	227
24.20. EtherNet/IP .....	227
24.21. PROFINET .....	228
<b>25. Move Tab .....</b>	<b>229</b>
25.1. Move Tool .....	229
25.2. Robot .....	229
25.2.1. Feature .....	230
25.2.2. Active TCP .....	230
25.2.3. Home .....	230
25.2.4. Freedrive .....	230
25.2.5. Align .....	230
25.3. Tool Position .....	230
25.4. Joint Position .....	230
25.5. Pose Editor Screen .....	231
<b>26. I/O Tab .....</b>	<b>235</b>
26.1. Robot .....	235
26.2. MODBUS .....	236
<b>27. Log Tab .....</b>	<b>239</b>
27.1. Readings and Joint Load .....	239
27.2. Date Log .....	239
27.3. Saving Error Reports .....	240
27.4. Technical Support File .....	241
<b>28. Program and Installation Manager .....</b>	<b>243</b>
28.1. Open... .....	243
28.2. New... .....	244
28.3. Save... .....	245
28.4. File manager .....	246
<b>29. Hamburger menu .....</b>	<b>247</b>
29.1. Help .....	247
29.2. About .....	247
29.3. Settings .....	247
29.3.1. Preferences .....	247
29.3.2. Password .....	248

---

29.4. System .....	249
29.4.1. Backup and Restore .....	249
29.4.2. Update .....	249
29.4.3. Network .....	250
29.4.4. Managing URCaps .....	250
29.4.5. Remote Control .....	251
29.4.6. Security .....	252
Running magic files .....	252
Configure Inbound Connections .....	253
Authentication .....	254
Managing Authorized Keys .....	254
29.5. Shutdown Robot .....	254
<b>30. Glossary .....</b>	<b>255</b>
30.1. Index .....	256

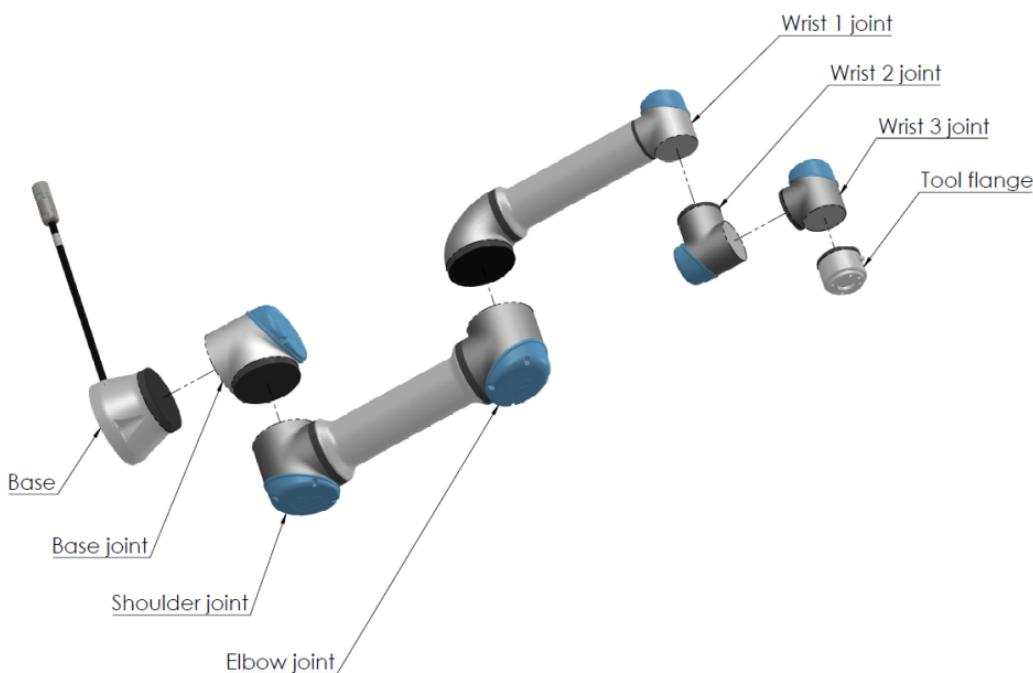
# 1. Preface



Congratulations on the purchase of your new Universal Robots e-Series robot.

The robot can be programmed to move a tool, and communicate with other machines using electrical signals. It is an arm composed of extruded aluminium tubes and joints.

Using our patented programming interface, PolyScope, it is easy to program the robot to move the tool along a desired trajectory.



2.1: The joints, the base and the tool flange of the Robot Arm.

With six joints and a wide scope of flexibility, Universal Robots e-Series collaborative robot arms are designed to mimic the range of motion of a human arm. Using our patented programming interface, PolyScope, it is easy to program the robot to move tools and communicate with other machines using electrical signals. Figure 2.1: The joints, the base and the tool flange of the Robot Arm. above illustrates the main components of the robot arm and can be used as a reference throughout the manual.

## 1.1. What Do the Boxes Contain

When you order a robot, you receive two boxes. One contains the Robot Arm, the other contains:

- Control Box with Teach Pendant
- Mounting bracket for the Control Box
- Mounting bracket for the Teach Pendant
- Key for opening the Control Box
- Cable for connecting the robot arm and the Control Box (see options in [15. Technical Specifications on page 80](#))
- Mains cable or Power cable compatible to your region
- This manual

## 1.2. Important Safety Notice

The robot is **partly completed machinery** (see [2. Safety on page 7](#)) and as such a risk assessment is required for each installation of the robot. You must follow all of the safety instructions in chapter [2. Safety on page 7](#).

## 1.3. How to Read This Manual

This manual contains instructions for installing and programming the robot. The manual is separated into two parts:

*Hardware Installation Manual*

The mechanical and electrical installation of the robot.

*PolyScope Manual*

Programming of the robot.

This manual is intended for the robot integrator who must have a basic level of mechanical and electrical training, as well as be familiar with elementary programming concepts.

## 1.4. Where to Find More Information

The support website (<http://www.universal-robots.com/support>) contains the following:

- Other language versions of this manual
- **PolyScope Manual**
- The **Service Manual** with instructions for troubleshooting, maintenance and repair
- The **Script Manual** for advanced users

### 1.4.1. UR+

The UR+ site (<http://www.universal-robots.com/plus/>) is an online showroom that provides cutting-edge products to customize your UR robot application. You can find everything you need in one place—from end-effectors and accessories to vision cameras and software. All products are tested and approved to integrate with UR robots, ensuring simple set-up, reliable operation, smooth user experience, and easy programming. You can also use the site to join the UR+ Developer Program to access our new software platform which allows you to design more user-friendly products for UR robots.



# Part I

# Hardware Installation Manual



# 2. Safety

## 2.1. Introduction

This chapter contains important safety information, which must be read and understood by the integrator of Universal Robots e-Series robots **before** the robot is powered on for the first time.

In this chapter, the first subsections are general. The later subsections contain specific engineering data relevant to enable setting up and programming the robot. Chapter [3. Safety-related Functions and Interfaces on page 17](#) describes and defines safety-related functions particularly relevant for collaborative applications.

Instructions and guidance provided in chapter [3. Safety-related Functions and Interfaces on page 17](#) as well as in section [2.7. Risk Assessment on page 13](#) are particularly important.

It is essential to observe and follow all assembly instructions and guidance provided in other chapters and parts of this manual.

Special attention shall be paid to text associated with warning symbols.



### NOTE

Universal Robots disclaims any and all liability if the robot (arm control box and/or teach pendant) is damaged, changed or modified in any way. Universal Robots cannot be held responsible for any damages caused to the robot or any other equipment due to programming errors or malfunctioning of the robot.

## 2.2. Validity and Responsibility

The information in this manual does not cover designing, installing and operating a complete robot application, nor does it cover all peripheral equipment that can influence the safety of the complete system. The complete system must be designed and installed in accordance with the safety requirements set forth in the standards and regulations of the country where the robot is installed.

The integrators of Universal Robots e-Series robots are responsible for ensuring that the applicable safety laws and regulations in the country concerned are observed and that any significant hazards in the complete robot application are eliminated. This includes, but is not limited to:

- Performing a risk assessment for the complete robot system
- Interfacing other machines and additional safety devices if defined by the risk assessment
- Setting up the appropriate safety settings in the software
- Ensuring that the user will not modify any safety measures
- Validating that the total robot system is designed and installed correctly

- Specifying instructions for use
- Marking the robot installation with relevant signs and contact information of the integrator
- Collecting all documentation in a technical file; including the risk assessment and this manual

## 2.3. Limitation of Liability

Any safety information provided in this manual must not be construed as a warranty, by UR, that the industrial manipulator will not cause injury or damage, even if industrial manipulator complies with all safety instructions.

## 2.4. Warning Symbols in this Manual

The symbols below define the captions specifying the danger levels used throughout this manual. The same warning signs are used on the product.



### WARNING

This indicates an imminently hazardous electrical situation which, if not avoided, could result in death or serious injury.



### WARNING

This indicates an imminently hazardous situation which, if not avoided, could result in death or serious injury.



### WARNING

This indicates a potentially hazardous electrical situation which, if not avoided, could result in injury or major damage to the equipment.



### WARNING

This indicates a potentially hazardous situation which, if not avoided, could result in injury or major damage to the equipment.



### WARNING

This indicates a potentially hazardous hot surface which, if touched, could result in injury.

**CAUTION**

This indicates a situation which, if not avoided, could result in damage to the equipment.

## 2.5. General Warnings and Cautions

This section contains some general warnings and cautions that can be repeated or explained in different parts of this manual. Other warnings and cautions are present throughout this manual.

**WARNING**

Make sure to install the robot and all electrical equipment according to the specifications and warnings found in chapters [5. Mechanical Interface](#) on page 27 and



## WARNING

1. Make sure the robot arm and tool/end effector are properly and securely bolted in place.
2. Make sure the robot arm has ample space to operate freely.
3. Make sure that safety measures and/or robot safety configuration parameters have been set up to protect both programmers, operators and bystanders, as defined in the risk assessment.
4. Do not wear loose clothing or jewellery when working with the robot. Make sure long hair is tied back when working with the robot.
5. Never use the robot if it is damaged, for example if joint caps are loose, broken or removed.
6. If the software prompts an error, immediately press emergency stop, write down the conditions that led to the error, find the corresponding error codes on the log screen, and contact your supplier.
7. Do not connect any safety equipment to standard I/O. Use safety-related I/O only.
8. Make sure to use the correct installation settings (e.g. Robot mounting angle, mass in TCP, TCP offset, safety configuration). Save and load the installations file along with the program.
9. The freedrive function shall only be used in installations where the risk assessment allows it.
10. Tool/end effectors and obstacles shall not have sharp edges or pinch points.
11. Make sure to warn people to keep their heads and faces outside the reach of the operating robot or robot about to start operating.
12. Be aware of robot movement when using the teach pendant.
13. If determined by the risk assessment, do not enter the safety range of the robot or touch the robot when the system is in operation.
14. Combining different machines can increase hazards or create new hazards. Always make an overall risk assessment for the complete installation. Depending on the assessed risk, different levels of functional safety can apply; as such, when different safety and emergency stop performance levels are needed, always choose the highest performance level. Always read and understand the manuals for all equipment used in the installation.
15. Never modify the robot. A modification might create hazards that are unforeseen by the integrator. All authorized reassembling shall be done according to the newest version of all relevant service manuals.
16. If the robot is purchased with an extra module (e.g. euromap67 interface) then look up that module in the respective manual.



17. Make sure the users of the robot are informed of the location of the emergency stop button(s) and are instructed to activate the emergency stop in case of emergency or abnormal situations.



### WARNING

1. The robot and its controller box generate heat during operation. Do not handle or touch the robot while in operation or immediately after operation as prolonged contact can cause discomfort. You can check the temperature on the log screen before handling or touching the robot, or you can cool down the robot by powering it off and waiting one hour.
2. Never stick fingers behind the internal cover of the controller box.



### CAUTION

1. When the robot is combined, or working, with machines capable of damaging the robot, it is highly recommended to test all functions and the robot program separately.
2. Do not expose the robot to permanent magnetic fields. Very strong magnetic fields can damage the robot.

## 2.6. Intended Use

Universal Robots e-Series are industrial robots intended to handle tools/end effectors and fixtures, or to process or transfer components or products. For details about the environmental conditions under which the robot should operate, see appendices and .

Universal Robots e-Series are equipped with special safety functions, which are purposely designed to enable collaborative operation, where the robot system operates without fences and/or together with a human.

Collaborative operation is only intended for non-hazardous applications, where the complete application, including tool/end effector, work piece, obstacles and other machines, is without any significant hazards according to the risk assessment of the specific application.

Any use or application deviating from intended use is deemed to be impermissible misuse. This includes, but is not limited to:

- Use in potentially explosive environments.
- Use in medical and life critical applications.
- Use in applications where there is direct contact with food, beverage and/or pharmaceutical products.
- Use before performing a risk assessment.
- Use outside of stated specifications.
- Use as a climbing aid.
- Operation outside the permissible operating parameters.



### WARNING

- Only use this industrial robot in accordance with the intended use and specifications provided in the User Manual.
- The product is not designed or intended for use in hazardous locations or explosive environments.
- The product is not designed or intended for medical applications with contact or proximity to patients.
- The product is not designed or intended for use in any application requiring compliance with specific hygienic and/or sanitation standards, such as direct contact with food, beverage and/or pharmaceutical products.
- Any use or application deviating from the intended use, specifications, and certifications is prohibited as the result could be death, personal injury and/or property damage.

UNIVERSAL ROBOTS EXPRESSLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR ANY MISUSE.

## 2.7. Risk Assessment

One of the most important things that an integrator needs to do is to perform a risk assessment. In many countries this is required by law. The robot itself is partly completed machinery, as the safety of the robot installation depends on how the robot is integrated (E.g. tool/end effector, obstacles and other machines). It is recommended that the integrator uses ISO 12100 and ISO 10218-2 to conduct the risk assessment. Additionally the integrator can choose to use the Technical Specification ISO/TS 15066 as additional guidance. The risk assessment that the integrator conducts shall consider all work tasks throughout the lifetime of the robot application, including but not limited to:

- Teaching the robot during set-up and development of the robot installation
- Troubleshooting and maintenance
- Normal operation of the robot installation

A risk assessment must be conducted **before** the robot arm is powered on for the first time. A part of the risk assessment conducted by the integrator is to identify the proper safety configuration settings, as well as the need for additional emergency stop buttons and/or other protective measures required for the specific robot application.

Identifying the correct safety configuration settings is a particularly important part of developing collaborative robot applications. See chapter [3. Safety-related Functions and Interfaces on page 17](#) and part [Part II PolyScope Manual on page 93](#) for detailed information.

Some safety functions are purposely designed for collaborative robot applications. These features are configurable through the safety configuration settings and are particularly relevant when addressing specific risks in the risk assessment conducted by the integrator:

- **Force and power limiting:** Used to reduce clamping forces and pressures exerted by the robot in the direction of movement in case of collisions between the robot and the operator.
- **Momentum limiting:** Used to reduce high transient energy and impact forces in case of collisions between robot and operator by reducing the speed of the robot.
- **Joint, elbow and tool/end effector position limiting:** Particularly used to reduce risks associated with certain body parts. E.g. to avoid movement towards head and neck.
- **Tool/end effector orientation limiting:** Particularly used to reduce risks associated with certain areas and features of the tool/end effector and work-piece. E.g. to avoid sharp edges to be pointed towards the operator.
- **Speed limitation:** Particularly used to ensure a low speed of the robot arm.

The integrator must prevent unauthorized access to the safety configuration by using password protection.

A collaborative robot application risk assessment for contacts that are intentional and/or due to reasonably foreseeable misuse is required and must address:

- Severity of individual potential collisions
- Likeliness of occurrence of individual potential collisions
- Possibility to avoid individual potential collisions

If the robot is installed in a non-collaborative robot application where hazards cannot be reasonably eliminated or risks cannot be sufficiently reduced by use of the built-in safety-related functions (e.g. when using a hazardous tool/end effector), then the risk assessment conducted by the integrator must conclude the need for additional protective measures (e.g. an enabling device to protect the operator during set-up and programming).

Universal Robots identifies the potential significant hazards listed below as hazards that must be considered by the integrator. Other significant hazards can be present in a specific robot installation.

1. Penetration of skin by sharp edges and sharp points on tool/end effector or tool/end effector connector.
2. Penetration of skin by sharp edges and sharp points on obstacles near the robot track.
3. Bruising due to contact with the robot.
4. Sprain or bone fracture due to strokes between a heavy payload and a hard surface.
5. Consequences due to loose bolts that hold the robot arm or tool/end effector.
6. Items falling out of tool/end effector, e.g. due to a poor grip or power interruption.
7. Mistakes due to different emergency stop buttons for different machines.
8. Mistakes due to unauthorized changes to the safety configuration parameters.

Information on stopping times and stopping distances are found in chapter [3. Safety-related Functions and Interfaces](#) on page 17 and appendix .

## 2.8. Pre-Use Assessment

The following tests must be conducted before using the robot for the first time or after any modifications are made. Verify that all safety input and output are appropriately and correctly connected. Test that all connected safety input and output, including devices common to multiple machines or robots, are functioning. As such you must:

- Test that emergency stop buttons and input stop the robot and engage brakes.
- Test that safeguard input stop the robot motion. If safeguard reset is configured, check that it needs to be activated before motion can be resumed.
- Examine the initialization screen to test that reduced mode can switch the safety mode to reduced mode.
- Test that the operational mode switches the operational mode, see icon in top right corner of user interface.
- Test that the 3-position enabling device must be pressed to enable motion in manual mode and that the robot is under reduced speed control.
- Test that System Emergency Stop outputs are actually capable of bringing the whole system to a safe state.
- Test that the system connected to Robot Moving output, Robot Not Stopping output, Reduced Mode output, or Not Reduced Mode output can actually detect the output changes

## 2.9. Emergency Stop

Activate the emergency stop push-button to immediately stop all robot motion.

According to IEC 60204-1 and ISO 13850, emergency devices are not safeguards. They are complimentary protective measures and are not intended to prevent injury.

The risk assessment of the robot application shall conclude if additional emergency stop buttons are needed. Emergency stop push-buttons must comply with IEC 60947-5-5 (see section ).

## 2.10. Movement Without Drive Power

In the unlikely event of an emergency, you can use forced back-driving where you must move the robot joint/s, but robot power is either impossible or unwanted.

To perform forced back-driving you must push, or pull, the robot arm hard to move the joint. Each joint brake has a friction clutch that enables movement during the high forced torque.

Forced back-driving is only intended for emergency purposes.



# 3. Safety-related Functions and Interfaces

## 3.1. Introduction

Universal Robots e-Series robots are equipped with a range of built-in safety functions as well as safety I/O, digital and analog control signals to or from the electrical interface, to connect to other machines and additional protective devices. Each safety function and I/O is constructed according to EN ISO13849-1 (see chapter [9. Certifications on page 59](#) for certifications) with Performance Level d (PLd) using a category 3 architecture.

See chapter [21. Safety Configuration on page 115](#) in part [Part II PolyScope Manual on page 93](#) for configuration of the safety functions, inputs and outputs in the user interface. See chapter [9. Certifications on page 59](#) for descriptions on how to connect safety devices to I/O.



### NOTE

1. The use and configuration of safety functions and interfaces must follow the risk assessment procedures for each robot application. (see chapter [2. Safety on page 7](#) section [2.7. Risk Assessment on page 13](#))
2. If the robot discovers a fault or violation in the safety system (e.g. if one of the wires in the Emergency Stop circuit is cut or a safety limit is exceeded) then a Stop Category 0 is initiated.
3. The stopping time should be taken into account as part of the application risk assessment



### WARNING

1. The use of safety configuration parameters different from those determined by the risk assessment can result in hazards that are not reasonably eliminated or risks that are not sufficiently reduced
2. Ensure tools and grippers are connected appropriately so if there is an interruption of power, no hazards occur!
3. Use caution with 12V, since an error made by the programmer can cause the voltage to change to 24V, which might damage the equipment and cause a fire
4. The end effector is not protected by the UR safety system. The functioning of the end effector and/or connection cable is not monitored

## 3.2. Stop Categories

Depending on the circumstances, the robot can initiate three types of stop categories defined according to IEC 60204-1. These categories are defined in the following table.

Stop Category	Description
0	Stop the robot by immediate removal of power.
1	Stop the robot in an orderly, controlled manner. Power is removed once the robot is stopped.
2	*Stop the robot with power available to the drives, while maintaining the trajectory. Drive power is maintained after the robot is stopped.

\*Universal Robots robots' Category 2 stops are further described as SS1 or as SS2 type stops according to IEC 61800-5-2.

### 3.3. Configurable Safety Functions

Universal Robots robot safety functions, as listed in the table below, are in the robot but are meant to control the robot system i.e. the robot with its attached tool/end effector. The robot safety functions are used to reduce robot system risks determined by the risk assessment. Positions and speeds are relative to the base of the robot.

Safety Function	Description
Joint Position Limit	Sets upper and lower limits for the allowed joint positions.
Joint Speed Limit	Sets an upper limit for joint speed.
Safety Planes	Defines planes, in space, that limit robot position. Safety planes limit either the tool/end effector alone or both the tool/end effector and the elbow.
Tool Orientation	Defines allowable orientation limits for the tool.
Speed Limit	Limits maximum robot speed. The speed is limited at the elbow, at the tool/end effector flange, and at the center of the user-defined tool/end effector positions.
Force Limit	Limits maximum force exerted by the robot tool/end effector and elbow in clamping situations. The force is limited at the tool/end effector, elbow flange and center of the user-defined tool/end effector positions.
Momentum Limit	Limits maximum momentum of the robot.
Power Limit	Limits mechanical work performed by the robot.
Stopping Time Limit	Limits maximum time the robot uses for stopping after a protective stop is initiated.

Safety Function	Description
Stopping Distance Limit	Limits maximum distance travelled by the robot after a protective stop is initiated.

When performing the application risk assessment, it is necessary to take into account the motion of the robot after a stop has been initiated. In order to ease this process, the safety functions *Stopping Time Limit* and *Stopping Distance Limit* can be used.

These safety functions dynamically reduces the speed of the robot motion such that it can always be stopped within the limits. The joint position limits, the safety planes and the tool/end effector orientation limits take the expected stopping distance travel into account i.e. the robot motion will slow down before the limit is reached.

The functional safety can be summarized as:

Safety Function	Tolerance	Performance Level	Category
Emergency Stop	-	d	3
Safeguard Stop	-	d	3
Joint Position Limit	5 °	d	3
Joint Speed Limit	1.15 °/s	d	3
Safety Planes	40 mm	d	3
Tool Orientation	3 °	d	3
Speed Limit	50 mm/s	d	3
Force Limit	25 N	d	3
Momentum Limit	3 kg m/s	d	3
Power Limit	10 W	d	3
Stopping Time Limit	50 ms	d	3
Stopping Distance Limit	40 mm	d	3
Safe Home	1.7 °	d	3

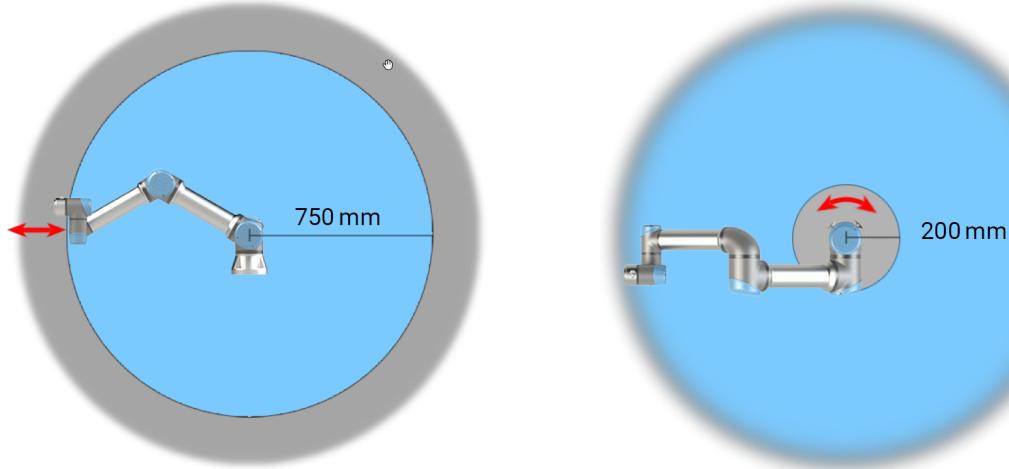


### WARNING

There are two exceptions to the force limiting function that are important when designing an application (Figure 4.1). As the robot stretches out, the knee-joint effect can give high forces in the radial direction (away from the base) at low speeds. Similarly, the short leverage arm, when the tool/end effector is close to the base and moving around the base, can cause high forces at low speeds. Pinching hazards can be avoided by removing obstacles in these areas, placing the robot differently, or by using a combination of safety planes and joint limits to eliminate the hazard by preventing the robot moving into this region of its workspace.

**WARNING**

If the robot is used in manual hand-guiding applications with linear movements, the speed limit must be set to maximum 250 mm/s for the tool/end effector and elbow unless a risk assessment shows that higher speeds are acceptable. This will prevent fast movements of the robot elbow near singularities.



**4.1:** Due to the physical properties of the robot arm, certain workspace areas require attention regarding pinching hazards. One area (left) is defined for radial motions when the wrist 1 joint is at least 750 mm from the base of the robot. The other area (right) is within 200 mm of the base of the robot, when moving tangentially.

The robot also has the following safety inputs:

Safety Input	Description
Emergency Stop Button	Performs a Stop Category 1 (IEC 60204-1) informing other machines using the <i>System Emergency Stop</i> 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 <i>System Emergency Stop</i> output, if that output is defined.
System Emergency Stop	Performs a Stop Category 1 (IEC 60204-1) on robot only, in all modes and takes precedence over all other commands.
Safeguard Stop	Performs a Stop Category 2 (IEC 60204-1) in all modes, except when using a 3-Position Enabling Device and a mode selector - then when in Manual Mode, the Safeguard Stop can be set to only function in Automatic Mode.

Safety Input	Description
Automatic Mode Safeguard Stop	Performs a Stop Category 2 (IEC 60204-1) in Automatic mode ONLY. <i>Automatic Mode Safeguard Stop</i> can only be selected when a Three-Position Enabling Device is configured and installed.
Safeguard Reset	Returns from the <i>Safeguard Stop</i> state, when a rising edge on the Safeguard Reset input occurs.
Reduced Mode	Transitions the safety system to use the <i>Reduced mode</i> limits.
Three-Position Enabling Device	Initiates a Stop Category 2 (IEC 60204-1) when the enabling device is fully pressed or fully released in manual mode only. Three-Position Enabling Device Stop is triggered when an input goes low. It is unaffected by a Safeguard Reset.
Operational Mode	Switches between Operational modes. The robot is in Automatic mode when input is low, Manual mode when input is high.
Automatic Mode Safeguard Reset	Returns from the <i>Automatic Mode Safeguard Stop</i> state, when a rising edge on the Automatic Mode Safeguard Reset input occurs.

For interfacing with other machines, the robot is equipped with the following safety outputs:

Safety Output	Description
System Emergency Stop	While this signal is logic low, the <i>Robot Emergency Stop</i> input is logic low or the Emergency Stop button is pressed.
Robot Moving	While this signal is logic high, no single joint of the robot moves more than 0.1rad/s.
Robot Not Stopping	Logic 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 Mode	Logic low when the safety system is in Reduced Mode.
Not Reduced Mode	Logic low when the system is not in Reduced Mode.
Safe Home	Logic high when robot is in the configured Safe Home Position.

All safety I/O are dual channel, meaning they are safe when low (e.g., the Emergency Stop is active when the signals are low).

## 3.4. Safety Function

The safety system acts by monitoring if any of the safety limits are exceeded or if an Emergency Stop or a Safeguard Stop is initiated.

The reactions of the safety system are:

Trigger	Reaction
Emergency Stop	Stop Category 1.
Safeguard Stop	Stop Category 2.
Limit Violation	Stop Category 0.
Fault Detection	Stop Category 0.



#### NOTE

If the safety system detects any fault or violation, all safety outputs re-set to low.

## 3.5. Modes

### Normal and Reduced Mode

The safety system has two configurable Modes: **Normal** and **Reduced**. Safety limits can be configured for each of these two modes. Reduced Mode is active when the robot tool/end effector is positioned on the Reduced Mode side of a **Trigger Reduced Mode** plane or when triggered by a safety input.

**Using a plane to trigger Reduced Mode:** When the robot moves from the Reduced Mode side of the trigger plane, back to the Normal Mode side, there is a 20mm area around the trigger plane where both Normal and Reduced Mode limits are allowed. It prevents the Safety Mode from flickering if the robot is right at the limit.

**Using an input to trigger Reduced Mode:** When an input is used (to either start or stop Reduced Mode), up to 500ms can elapse before the new mode limit values are applied. This could happen either when changing Reduced Mode to Normal Mode OR changing Normal Mode to Reduced Mode. It allows the robot to adapt e.g. the speed to the new safety limits.

### Recovery Mode

When a safety limit is exceeded, the safety system must be restarted. If the system is outside a safety limit at start-up (e.g.outside a joint position limit), the special Recovery Mode is entered. In Recovery Mode, it is not possible to run programs for the robot, but the robot arm can be manually moved back within limits either by using Freedrive Mode or by using the Move tab in PolyScope (see part [Part II PolyScope Manual on page 93 PolyScope Manual](#) ). The safety limits of Recovery Mode are:

Safety Function	Limit
Joint Speed Limit	30 °/s
Speed Limit	250 mm/s
Force Limit	100 N
Momentum Limit	10 kg m/s
Power Limit	80 W

The safety system issues a Stop Category 0 if a violation of these limits appears.

**WARNING**

Limits for the joint positions, the safety planes, and the tool/end effector orientation are disabled in Recovery Mode. Take caution when moving the robot arm back within the limits.



# 4. Transportation

As supplied on the pallet, the robot and Control Box are a calibrated set. Do not separate them as this would require recalibration.

Only transport the robot in its original packaging. Save the packaging material in a dry place if you want to move the robot later.

When moving the robot from its packaging to the installation space, hold both tubes of the robot arm at the same time. Hold the robot in place until all mounting bolts are securely tightened at the base of the robot.

Lift the Control Box by its handle.



## WARNING

1. Make sure not to overload your back or other bodyparts when lifting the equipment. Use proper lifting equipment. All regional and national lifting guidelines shall be followed. Universal Robots cannot be held responsible for any damage caused by transportation of the equipment.
2. Make sure to mount the robot according to the instructions in chapter [5. Mechanical Interface](#) on page 27.



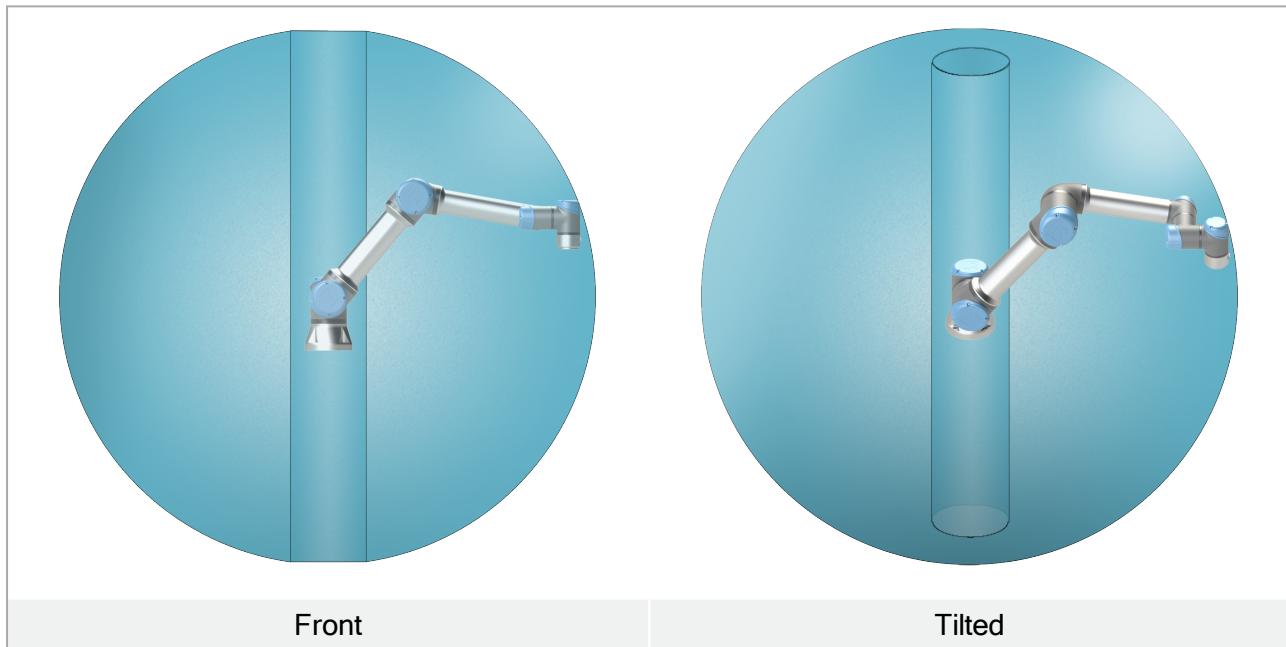
# 5. Mechanical Interface

## 5.1. Introduction

This chapter describes the basics of mounting the parts of the robot system. Electrical installation instructions in chapter must be observed.

## 5.2. Workspace of the Robot

The workspace of the robot extends 850 mm from the base joint. It is important to consider the cylindrical volume directly above and directly below the robot base when choosing a mounting place for the robot. Moving the tool close to the cylindrical volume should be avoided because it causes the joints to move fast even when the tool is moving slowly, which causes the robot to work inefficiently and makes it difficult to conduct a risk assessment.



## 5.3. Mounting

### Robot Arm

The robot arm is mounted with four 8.8 strength, M8 bolts and four 8.5 mm mounting holes at the base.



#### WARNING

Make sure the robot arm is properly and securely bolted in place. Unstable mounting can lead to accidents.

## Securing the Robot Arm

Figure 6.1 shows where to drill holes and mount the screws. You can purchase an accurate base counterpart as an accessory.

1. Mount the robot on a sturdy, vibration-less, surface that can withstand at least ten times the full torque of the base joint and at least five times the weight of the robot arm.  
If the robot is mounted on a linear axis, or a moving platform, then the acceleration of the moving mounting base is very low. High acceleration might cause the robot to make a safety stop.
2. Tighten the bolts to 10 Nm torque.
3. Use the two Ø8 holes provided, with a pin, to accurately reposition robot arm.



### WARNING

Power down the robot arm to prevent unexpected start-up during mounting and dismounting.

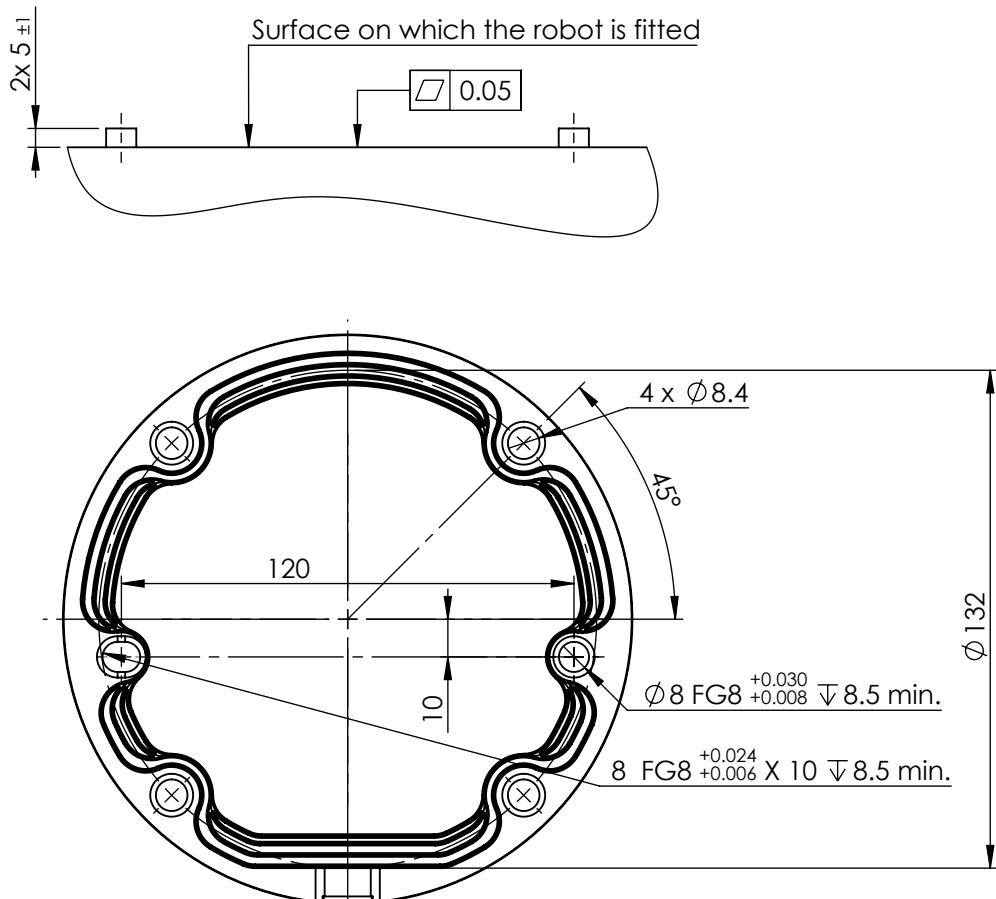
To power down the robot arm:

1. Press the power button on the Teach Pendant to turn off the robot.
2. Unplug the mains cable / power cord from the wall socket.
3. Allow 30 seconds for the robot to discharge any stored energy.



### CAUTION

Mount the robot in an environment suited to the IP rating. The robot must not be operated in environments that exceed those corresponding to the IP ratings of the robot (IP54), Teach Pendant (IP54) and Control Box (IP44)



6.1: Holes for mounting the robot. Use four M8 bolts. All measurements are in mm .

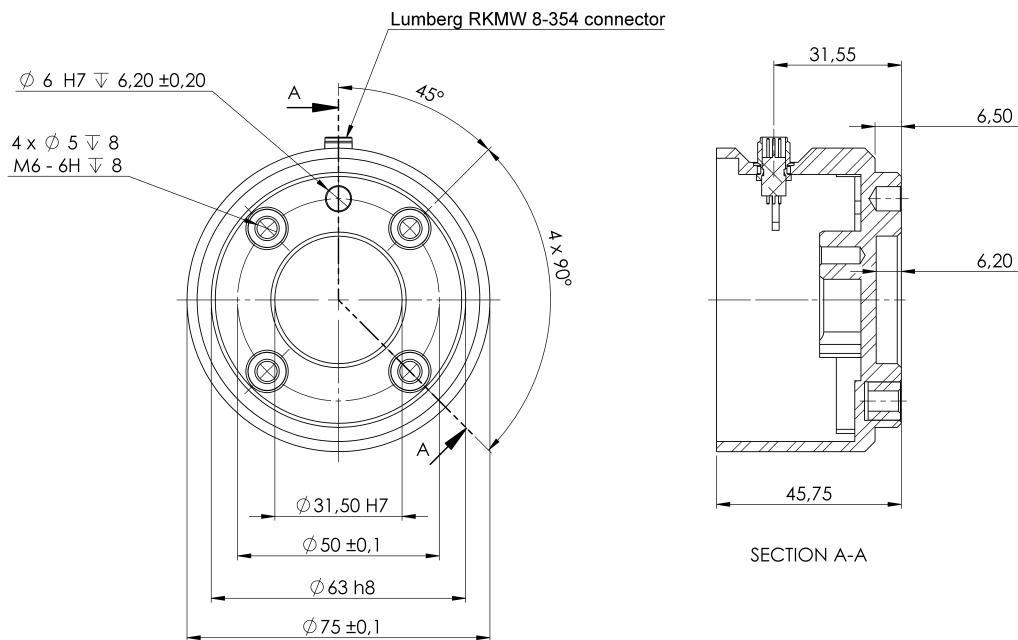
## Tool

The robot tool flange has four M6 thread holes for attaching a tool to the robot. The M6 bolts must be tightened with 8 N m, strength class 8.8. For accurate tool repositioning, use a pin in the Ø6 hole provided. Figure 6.2 shows dimensions and hole pattern of the tool flange. It is recommended to use a radially slotted hole for the pin to avoid over-constraining, while keeping precise position. Do not use bolts that extend beyond 8mm to mount the tool. Very long M6 bolts can press against the bottom of the tool flange and short circuit the robot.



### WARNING

1. Ensure the tool is properly and securely bolted in place.
2. Ensure the tool is constructed such that it cannot create a hazardous situation by dropping a part unexpectedly.
3. Mounting a tool on the robot with M6 bolts that extend beyond 8mm can push into the tool flange and cause irreparable damage, leading to end joint replacement.



6.2: The tool output flange (ISO 9409-1-50-4-M6) is where the tool is mounted at the tip of the robot.  
All measures are in mm.

## Control Box

The Control Box can be hung on a wall or placed on the ground. A clearance of 50 mm on each side of the Control Box is needed for sufficient airflow.

## Teach Pendant

The Teach Pendant can be hung on a wall or on the Control Box. Verify that the cable does not cause tripping hazard.

You can buy extra brackets for mounting the Control Box and Teach Pendant.



### WARNING

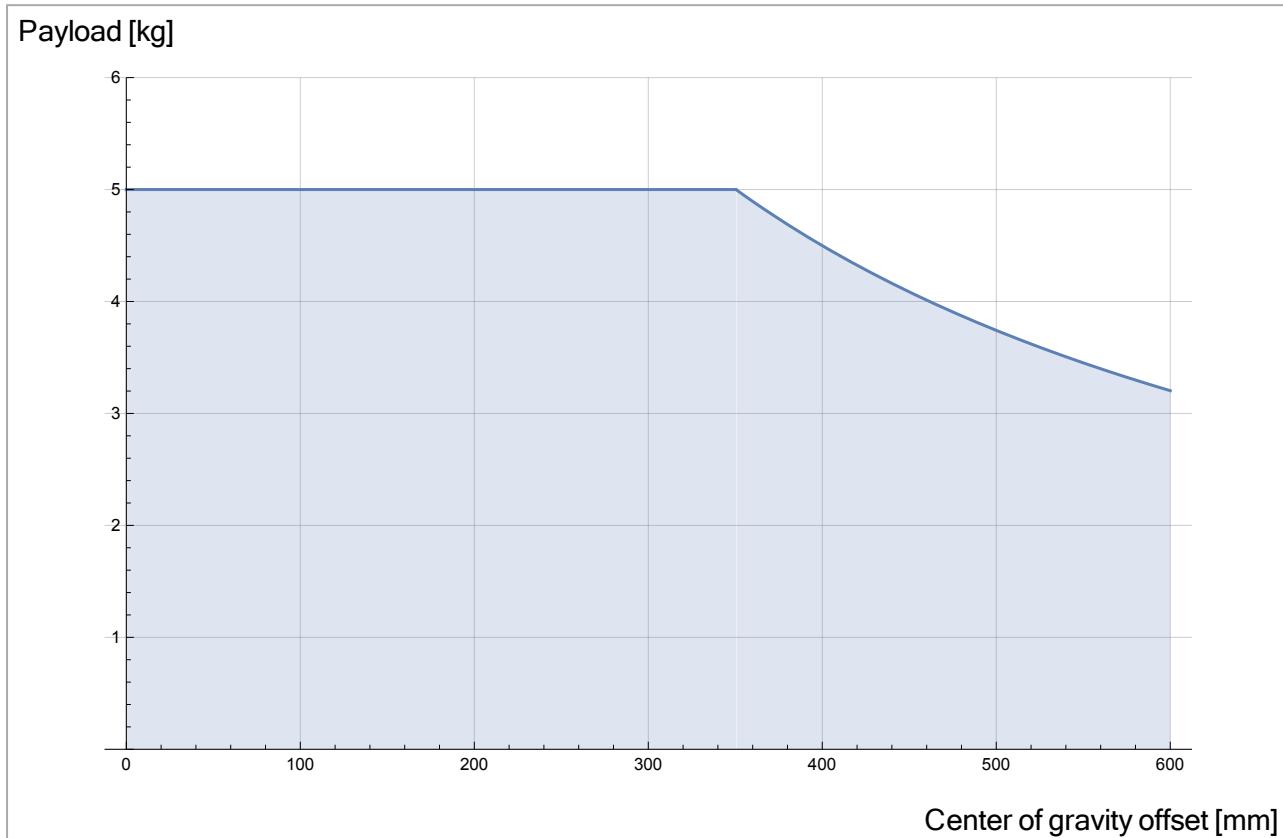
1. Make sure the Control Box, Teach Pendant and cables do not come into contact with liquids. A wet Control Box could cause fatal injury.
2. Place the Teach Pendant (IP54) and Control Box (IP44) in an environment suited for the IP rating.

## 5.4. Maximum Payload

The rated payload of the Robot Arm depends on the *center of gravity offset* of the payload, see Figure 6.3.

The center of gravity offset is defined as the distance from the center of the tool flange to the center of gravity of the attached payload.

When computing the payload mass in a pick and place application, for example, consider both the gripper and object handled by the gripper.



6.3: *The relationship between the rated payload and the center of gravity offset.*

# 6. Electrical Interface

## 6.1. Introduction

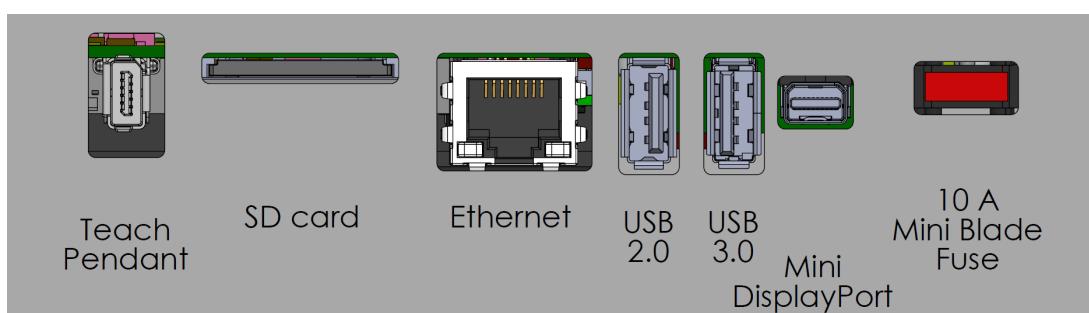
This chapter describes electrical interface groups for the Robot Arm in the Control Box. Examples are given for most types of I/O. The term I/O refers to both digital and analog control signals to or from the electrical interface groups listed below.

- Mains connection
- Robot connection
- Controller I/O
- Tool I/O
- Ethernet

### 6.1.1. Control Box Bracket

On the underside of the I/O interface groups, there is a bracket with ports that allows for additional connections (illustrated below). The base of the Control Box has a capped opening for easy connection (see [6.2. Ethernet below](#)).

The Mini Displayport supports monitors with Displayport and requires an active Mini Display to DVI or HDMI converter to connect monitors with DVI/HDMI interface. Passive converters do not work with DVI/HDMI ports.



The Fuse must be UL marked, Mini Blade type with maximum current rating: 10A and minimum voltage rating: 32V

## 6.2. Ethernet

The Ethernet interface can be used for:

- MODBUS, EtherNet/IP and PROFINET (see part [Part II PolyScope Manual on page 93](#)).
- Remote access and control.

To connect the Ethernet cable by passing it through the hole at the base of the Control Box, and plugging it into the Ethernet port on the underside of the bracket.

Replace the cap at the base of the Control Box with an appropriate cable gland to connect the cable to the Ethernet port.



The electrical specifications are shown in the table below.

Parameter	Min	Typ	Max	Unit
Communication speed	10	-	1000	Mb/s

## 6.3. Electrical Warnings and Cautions

Observe the following warnings for all the aforementioned interface groups, in addition to when the robot application is designed and installed.

All voltages and currents are in Direct Current (DC) unless otherwise specified.



### WARNING

1. Never connect safety signals to a PLC which is not a safety PLC with the correct safety level. Failure to follow this warning could result in serious injury or death as the safety functions could be overridden. It is important to keep safety interface signals separated from the normal I/O interface signals.
2. All safety-related signals are constructed redundantly (two independent channels). Keep the two channels separate so that a single fault cannot lead to loss of the safety function.
3. Some I/Os inside the Control Box can be configured for either normal or safety-related I/O. Read and understand the complete section [6.4. Controller I/O on the next page](#).



### WARNING

1. Make sure all equipment not rated for water exposure remain dry. If water is allowed to enter the product, lockout-tagout all power and then contact your local Universal Robots service provider for assistance.
2. Only use the original cables supplied with the robot only. Do not use the robot for applications where the cables are subject to flexing.
3. Negative connections are referred to as Ground (GND) and are connected to the casing of the robot and the Control Box. All mentioned GND connections are only for powering and signalling. For PE (Protective Earth) use the M6-size screw connections marked with earth symbols inside the Control Box. The grounding conductor shall have at least the current rating of the highest current in the system.
4. Use caution when installing interface cables to the robot I/O. The metal plate in the bottom is intended for interface cables and connectors. Remove the plate before drilling holes. Make sure that all shavings are removed before reinstalling the plate. Remember to use correct gland sizes.



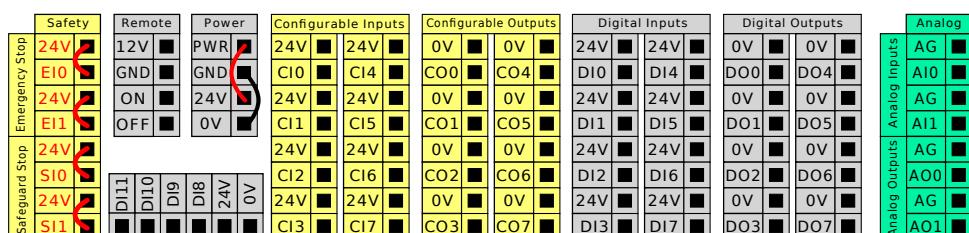
### CAUTION

1. The robot has been tested according to international IEC standards for **ElectroMagnetic Compatibility (EMC)**. Disturbing signals with levels higher than those defined in the specific IEC standards can cause unexpected behaviors from the robot. Very high signal levels or excessive exposure can damage the robot permanently. EMC problems are found to happen usually in welding processes and are normally prompted by error messages in the log. Universal Robots cannot be held responsible for any damages caused by EMC problems.
2. I/O cables going from the Control Box to other machinery and factory equipment may not be longer than 30m, unless additional tests are performed.

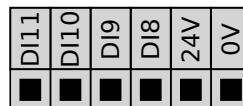
## 6.4. Controller I/O

You can use the I/O inside the Control Box for a wide range of equipment including pneumatic relays, PLCs and emergency stop buttons.

The illustration below shows the layout of electrical interface groups inside the Control Box.



You can use the horizontal Digital Inputs block (DI8-DI11), illustrated below, for quadrature encoding Conveyor Tracking (see [6.4.1. Common specifications for all digital I/O below](#)) for these types of input.



The meaning of the color schemes listed below must be observed and maintained.

Yellow with red text	Dedicated safety signals
Yellow with black text	Configurable for safety
Gray with black text	General purpose digital I/O
Green with black text	General purpose analog I/O

In the GUI, you can set up **configurable I/O** as either **safety-related I/O** or **general purpose I/O** (see part [Part II PolyScope Manual on page 93](#)).

## 6.4.1. Common specifications for all digital I/O

This section defines electrical specifications for the following 24V digital I/O of the Control Box.

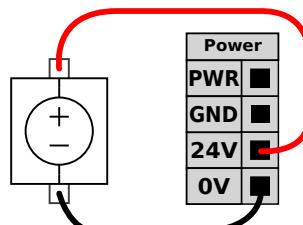
- Safety I/O.
- Configurable I/O.
- General purpose I/O.

Install the robot according to the electrical specifications which are the same for all three inputs.

It is possible to power the digital I/O from an internal 24V power supply or from an external power source by configuring the terminal block called **Power**. This block consists of four terminals. The upper two (PWR and GND) are 24V and ground from the internal 24V supply. The lower two terminals (24V and 0V) in the block are the 24V input to supply the I/O. The default configuration uses the internal power supply (see below).



If more current is needed, connect an external power supply as shown below.



The electrical specifications for both the internal and external power supply are shown below.

Terminals	Parameter	Min	Typ	Max	Unit
<i>Internal 24V power supply</i>					
[PWR - GND]	Voltage	23	24	25	V
[PWR - GND]	Current	0	-	2*	A
<i>External 24V input requirements</i>					
[24V - 0V]	Voltage	20	24	29	V
[24V - 0V]	Current	0	-	6	A

\*3.5A for 500ms or 33% duty cycle.

The digital I/O are constructed in compliance with IEC 61131-2. The electrical specifications are shown below.

Terminals	Parameter	Min	Typ	Max	Unit
<i>Digital Outputs</i>					
[CO <sub>x</sub> / DO <sub>x</sub> ]	Current*	0	-	1	A
[CO <sub>x</sub> / DO <sub>x</sub> ]	Voltage drop	0	-	0.5	V
[CO <sub>x</sub> / DO <sub>x</sub> ]	Leakage current	0	-	0.1	mA
[CO <sub>x</sub> / DO <sub>x</sub> ]	Function	-	PNP	-	Type
[CO <sub>x</sub> / DO <sub>x</sub> ]	IEC 61131-2	-	1A	-	Type
<i>Digital Inputs</i>					
[EI <sub>x</sub> /SI <sub>x</sub> /CI <sub>x</sub> /DI <sub>x</sub> ]	Voltage	-3	-	30	V
[EI <sub>x</sub> /SI <sub>x</sub> /CI <sub>x</sub> /DI <sub>x</sub> ]	OFF region	-3	-	5	V
[EI <sub>x</sub> /SI <sub>x</sub> /CI <sub>x</sub> /DI <sub>x</sub> ]	ON region	11	-	30	V
[EI <sub>x</sub> /SI <sub>x</sub> /CI <sub>x</sub> /DI <sub>x</sub> ]	Current (11-30V)	2	-	15	mA
[EI <sub>x</sub> /SI <sub>x</sub> /CI <sub>x</sub> /DI <sub>x</sub> ]	Function	-	PNP +	-	Type
[EI <sub>x</sub> /SI <sub>x</sub> /CI <sub>x</sub> /DI <sub>x</sub> ]	IEC 61131-2	-	3	-	Type

\*For resistive loads or inductive loads of maximum 1H.



#### NOTE

The word **configurable** is used for I/O configured as either safety-related I/O or normal I/O. These are the yellow terminals with black text.

## 6.4.2. Safety I/O

This section describes dedicated safety input (Yellow terminal with red text) and configurable I/O (Yellow terminals with black text) when configured as safety I/O. Follow the Common specifications for all digital I/O in section [6.4.1. Common specifications for all digital I/O on the previous page](#).

Safety devices and equipment must be installed according to the safety instructions and the risk assessment in chapter [2. Safety on page 7](#).

All safety I/O are paired (redundant) and must be kept as two separate branches. A single fault does not cause loss of the safety function.

There are two permanent safety input types:

- **Robot Emergency Stop** for emergency stop equipment only
- **Safeguard Stop** for protective devices

The functional difference is shown below.

	Emergency Stop	Safeguard Stop
Robot stops moving	Yes	Yes
Program execution	Pauses	Pauses
Drive power	Off	On
Reset	Manual	Automatic or manual
Frequency of use	Infrequent	Every cycle to infrequent
Requires re-initialization	Brake release only	No
Stop Category (IEC 60204-1)	1	2
Performance level of monitoring function (ISO 13849-1)	PLd	PLd

Use the configurable I/O to set up additional safety I/O functionality, e.g. Emergency Stop Output. Configuring a set of configurable I/O for safety functions are done through the GUI, (see part [Part II PolyScope Manual on page 93](#)).



### WARNING

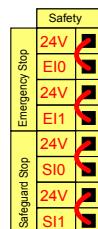
1. Never connect safety signals to a PLC that is not a safety PLC with the correct safety level. Failure to follow this warning could result in serious injury or death as the safety functions could be overridden. It is important to keep safety interface signals separated from the normal I/O interface signals.
2. All safety-related I/O are constructed redundantly (two independent channels). Keep the two channels separate so that a single fault cannot lead to loss of the safety function.
3. Safety functions must be verified before putting the robot into operation. Safety functions must be tested regularly.
4. The robot installation shall conform to these specifications. Failure to do so could result in serious injury or death as the safety function could be overridden.

## OSSD Signals

All configured and permanent safety inputs are filtered to allow the use of OSSD safety equipment with pulse lengths under 3ms. The safety input is sampled every millisecond and the state of the input is determined by the most frequently seen input signal over the last 7 milliseconds. OSSD pulses on the safety outputs are detailed in part [Part II PolyScope Manual on page 93](#).

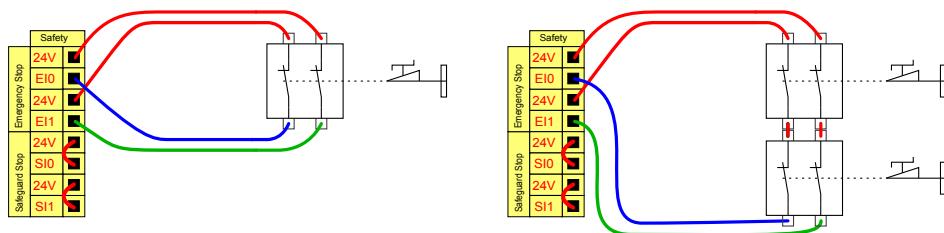
## Default safety configuration

The robot is delivered with a default configuration, which enables operation without any additional safety equipment (see illustration below).



## Connecting emergency stop buttons

Most applications require one or more extra emergency stop buttons. The illustration below shows how one or more emergency stop buttons can be connected.

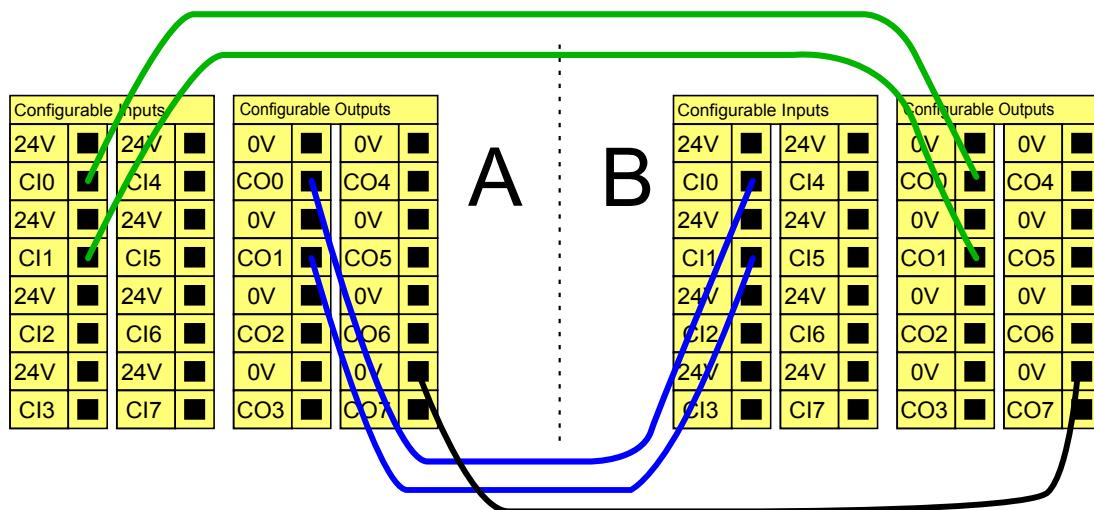


## Sharing the Emergency Stop with other machines

You can set up a shared emergency stop function between the robot and other machines by configuring the following I/O functions via the GUI. The Robot Emergency Stop Input cannot be used for sharing purposes. If more than two UR robots or other machines need to be connected, a safety PLC must be used to control the emergency stop signals.

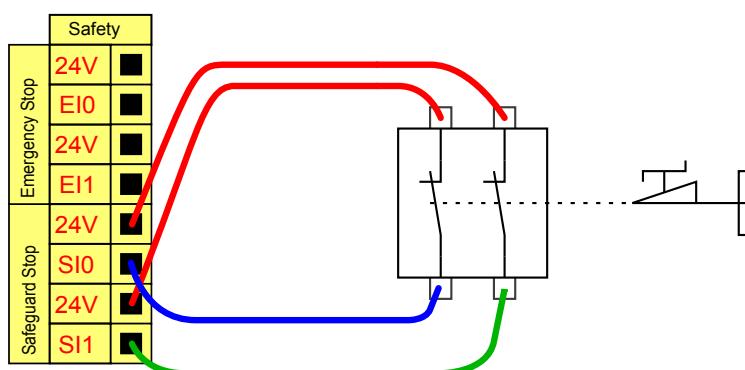
- Configurable input pair: External emergency stop.
- Configurable output pair: System emergency stop.

The illustration below shows how two UR robots share their emergency stop functions. In this example the configured I/Os used are CI0-CI1 and CO0-CO1.



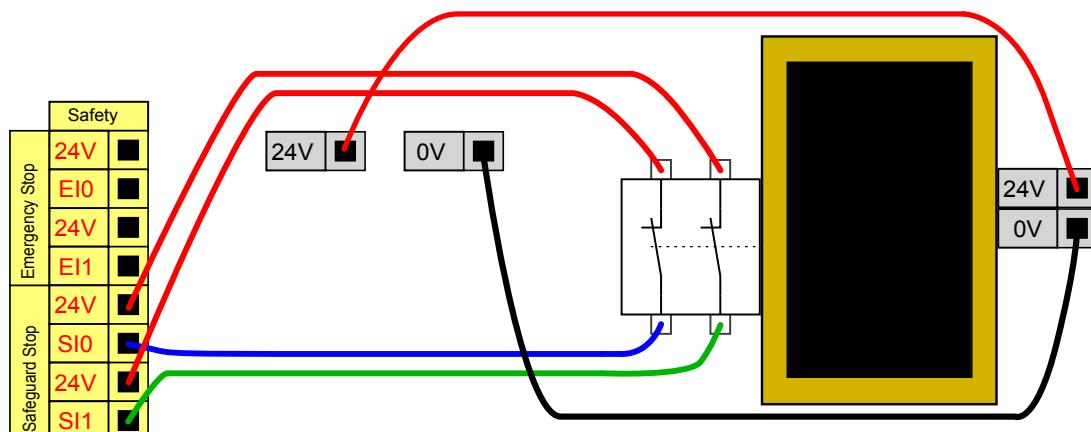
## Safeguard stop with automatic resume

An example of a basic safeguard stop device is a door switch where the robot is stopped when a door is opened (see illustration below).



This configuration is only intended for applications where the operator cannot go through the door and close it behind him. The configurable I/O is used to setup a reset button outside the door to reactivate robot motion.

Another example where automatic resume is appropriate is when using a safety mat or a safety-related laser scanner (see below).

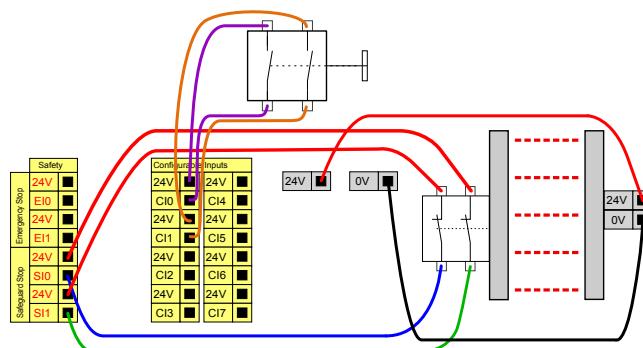


**WARNING**

1. The robot resumes movement automatically when the signal is re-established.  
Do not use this configuration if signal can be re-established from the inside of the safety perimeter.

## Safeguard Stop with reset button

If the safeguard interface is used to interact with a light curtain, a reset outside the safety perimeter is required. The reset button must be a two channel type. In this example the I/O configured for reset is C10-C11 (see below).

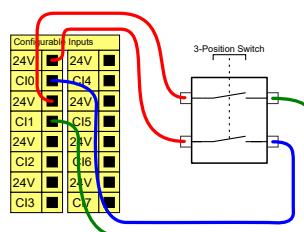


## Three-Position Enabling Device

The illustration below shows how to connect a Three-Position Enabling Device. See section [20. Operational Mode Selection on page 111](#) for more about Three-Position Enabling Device.

**NOTE**

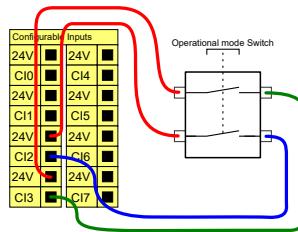
The Universal Robots safety system does not support multiple external Three-Position Enabling Devices.

**NOTE**

The two input channels for the Three-Position Enabling Device input have a disagreement tolerance of 1 second.

## Operational Mode Switch

The illustration below shows an Operational Mode Switch. See section [20.1. Operational Modes on page 111](#) for more about operational Modes.



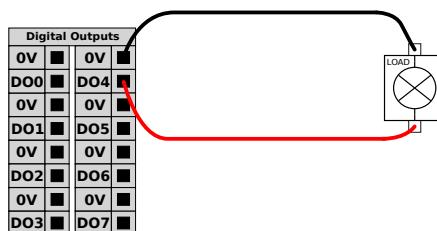
### 6.4.3. General purpose digital I/O

This section describes the general purpose 24V I/O (Gray terminals) and the configurable I/O (Yellow terminals with black text) when not configured as safety I/O. The common specifications in section [6.4.1. Common specifications for all digital I/O on page 35](#) must be observed.

The general purpose I/O can be used to drive equipment like pneumatic relays directly or for communication with other PLC systems. All Digital Outputs can be disabled automatically when program execution is stopped, see part [Part II PolyScope Manual on page 93](#). In this mode, the output is always low when a program is not running. Examples are shown in the following subsections. These examples use regular Digital Outputs but any configurable outputs could also have been used if they are not configured to perform a safety function.

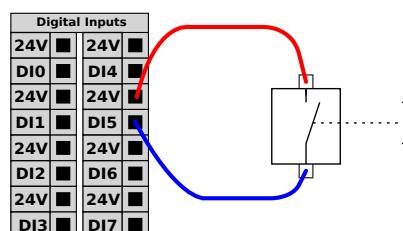
### Load controlled by a Digital Outputs

This example shows how a load is controlled from a Digital Outputs when connected.



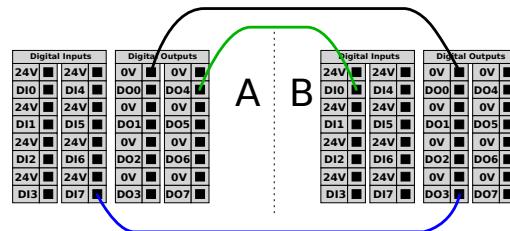
### 6.4.4. Digital Inputs from a button

This example illustrates connecting a simple button to a Digital Inputs.



## 6.4.5. Communication with other machines or PLCs

You can use the digital I/O to communicate with other equipment if a common GND (0V) is established and if the machine uses PNP technology, see below.



## 6.4.6. General purpose analog I/O

The analog I/O interface is the green terminal. It is used to set or measure voltage (0-10V) or current (4-20mA) to and from other equipment.

The following directions is recommended to achieve the highest accuracy.

- Use the AG terminal closest to the I/O. The pair share a common mode filter.
- Use the same GND (0V) for equipment and Control Box. The analog I/O is not galvanically isolated from the Control Box.
- Use a shielded cable or twisted pairs. Connect the shield to the GND terminal at the terminal called **Power**.
- Use equipment that works in current mode. Current signals are less sensitive to interferences.

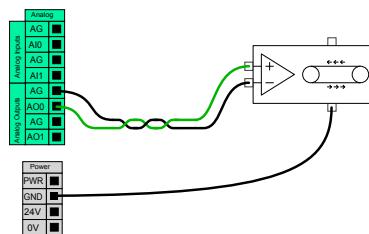
In the GUI you can select input modes (see part [Part II PolyScope Manual on page 93](#)). The electrical specifications are shown below.

Terminals	Parameter	Min	Typ	Max	Unit
<i>Analog Input in current mode</i>					
[AIx - AG]	Current	4	-	20	mA
[AIx - AG]	Resistance	-	20	-	ohm
[AIx - AG]	Resolution	-	12	-	bit
<i>Analog Input in voltage mode</i>					
[AIx - AG]	Voltage	0	-	10	V
[AIx - AG]	Resistance	-	10	-	Kohm
[AIx - AG]	Resolution	-	12	-	bit
<i>Analog Output in current mode</i>					
[AOx - AG]	Current	4	-	20	mA
[AOx - AG]	Voltage	0	-	24	V
[AOx - AG]	Resolution	-	12	-	bit
<i>Analog Output in voltage mode</i>					

Terminals	Parameter	Min	Typ	Max	Unit
[AOx - AG]	Voltage	0	-	10	V
[AOx - AG]	Current	-20	-	20	mA
[AOx - AG]	Resistance	-	1	-	ohm
[AOx - AG]	Resolution	-	12	-	bit

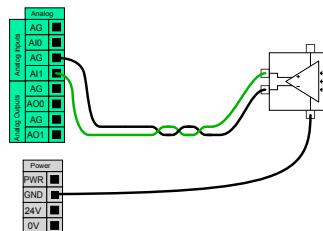
## Using an Analog Output

This example illustrates controlling a conveyor belt with an analog speed control input.



## Using an Analog Input

This example illustrates connecting an analog sensor.



### 6.4.7. Remote ON/OFF control

Use remote ON/OFF control to turn the Control Box on and off without using the Teach Pendant. It is typically used:

- When the Teach Pendant is inaccessible.
- When a PLC system must have full control.
- When several robots must be turned on or off at the same time.

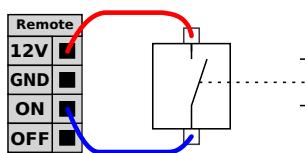
The remote ON/OFF control provides a auxiliary 12V supply, kept active when the Control Box is turned off. The ON input is intended only for short time activation and works in the same way as the POWER button. The OFF input can be held down as desired. Use a software feature to load and start programs automatically (see part [Part II PolyScope Manual on page 93](#)).

The electrical specifications are shown below.

Terminals	Parameter	Min	Typ	Max	Unit
[12V - GND]	Voltage	10	12	13	V
[12V - GND]	Current	-	-	100	mA
[ON / OFF]	Inactive voltage	0	-	0.5	V
[ON / OFF]	Active voltage	5	-	12	V
[ON / OFF]	Input current	-	1	-	mA
[ON]	Activation time	200	-	600	ms

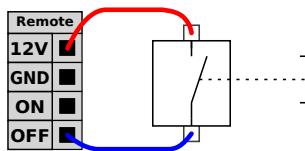
## Remote ON button

This example illustrates connecting a remote ON button.



## Remote OFF button

This example illustrates connecting a remote OFF button.



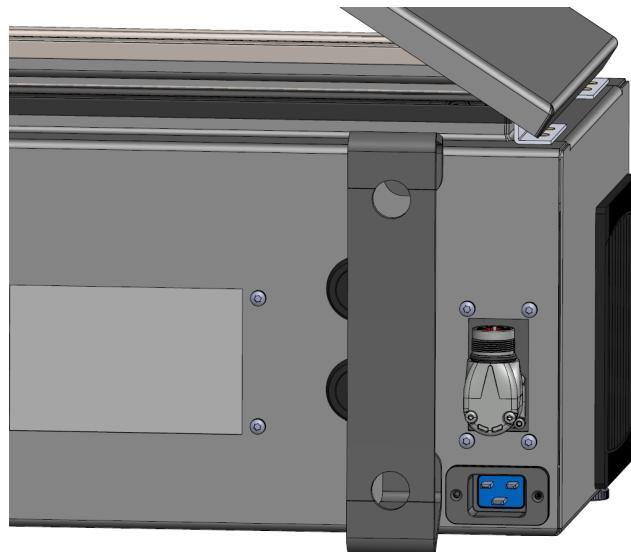
### CAUTION

Do not press and hold the **ON** input or the **POWER** button as it switches off the Control Box without saving. You must use the **OFF** input for remote off control as this signal allows the Control Box to save open files and shut down correctly.

## 6.5. Mains Connection

The mains cable from the Control Box has a standard IEC plug at the end. Connect a country specific mains plug, or cable, to the IEC plug.

In order to energize the robot, the Control Box must be connected to the mains via the standard IEC C20 plug at the bottom of the Control Box, through a corresponding IEC C19 cord (see illustration below).



The mains supply is equipped with the following:

- Connection to ground
- Main fuse
- Residual current device

It is recommended to install a main switch to power off all equipment in the robot application as an easy means for lockout-tagout under service. The electrical specifications are shown in the table below.

Parameter	Min	Typ	Max	Unit
Input voltage	100	-	240	VAC
External mains fuse (@ 100-200V)	8	-	16	A
External mains fuse (@ 200-265V)	8	-	16	A
Input frequency	47	-	440	Hz
Stand-by power	-	-	<1.5	W
Nominal operating power	90	150	325	W

**WARNING**

1. Ensure the robot is grounded correctly (electrical connection to ground). Use the unused bolts associated with grounding symbols inside the Control Box to create common grounding of all equipment in the system. The grounding conductor shall have at least the current rating of the highest current in the system.
2. Ensure the input power to the Control Box is protected with a Residual Current Device (RCD) and a correct fuse.
3. Lockout-tagout all power for the complete robot installation during service. Other equipment shall not supply voltage to the robot I/O when the system is locked out.
4. Ensure all cables are connected correctly before the Control Box is powered. Always use the original power cord.

## 6.6. Robot Connection: Robot Cable

This subsection describes the connection for a robot arm configured with a fixed 6 meter Robot Cable. For information on connecting a robot arm configured with a Base Flange Cable connector, see [6.7. Robot Connection: Base Flange Cable on the facing page](#).

### 6.6.1. Robot Cable connector

Establish the robot connection by connecting the robot arm to the Control Box with the Robot Cable.

Plug and lock the cable from the robot into the connector at the bottom of the Control Box (see illustration below). Twist the connector twice to ensure it is properly locked before turning on the robot arm.

You can turn the connector to the right to make it easier to lock after the cable is plugged in.



**CAUTION**

Improper robot connection can result in loss of power to the robot arm.

- Do not disconnect the Robot Cable when the robot arm is turned on.
- Do not extend or modify the original Robot Cable.

## 6.7. Robot Connection: Base Flange Cable

This subsection describes the connection for a robot arm configured with a Base Flange Cable connector. For information on connecting a robot arm configured with a fixed 6 meter Robot Cable, see [6.6. Robot Connection: Robot Cable on the previous page](#).

### 6.7.1. Base Flange Cable connector

Establish the robot connection by connecting the robot arm to the Control Box with the Robot Cable. The Robot Cable connects to the Base Flange Cable connector and to the Control Box connector. You can lock each connector when robot connection is established.

**CAUTION**

The maximum robot connection from the robot arm to the Control Box is 12 m.

Improper robot connection can result in loss of power to the robot arm.

- Do not extend a 6 m Robot Cable.

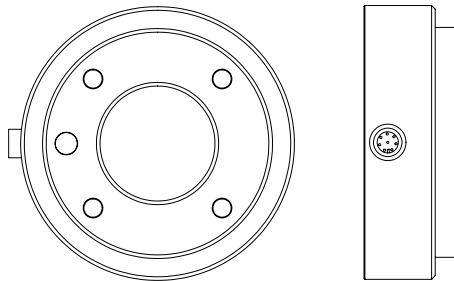
**NOTICE**

Connecting the Base Flange Cable directly to any Control Box can result in equipment or property damage.

- Do not connect the Base Flange Cable directly to the Control Box.

## 6.8. Tool I/O

Adjacent to the tool flange on Wrist #3, there is an eight-pinned connector that provides power and control signals for different grippers and sensors that can be attached to the robot. The Lumberg RKMV 8-354 is a suitable industrial cable. Each of the eight wires inside the cable have different colors representing different functions.



This connector provides power and control signals for grippers and sensors used on a specific robot tool. The industrial cable listed below is suitable:

- Lumberg RKMV 8-354.



### NOTE

The Tool Connector must be manually tightened up to a maximum of 0.4Nm.

The eight wires inside the cable have different colors that designate different functions. See table below:

Color	Signal	Description
Red	GND	Ground
Gray	POWER	0V/12V/24V
Blue	TO0/PWR	Digital Outputs 0 or 0V/12V/24V
Pink	TO1/GND	Digital Outputs 1 or Ground
Yellow	TI0	Digital Inputs 0
Green	TI1	Digital Inputs 1
White	AI2 / RS485+	Analog in 2 or RS485+
Brown	AI3 / RS485-	Analog in 3 or RS485-

Access Tool I/O in the Installation Tab (see part [Part II PolyScope Manual on page 93](#)) to set the internal power supply to 0V, 12V or 24V. The electrical specifications are shown below:

Parameter	Min	Typ	Max	Unit
Supply voltage in 24V mode	23.5	24	24.8	V
Supply voltage in 12V mode	11.5	12	12.5	V
Supply current (single pin)*	-	1000	2000**	mA
Supply current (dual pin)*	-	1500	2000**	mA
Supply capacitive load	-	-	8000***	uF

\*It is highly recommended to use a protective diode for inductive loads.

\*\*Peak for max 1 second, duty cycle max:10%. Average current over 10 seconds must not exceed typical current.

\*\*\*When tool power is enabled, a 400ms soft start time begins allowing a capacitive load of 8000uF to be connected to the tool power supply at startup. Hot-plugging the capacitive load is not allowed.

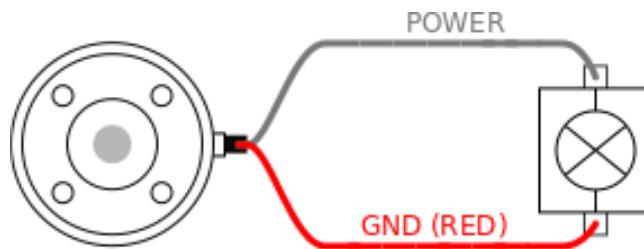
**NOTE**

The tool flange is connected to GND (same as the red wire).

### 6.8.1. Tool Power Supply

### 6.8.2. Power Supply

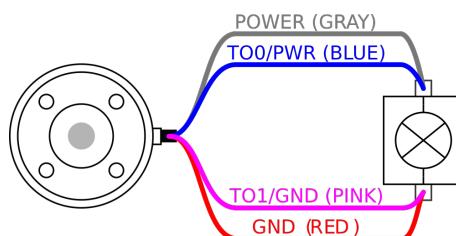
Access Tool I/O in the Installation Tab (see part [Part II PolyScope Manual on page 93](#)) to set the internal power supply to 0V, 12V or 24V.



### 6.8.3. Dual Pin Power Supply

In Dual Pin Power mode, the output current can be increased as listed in ([6.8. Tool I/O on page 47](#) table two).

1. In the Header, tap **Installation**.
2. In the list on the left, tap **General**.
3. Tap **Tool IO** and select **Dual Pin Power**.
4. Connect the wires Power (gray) to TO0 (blue) and Ground (red) to TO1 (pink).

**NOTE**

Once the robot makes an Emergency Stop, the voltage is set to 0V for both Power Pins (power is off).

### 6.8.4. Tool Digital Outputs

Digital Outputs support three different modes:

Mode	Active	Inactive
Sinking (NPN)	Low	Open
Sourcing (PNP)	High	Open
Push / Pull	High	Low

Access Tool I/O in the Installation Tab (see part [Part II PolyScope Manual on page 93](#)) to configure the output mode of each pin. The electrical specifications are shown below:

Parameter	Min	Typ	Max	Unit
Voltage when open	-0.5	-	26	V
Voltage when sinking 1A	-	0.08	0.09	V
Current when sourcing/sinking	0	1000	1000	mA
Current through GND	0	1000	3000*	mA

\*Peak for max 1 second, duty cycle max: 10%. Average current over 10 seconds must not exceed typical current.



#### NOTE

Once the robot makes an Emergency Stop, the Digital Outputs (DO0 and DO1) are deactivated (High Z).

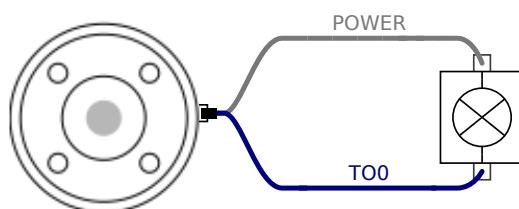


#### CAUTION

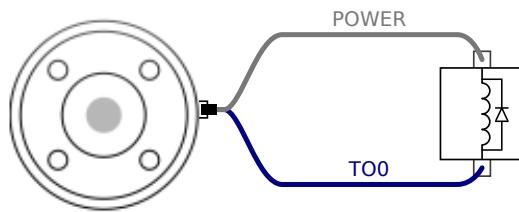
The Digital Outputs in the tool are not current-limited. Overriding the specified data can cause permanent damage.

## Using Tool Digital Outputs

This example illustrates turning on a load using the internal 12V or 24V power supply. The output voltage at the I/O tab must be define. There is voltage between the POWER connection and the shield/ground, even when the load is turned off.



It is recommended to use a protective diode for inductive loads, as shown below.



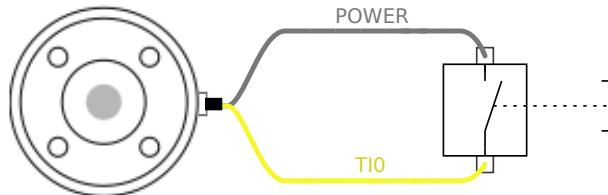
### 6.8.5. Tool Digital Inputs

The Digital Inputs are implemented as PNP with weak pull-down resistors. This means that a floating input always reads as low. The electrical specifications are shown below.

Parameter	Min	Type	Max	Unit
Input voltage	-0.5	-	26	V
Logical low voltage	-	-	2.0	V
Logical high voltage	5.5	-	-	V
Input resistance	-	47k	-	$\Omega$

### Using the Tool Digital Inputs

This example illustrates connecting a simple button.



### 6.8.6. Tool Analog Input

Tool Analog Input are nondifferential and can be set to either voltage (0-10V) or current (4-20mA) on the I/O tab (see part [Part II PolyScope Manual on page 93](#)). The electrical specifications are shown below.

Parameter	Min	Type	Max	Unit
Input voltage in voltage mode	-0.5	-	26	V
Input resistance @ range 0V to 10V	-	10.7	-	$k\Omega$
Resolution	-	12	-	bit
Input voltage in current mode	-0.5	-	5.0	V
Input current in current mode	-2.5	-	25	mA
Input resistance @ range 4mA to 20mA	-	182	188	$\Omega$
Resolution	-	12	-	bit

Two examples of using Analog Input are shown in the following subsections.

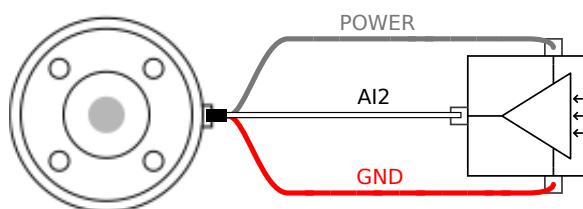
**CAUTION**

1. Analog Inputs are not protected against overvoltage in current mode. Exceeding the limit in the electrical specification can cause permanent damage to the input.

## Using Tool Analog Inputs, Nondifferential

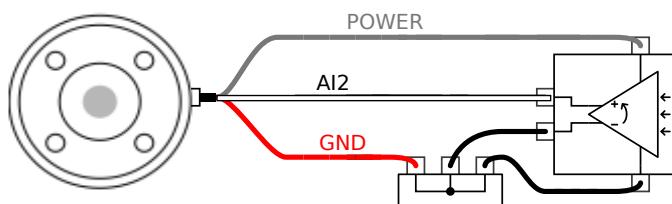
This example shows an analog sensor connection with a nondifferential output. The sensor output can be either current or voltage, as long as the input mode of that Analog Input is set to the same on the I/O tab.

Note: You can check that a sensor with voltage output can drive the internal resistance of the tool, or the measurement might be invalid.



## Using Tool Analog Inputs, Differential

This example shows an analog sensor connection with a differential output. Connecting the negative output part to GND (0V), works in the same way as a nondifferential sensor.



### 6.8.7. Tool Communication I/O

- **Signal requests** The RS485 signals use internal fail-safe biasing. If the attached device does not support this fail-safe, signal biasing must either be done in the attached tool, or added externally by adding pull-up resistors to RS485+ and pull-down to RS485-.
- **Latency** The latency of messages sent via the tool connector ranges from 2ms to 4ms, from the time the message is written on the PC to the start of the message on the RS485. A buffer stores data sent to the tool connector until the line goes idle. Once 1000 bytes of data have been received, the message is written on the device.

Baud Rates	9.6k, 19.2k, 38.4k, 57.6k, 115.2k, 1M, 2M, 5M
Stop Bits	1, 2
Parity	None, Odd, Even

# 7. Maintenance and Repair

Perform any visual or working condition inspection, in compliance with all safety instructions in this manual.

Conduct all maintenance, inspection, calibration and repair work according to the latest version of Service Manual on the support website: <http://www.universal-robots.com/support>.

Repair work must only be done by Universal Robots or authorized system integrators. Client designated, trained individuals can also do repair work, provided they follow the inspection plan outlined in the Service Manual. See the Service Manual: Chapter 5 for full inspection plan for trained individuals

All parts returned to Universal Robots shall be returned according to terms in the Service Manual.

## 7.1. Safety Instructions

After maintenance and repair work, checks must be done to ensure the required safety level. Checks must adhere to valid national or regional work safety regulations. The correct functioning of all safety functions shall also be tested.

The purpose of maintenance and repair work is to ensure that the system is kept operational or, in the event of a fault, to return the system to an operational state. Repair work includes troubleshooting in addition to the actual repair itself.

When working on the robot arm or control box, you must observe the procedures and warnings below.



### WARNING

1. Do not change anything in the safety configuration of the software (e.g. the force limit). The safety configuration is described in the PolyScope Manual. If any safety parameter is changed, the complete robot system shall be considered new, meaning that the overall safety approval process, including risk assessment, shall be updated accordingly.
2. Replace faulty components using new components with the same article numbers or equivalent components approved by Universal Robots for this purpose.
3. Reactivate any deactivated safety measures immediately after the work is completed.
4. Document all repairs and save this documentation in the technical file associated with the complete robot system.

**WARNING**

1. Remove the mains input cable from the bottom of the control box to ensure that it is completely unpowered. Deenergize any other source of energy connected to the robot arm or control box. Take necessary precautions to prevent other persons from energizing the system during the repair period.
2. Check the earth connection before re-powering the system.
3. Observe ESD regulations when parts of the robot arm or control box are disassembled.
4. Avoid disassembling the power supplies inside the control box. High voltages (up to 600 V) can be present inside these power supplies for several hours after the control box has been switched off.
5. Prevent water and dust from entering the robot arm or control box.

## 7.2. Cleaning

### Everyday cleaning

You can wipe away any dust/dirt/oil observed on the robot arm using a cloth and one of the following cleaning agents: Water, Isopropyl alcohol, 10% Ethanol alcohol or 10% Naphtha. In rare cases, very small amounts of grease can be visible from the joint. This does not affect the function, use, or lifetime of the joint.

### Additional cleaning

Due to the added focus on cleaning your robot, UR recommends cleaning with 70% isopropyl alcohol (rubbing alcohol).

1. Wipe the robot with a hard twisted micro fiber cloth and 70% isopropyl alcohol (rubbing alcohol).
2. Let the 70% isopropyl alcohol dwell on the robot for 5 minutes, and then clean the robot using standard cleaning procedure.

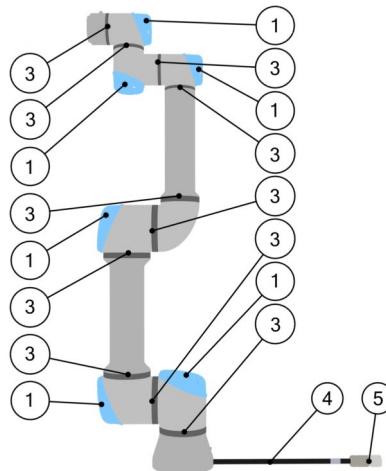
**DO NOT USE BLEACH.** Do not use bleach in any diluted cleaning solution.

## 7.3. Inspection

### 7.3.1. Robot Arm Inspection Plan

	Timeframe		
	Monthly	Biannually	Annually
V	X		
V		X	
V		X	
V		X	

V = Visual inspection \* = Must also be checked after heavy collision



### 7.3.2. Robot Arm Visual Inspection



#### NOTICE

Using compressed air to clean the robot arm can damage the robot arm components.

- Never use compressed air to clean the robot arm.



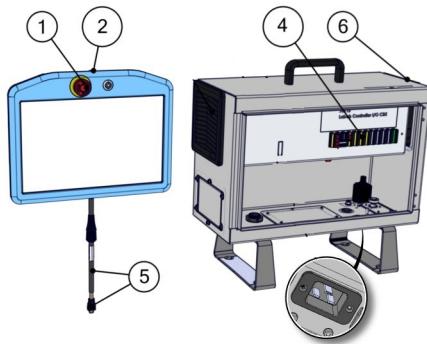
#### NOTICE

If any damage is observed on a robot within the warranty period, contact the distributor where the robot was purchased.

### 7.3.3. Control Box Inspection Plan

		Timeframe		
		Monthly	Biannually	Annually
1		F		✗
2 & 3		F		✗
		F		✗
4 & 5	Check Teach Pendant cable and connector	V		✗
6		V		✗

V = Visual inspection F = Functional inspection



### 7.3.4. Control Box Visual Inspection

1. Disconnect the power cable from Control Box.
2. Check for any dirt/dust outside the Control Box.
  - Clean with ESD vacuum cleaner if needed.



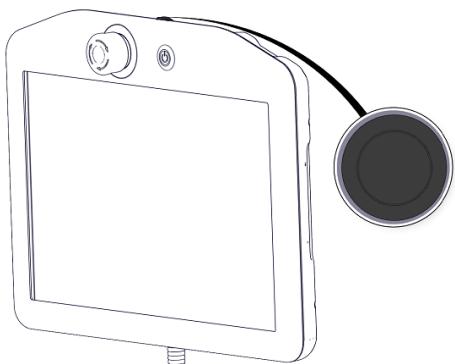
#### NOTICE

Using compressed air to clean inside the Control Box can damage components.

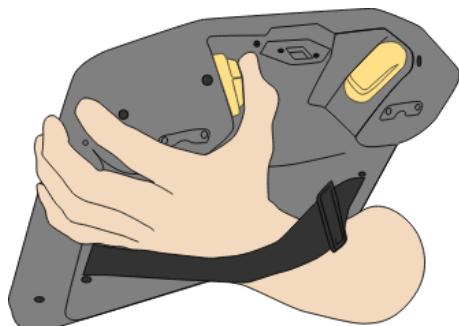
- Do not use compressed air to clean inside the Control Box.

### 7.3.5. Freedrive Inspection

1. Unmount attachment or set TCP/Payload/CoG according to tool specifications.
2. To move the robot arm in Freedrive:
  - On a standard Teach Pendant, press and hold the Freedrive button.
  - On a 3PE Teach Pendant, rapidly light-press, then light-press-and-hold the 3PE button.

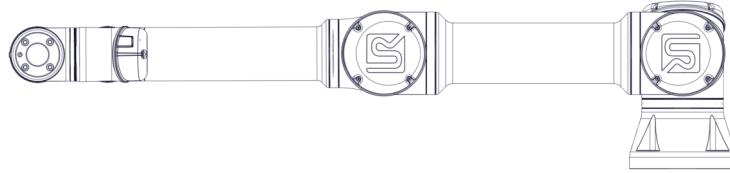


Standard TP



3PE TP

3. Pull/Push the robot arm to a horizontally elongated position and release.



4. Verify the robot can maintain its position when not being held with the Freedrive button still pressed.

## 8. Disposal and Environment

Universal Robots e-Series robots must be disposed of in accordance with the applicable national laws, regulations and standards.

Universal Robots e-Series robots are produced with restricted use of hazardous substances to protect the environment; as defined by the European RoHS directive 2011/65/EU. These substances include mercury, cadmium, lead, chromium VI, polybrominated biphenyls and polybrominated diphenyl ethers.

Fee for disposal and handling of electronic waste of Universal Robots e-Series robots sold on the Danish market is prepaid to DPA-system by Universal Robots A/S. Importers in countries covered by the European WEEE Directive 2012/19/EU must make their own registration to the national WEEE register of their country. The fee is typically less than 1€/robot. A list of national registers can be found here: <https://www.ewrn.org/national-registers>.

The following symbols are affixed on the robot to indicate conformity with the above legislations:



## 9. Certifications

Third party certification is voluntary. However, to provide the best service to robot integrators, Universal Robots chooses to certify its robots at the recognized test institutes listed below.

You can find copies of all certificates in the chapter [Certificates](#)

	TÜV NORD	Universal Robots e-Series robots are safety approved by TÜV NORD, a notified body under the machinery directive 2006/42/EC in EU.
	CHINA RoHS	Universal Robots e-Series robots conform to CHINA RoHS management methods for controlling pollution by electronic information products.
	KCC Safety	Universal Robots e-Series robots have been assessed and conform to KCC mark safety standards.
	KC Registration	The Universal Robots e-Series robots have been evaluated for conformity assessment for use in a work environment. Therefore, there is a risk of radio interference when used in a domestic environment.
	Delta	Universal Robots e-Series robots are performance tested by DELTA.

Supplier Third Party Certification		
	Environment	As provided by our suppliers, Universal Robots e-Series robots shipping pallets comply with the ISMPM-15 Danish requirements for producing wood packaging material and are marked in accordance with this scheme.

Manufacturer Test Certification		
	Universal Robots	Universal Robots e-Series robots undergo continuous internal testing and end of line test procedures. UR testing processes undergo continuous review and improvement.

### Declarations According to EU directives

Although EU directives are relevant for Europe, some countries outside Europe recognize and/or require EU declarations. European directives are available on the official homepage: <http://eur-lex.europa.eu>.

According to the Machinery Directive, Universal Robots' robots are partly completed machines, as such a CE mark is not to be affixed.

You can find the Declaration of Incorporation (DOI) according to the Machinery Directive in the chapter [Declarations and Certificates](#)

# 10. Warranties

## 10.1. Product Warranty

Without prejudice to any claim the user (customer) may have in relation to the dealer or retailer, the customer shall be granted a manufacturer's Warranty under the conditions set out below:

In the case of new devices and their components exhibiting defects resulting from manufacturing and/or material faults within 12 months of entry into service (maximum of 15 months from shipment), Universal Robots shall provide the necessary spare parts, while the user (customer) shall provide working hours to replace the spare parts, either replace the part with another part reflecting the current state of the art, or repair the said part. This Warranty shall be invalid if the device defect is attributable to improper treatment and/or failure to comply with information contained in the user guides. This Warranty shall not apply to or extend to services performed by the authorized dealer or the customer themselves (e.g. installation, configuration, software downloads). The purchase receipt, together with the date of purchase, shall be required as evidence for invoking the Warranty. Claims under the Warranty must be submitted within two months of the Warranty default becoming evident. Ownership of devices or components replaced by and returned to Universal Robots shall vest in Universal Robots. Any other claims resulting out of or in connection with the device shall be excluded from this Warranty. Nothing in this Warranty shall attempt to limit or exclude a Customer's Statutory Rights nor the manufacturer's liability for death or personal injury resulting from its negligence. The duration of the Warranty shall not be extended by services rendered under the terms of the Warranty. Insofar as no Warranty default exists, Universal Robots reserves the right to charge the customer for replacement or repair. The above provisions do not imply a change in the burden of proof to the detriment of the customer. In case of a device exhibiting defects, Universal Robots shall not be liable for any indirect, incidental, special or consequential damages, including but not limited to, lost profits, loss of use, loss of production or damage to other production equipment.

In case of a device exhibiting defects, Universal Robots shall not cover any consequential damage or loss, such as loss of production or damage to other production equipment.



### CAUTION

It is generally advised to avoid the use of accelerations higher than needed for a given application. High accelerations, especially in combination with high loads, can lead to reduced life time of the robot. For applications with short cycle times and high requirements for speed, it is generally advised to use blends as much as possible to ensure smooth trajectories without the need for high accelerations.

## 10.2. Disclaimer

Universal Robots continues to improve reliability and performance of its products, and therefore reserves the right to upgrade the product without prior warning. Universal Robots takes every care that the contents of this manual are precise and correct, but takes no responsibility for any errors or missing information.

# 11. Stopping Time and Stopping Distance



## NOTE

You can set user-defined safety rated maximum stopping time and distance. See [3.1. Introduction on page 17](#) and [21.6. Safety Menu Settings on page 117](#).

If user-defined settings are used, the program speed is dynamically adjusted to always comply with the selected limits.

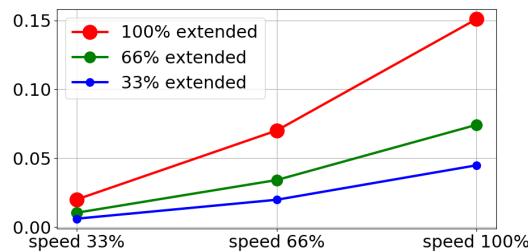
The graphical data provided for **Joint 0 (base)**, **Joint 1 (shoulder)** and **Joint 2 (elbow)** is valid for stopping distance and stopping time:

- Category 0
- Category 1
- Category 2

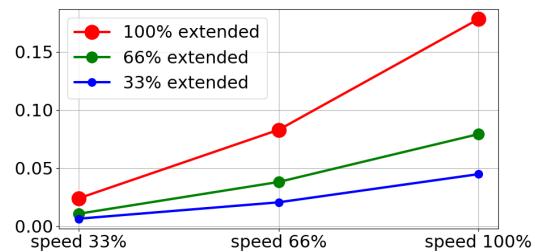
The **Joint 0** test was carried out by performing a horizontal movement, where the rotational axis was perpendicular to the ground.

During the **Joint 1** and **Joint 2** tests, the robot followed a vertical trajectory, where the rotational axes were parallel to the ground, and the stop was performed while the robot was moving downward.

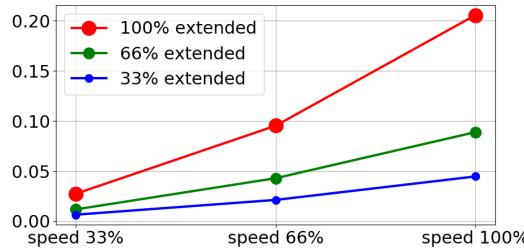
Stopping distance for joint 0 (BASE)



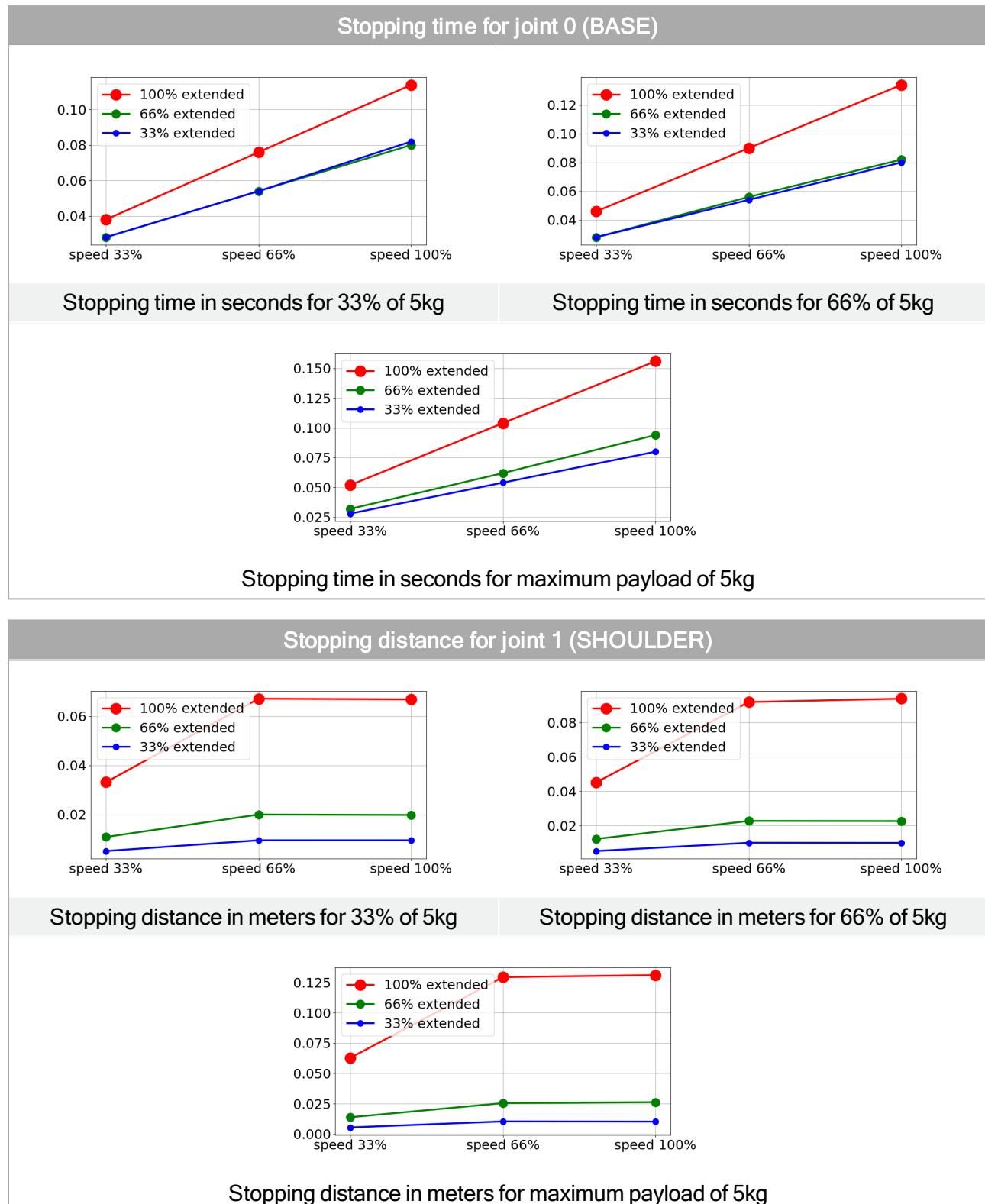
Stopping distance in meters for 33% of 5kg

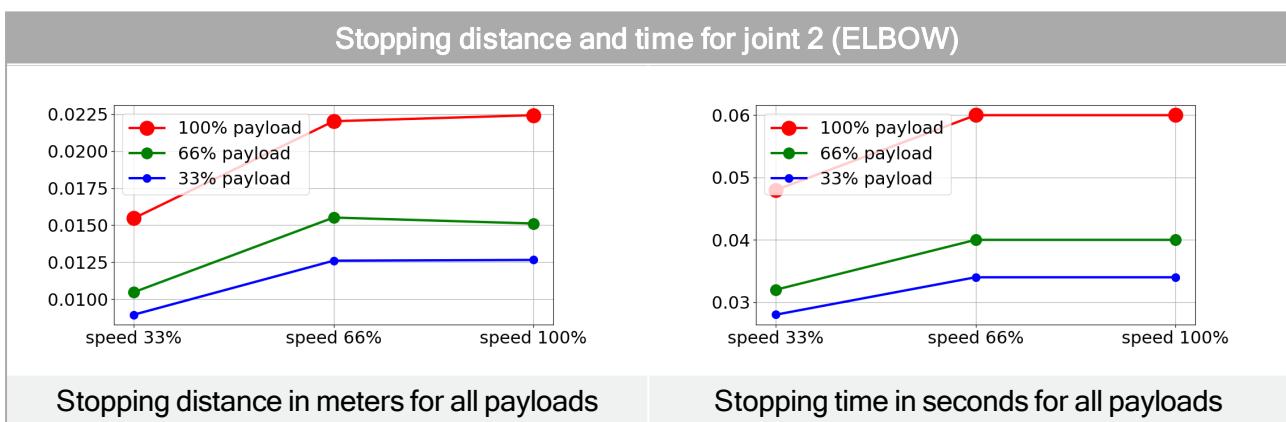
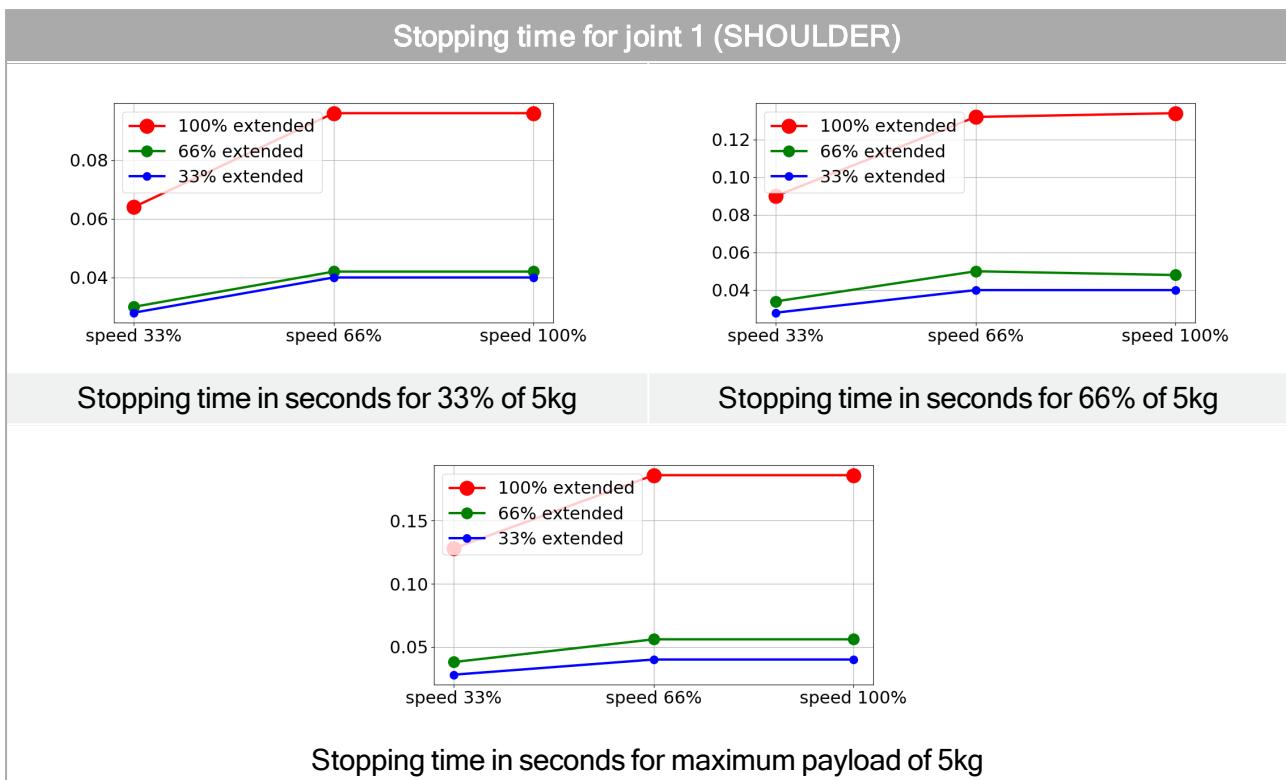


Stopping distance in meters for 66% of 5kg



Stopping distance in meters for maximum payload of 5kg





# 12. Declarations and Certificates

EU Declaration of Incorporation in accordance with ISO/IEC 17050-1:2010	
Manufacturer	Universal Robots A/S Energivej 25 DK-5260 Odense S Denmark
Person in the Community Authorized to Compile the Technical File	David Brandt Technology Officer, Research and Development Universal Robots A/S Energivej 25, DK-5260 Odense S
Description and Identification of the Partially-Completed Machine(s) Product and Function	Industrial robot (multi-axis manipulator with Control Box and Teach Pendant). Function is determined by the completed machine (with end-effector and intended use).
Model	UR3e, UR5e, UR10e, UR16e (e-Series)
Serial Number	Starting 20195000000 and higher – Effective 17 August 2019
Incorporation:	Universal Robots UR3e, UR5e, UR10e and UR16e shall only be put into service upon being integrated into a final complete machine (robot system, cell or application), which conforms with the provisions of the Machinery Directive and other applicable Directives.

**It is declared that the above products, for what is supplied, fulfill the following Directives as detailed below:**

I Machine Directive 2006/42/EC	The following essential requirements have been fulfilled: 1.1.2, 1.1.3, 1.1.5, 1.2.1, 1.2.4.3, 1.2.6, 1.3.4, 1.3.8.1, 1.5.1, 1.5.2, 1.5.6, 1.5.10, 1.6.3, 1.7.2, 1.7.4, 4.1.2.3  It is declared that the relevant technical documentation has been compiled in accordance with Part B of Annex VII of the Machinery Directive.
II Low-voltage Directive 2014/35/EU	Reference the LVD and the harmonized standards used below.
III EMC Directive 2014/30/EU	Reference the EMC Directive and the harmonized standards used below.
IV RoHS Directive 2011/65/EU	Reference the RoHS Directive 2011/65/EU.
V WEEE Directive 2012/19/EU.	Reference the WEEE Directive 2012/19/EU.

It is declared that the above products, for what is supplied, fulfill the following Directives as detailed below:

Reference the harmonized standards used, referred to in Article 7(2) of the MD and LV Directives and Article 6 of the EMC Directive

- (I) EN ISO 10218-1:2011 TUV Nord Cert. 4470814097607
- (I) EN ISO 13849-1:2015 TUV Nord Cert. 4420714097610
- (I) EN ISO 13732-1:2008, (I) EN 1037:1995+A1:2008
- (I) EN ISO 13849-2:2012, (I) EN ISO 13850:2015
- (II) EN 60204-1:2006/A1:2010, (II) EN 60320-1:2001/A1:2007
- (II) EN 60529:1991/A2:2013, (II) EN 60947-5-5:1997/A11:2013
- (III) EN 61000-6-2:2005, (III) EN 61000-6-4:2007/A1:2011

Reference to other technical standards and specifications used

- (I) ISO/TS 15066 as applicable
- (II) IEC 60664-1:2007, (II) IEC 60664-5:2007, (II) IEC 61326-3-1:2008
- (II) IEC 61784-3:2010 (SIL2) ISO 14664-1:2015 (Cleanroom Class 6 for control assembly with enclosure and Class 5 for UR3e, UR5e, UR10e and UR16e manipulators)
- (III) IEC 60068-2-1:2007, (III) IEC 60068-2-2:2007
- (III) IEC 60068-2-27:2008, (III) IEC 60068-2-64:2008

The manufacturer, or his authorized representative, shall transmit relevant information about the partly completed machinery in response to a reasoned request by the national authorities.

Approval of full quality assurance system (ISO 9001), by the notified body Bureau Veritas, certificate #DK008850.

Odense Denmark, 17 August 2019

Name:

Position/ Title

Universal Robots A/S, Energivej 25, DK-5260 Odense S, Denmark  
CVR-nr. 29 13 80 60



Roberta Nelson Shea  
Global Technical Compliance Officer

Phone +45 8993 8989  
Fax +45 3879 8989

info@universal-robots.com  
[www.universal-robots.com](http://www.universal-robots.com)

# 13. Certificates

TÜV NORD



Hiermit wird bescheinigt, dass die Firma / *This is to certify, that the company*

**Universal Robots A/S**  
**Energivej 25**  
**5260 Odense S**  
**Denmark**

berechtigt ist, das unten genannte Produkt mit dem abgebildeten Zeichen zu kennzeichnen.  
*is authorized to provide the product described below with the mark as illustrated.*

Fertigungsstätte:  
*Manufacturing plant:*

**Universal Robots A/S**  
**Energivej 25**  
**5260 Odense S**  
**Denmark**

Beschreibung des Produktes  
*(Details s. Anlage 1)*  
*Description of product*  
*(Details see Annex 1)*

**Universal Robots Safety System e-Series**  
**for UR16e, UR10e, UR5e and UR3e robots**

Geprüft nach:  
*Tested in accordance with:*

**EN ISO 13849-1:2015, Cat.3, PL d**



Registrier-Nr. / *Registered No.* 44 207 14097610  
 Prüfbericht Nr. / *Test Report No.* 3524 9741  
 Aktenzeichen / *File reference* 8003008239

Gültigkeit / *Validity*  
*von / from* 2019-07-16  
*bis / until* 2024-07-15

  
 Zertifizierungsstelle der TÜV NORD CERT GmbH  
*Certification body of TÜV NORD CERT GmbH*

Essen, 2019-07-16

TÜV NORD CERT GmbH    Langemarckstraße 20    45141 Essen    [www.tuev-nord-cert.de](http://www.tuev-nord-cert.de)    [technology@tuev-nord.de](mailto:technology@tuev-nord.de)  
 Bitte beachten Sie auch die umseitigen Hinweise  
*Please also pay attention to the information stated overleaf*



# ZERTIFIKAT CERTIFICATE

Hiermit wird bescheinigt, dass die Firma / *This certifies that the company*

**Universal Robots A/S**  
Energivej 25  
5260 Odense S  
Denmark

berechtigt ist, das unten genannte Produkt mit dem abgebildeten Zeichen zu kennzeichnen  
*is authorized to provide the product mentioned below with the mark as illustrated*

Fertigungsstätte  
*Manufacturing plant*

**Universal Robots A/S**  
Energivej 25  
5260 Odense S  
Denmark

Beschreibung des Produktes  
(Details s. Anlage 1)  
*Description of product*  
(*Details see Annex 1*)

**Industrial robot UR16e, UR10e, UR5e and UR3e**



Geprüft nach  
*Tested in accordance with*

**EN ISO 10218-1:2011**

Registrier-Nr. / *Registered No.* 44 780 14097607  
Prüfbericht Nr. / *Test Report No.* 3524 9416  
Aktenzeichen / *File reference* 8003008239

Gültigkeit / *Validity*  
von / *from* 2019-07-16  
bis / *until* 2024-07-15

  
Zertifizierungsstelle der  
TÜV NORD CERT GmbH

Essen, 2019-07-16

TÜV NORD CERT GmbH      Langemarckstraße 20      45141 Essen      [www.tuev-nord-cert.de](http://www.tuev-nord-cert.de)      [technology@tuev-nord.de](mailto:technology@tuev-nord.de)

Bitte beachten Sie auch die umseitigen Hinweise  
*Please also pay attention to the information stated overleaf*

## China RoHS

**Management Methods for Controlling Pollution  
by Electronic Information Products  
Product Declaration Table  
For Toxic or Hazardous Substances**

表1 有毒有害物质或元素名称及含量标识格式



Product/Part Name 产品/部件名称	Toxic and Hazardous Substances and Elements 有毒有害物质或元素					
	铅 Lead (Pb)	汞 Mercury (Hg)	镉 Cadmium (Cd)	六价 Hexavalent 铬 Chromium (Cr+6)	多溴联苯 Polybrominated biphenyls (PBB)	多溴二苯醚 Polybrominated diphenyl ethers (PBDE)
UR Robots UR3 / UR5 / UR10 UR机器人 UR3/UR5/UR10	X	O	X	O	X	X

O: Indicates that this toxic or hazardous substance contained in all of the homogeneous materials for this part is below the limit requirement in SJ/T11363-2006.

O: 表示该有毒有害物质在该部件所有均质材料中的含量均在SJ/T 11363-2006规定的限量要求以下。

X: Indicates that this toxic or hazardous substance contained in at least one of the homogeneous materials used for this part is above the limit requirement in SJ/T11363-2006.

X: 表示该有毒有害物质至少在该部件的某一均质材料中的含量超出SJ/T 11363-2006规定的限量要求。

(企业可在此处·根据实际情况对上表中打“X”的技术原因进行进一步说明。)

Items below are wear-out items and therefore can have useful lives less than environmental use period:  
下列项目是损耗品,因而它们的使用寿命可能短于环境使用时间:

Drives, Gaskets, Probes, Filters, Pins, Cables, Stiffener, Interfaces  
驱动器, 垫圈, 探针, 过滤器, 别针, 缆绳, 加强筋, 接口

Refer to product manual for detailed conditions of use.  
详细使用情况请阅读产品手册。

Universal Robots encourages that all Electronic Information Products be recycled but does not assume responsibility or liability.  
Universal Robots 鼓励回收再循环利用所有的电子信息产品,但 Universal Robots 不负任何责任或义务

To the maximum extent permitted by law, Customer shall be solely responsible for complying with, and shall otherwise assume all liabilities that may be imposed in connection with, any legal requirements adopted by any governmental authority related to the Management Methods for Controlling Pollution by Electronic Information Products (Ministry of Information Industry Order #39) of the Peoples Republic of China otherwise encouraging the recycle and use of electronic information products. Customer shall defend, indemnify and hold Universal Robots harmless from any damage, claim or liability relating thereto. At the time Customer desires to dispose of the Products, Customer shall refer to and comply with the specific waste management instructions and options set forth at <http://www.teradyne.com/about-teradyne/corporate-social-responsibility>, as the same may be amended by Teradyne or Universal Robots.

**KCC Safety**

**자율안전확인 신고증명서**

신청인	사업장명	Universal Robots A/S	사업장관리번호	2016E110079
	사업자등록번호	016E110079	대표자 성명	Klaus Vestergaard
	소재지	Energivej 25, 5260 Odense S, Denmark		
자율안전인증대상 기계·기구명	UR5e	산업용로봇	용량(등급)	6 axis
자율안전확인번호		18-AB2EQ-01603		
제조자		Universal Robots A/S		
소재지		Energivej 25, 5260 Odense S, Denmark		

「산업안전보건법」 제35조제1항 및 같은 법 시행규칙 제61조제3항에 따라  
자율안전확인 신고증명서를 발급합니다.

2018년 11월 06일



한국산업안전보건공단 서울지역본부장



## KC Registration

6C8D-81FA-93C2-784A

방송통신기자재등의 적합등록 풀증	
Registration of Broadcasting and Communication Equipments	
상호 또는 성명 Trade Name or Registrant	Universal Robots A/S
기자재명칭(제품명칭) Equipment Name	UR e-Series robot
기본모델명 Basic Model Number	UR5e
파생모델명 Series Model Number	
등록번호 Registration No.	R-R-URK-UR5e
제조자/제조(조립)국가 Manufacturer/Country of Origin	Universal Robots A/S / 펜마크
등록연월일 Date of Registration	2018-10-23
기타 Others	
위 기자재는 「전파법」 제58조의2 제3항에 따라 등록되었음을 증명합니다. It is verified that foregoing equipment has been registered under the Clause 3, Article 58-2 of Radio Waves Act.	
2018년(Year) 10월(Month) 23일(Day)  국립전파연구원장 Director General of National Radio Research Agency	
※ 적합등록 방송통신기자재는 반드시 "적합성평가표시"를 부착하여 유통하여야 합니다. 위반시 과태료 처분 및 등록이 취소될 수 있습니다.	

## Environmental Test Certificate

### Climatic and mechanical assessment



<b>Client</b> Universal Robots A/S Energivej 25 5260 Odense S Denmark	<b>Force Technology project no.</b> 117-32120
<b>Product identification</b> UR 3 robot arms UR 3 control boxes with attached Teach Pendants. UR 5 robot arms UR5 control boxes with attached Teach Pendants. UR10 robot arms: UR10 control boxes with attached Teach Pendants. See reports for details.	
<b>Force Technology report(s)</b> DELTA project no. 117-28266, DANAK-19/18069 DELTA project no. 117-28086, DANAK-19/17068	
<b>Other document(s)</b>	
<b>Conclusion</b> The three robot arms UR3, UR5 and UR10 including their control boxes and Teach Pendants have been tested according to the below listed standards. The test results are given in the Force Technology reports listed above. The tests were carried out as specified and the test criteria for environmental tests were fulfilled in general terms with only a few minor issues (see test reports for details).  IEC 60068-2-1, Test Ae; -5 °C, 16 h IEC 60068-2-2, Test Be; +35°C, 16h IEC 60068-2-2, Test Be; +50°C, 16 h IEC 60068-2-64, Test Fh; 5 – 10 Hz: +12 dB/octave, 10-50 Hz 0.00042 g <sup>2</sup> /Hz, 50 – 100 Hz: -12 dB/octave, 1,66 grms, 3 x 1½ h IEC 60068-2-27, Test Ea, Shock; 11 g, 11 ms, 3 x 18 shocks	
<b>Date</b>  Hørsholm, 25 August 2017	<b>Assessor</b>   Andreas Wendelboe Højsgaard M.Sc.Eng.

## EMC Test Certificate



# Attestation of Conformity

**AoC no. 1645**

Project / task no. 117-29565

DELTA has performed compliance test on electrical products since 1967. DELTA is an accredited test house according to EN17025 and participates in the international standardisation organisation CEN/CENELEC, IEC/CISPR and ETSI. This attestation of conformity with the below mentioned standards and/or normative documents is based on accredited tests and/or technical assessments carried out at DELTA – a part of FORCE Technology.

**Client**

Universal Robots A/S  
Energivej 25  
5260 Odense  
Denmark

**Product identification (type(s), serial no(s).)**

UR robot generation 5, G5 for models UR3, UR5, and UR10

**Manufacturer**

Universal Robots A/S

**Technical report(s)**

EMC test of UR robot generation 5, DELTA project no.117-29565-1 DANAK 19/18171

**Standards/Normative documents**

EMC Directive 2014/30/EU, Article 6  
EN 61326-3-1:2008 Industrial locations SIL 2  
EN/IEC 61000-6-1:2007  
EN/IEC 61000-6-2:2005  
EN/IEC 61000-6-3:2007+A1  
EN/IEC 61000-6-4:2007+A1  
EN/IEC 61000-3-2:2014  
EN/IEC 61000-3-3:2013

**DELTA – a part of  
FORCE Technology**  
Venlighedsvej 4  
2970 Hørsholm  
Denmark

Tel. +45 72 19 40 00  
Fax +45 72 19 40 01  
www.delta.dk  
VAT No. 55117314

The product identified above has been assessed and complies with the specified standards/normative documents. The attestation does not include any market surveillance. It is the responsibility of the manufacturer that mass-produced apparatus have the same properties and quality. This attestation does not contain any statements pertaining to the requirements pursuant to other standards, directives or laws other than the above mentioned.

Hørsholm, 15 August 2017

  
 Michael Nielsen  
 Specialist, Product Compliance

20aoc test-uk-1

# 14. Applied Standards

This section describes relevant standards applied under the development of the robot arm and control box. A European Directive number appearing in brackets indicates the standard is harmonized according to that Directive.

A standard is not a law, but a document developed by stakeholders within a given industry. Standards define the normal safety and performance requirements for a product or product group.

The abbreviations in this document mean the following:

ISO	International Standardization Organization
IEC	International Electrotechnical Commission
EN	European Norm
TS	Technical Specification
TR	Technical Report
ANSI	American National Standards Institute
RIA	Robotic Industries Association
CSA	Canadian Standards Association

Conformity with the following standards is only guaranteed if all assembly instructions, safety instructions and guidance in this manual are followed.

ISO 13849-1:2006 [PLd]  
ISO 13849-1:2015 [PLd]  
ISO 13849-2:2012  
EN ISO 13849-1:2008 (E) [PLd - 2006/42/EC]  
EN ISO 13849-2:2012 (E) (2006/42/EC)

*Safety of machinery - Safety-related parts of control systems*

*Part 1: General principles for design*

*Part 2: Validation*

The safety control system is designed as Performance Level d (PLd) according to the requirements of these standards.

ISO 13850:2006 [Stop Category 1]  
ISO 13850:2015 [Stop Category 1]  
EN ISO 13850:2008 (E) [Stop Category 1 - 2006/42/EC]  
EN ISO 13850:2015 [Stop Category 1 - 2006/42/EC]

*Safety of machinery - Emergency stop - Principles for design*

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.

**ISO 12100:2010****EN ISO 12100:2010 (E) [2006/42/EC]**

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

UR robots are evaluated according to the principles of this standard.

**ISO 10218-1:2011****EN ISO 10218-1:2011(E) [2006/42/EC]**

*Robots and robotic devices - Safety requirements for industrial robots*

**Part 1: Robots**

This standard is intended for the robot manufacturer, not the integrator. The second part (ISO 10218-2) is intended for the robot integrator, as it deals with the installation and design of the robot application.

**ANSI/RIA R15.06-2012**

*Industrial Robots and Robot Systems - Safety Requirements*

This American standard is the ISO standards ISO 10218-1 and ISO 10218-2 combined into one document. The language is changed from British English to American English, but the content is the same.

Part two (ISO 10218-2) of this standard is intended for the integrator of the robot system, and not Universal Robots.

**CAN/CSA-Z434-14**

*Industrial Robots and Robot Systems - General Safety Requirements*

This Canadian standard is the ISO standards ISO 10218-1 (see above) and -2 combined into one document. CSA added additional requirements for the user of the robot system. Some of these requirements might need to be addressed by the robot integrator.

Part two (ISO 10218-2) of this standard is intended for the integrator of the robot system, and not Universal Robots.

**IEC 61000-6-2:2005****IEC 61000-6-4/A1:2010****EN 61000-6-2:2005 [2004/108/EC]****EN 61000-6-4/A1:2011 [2004/108/EC]**

*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 61326-3-1:2008**

**EN 61326-3-1:2008**

*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 of UR robots provide safety even if other equipment exceeds the EMC emission limits defined in the IEC 61000 standards.

**IEC 61131-2:2007 (E)**

**EN 61131-2:2007 [2004/108/EC]**

*Programmable controllers*

*Part 2: Equipment requirements and tests*

Both normal and safety-rated 24V I/Os are constructed according to requirements of this standard to ensure reliable communication with other PLC systems.

**ISO 14118:2000 (E)**

**EN 1037/A1:2008 [2006/42/EC]**

*Safety of machinery - Prevention of unexpected start-up*

These two standards are very similar. They define safety principles for avoiding unexpected start-up, both as a result of unintended repowering during maintenance or repair, and as a result of unintended start-up commands from a control perspective.

**IEC 60947-5-5/A1:2005**

**EN 60947-5-5/A11:2013 [2006/42/EC]**

*Low-voltage switchgear and controlgear*

*Part 5-5: Control circuit devices and switching elements - Electrical emergency stop device with mechanical latching function*

The direct opening action and the safety lock mechanism of the emergency stop button comply with requirements in this standard.

## IEC 60529:2013 EN 60529/A2:2013

### *Degrees of protection provided by enclosures (IP Code)*

This standard defines enclosure ratings regarding protection against dust and water. UR robots are designed and classified with an IP code according to this standard, see robot sticker.

## IEC 60320-1/A1:2007 IEC 60320-1:2015 EN 60320-1/A1:2007 [2006/95/EC] EN 60320-1:2015

### *Appliance couplers for household and similar general purposes*

#### *Part 1: General requirements*

The mains input cable complies with this standard.

## ISO 9409-1:2004 [Type 50-4-M6]

### *Manipulating industrial robots - Mechanical interfaces*

#### *Part 1: Plates*

The tool flange on UR robots conforms to type 50-4-M6 of this standard. Robot tools should also be constructed according to this standard to ensure proper fitting.

## ISO 13732-1:2006 EN ISO 13732-1:2008 [2006/42/EC]

### *Ergonomics of the thermal environment - Methods for the assessment of human responses to contact with surfaces*

#### *Part 1: Hot surfaces*

The UR robots are designed so that the surface temperature is kept under the ergonomic limits defined in this standard.

## IEC 61140/A1:2004 EN 61140/A1:2006 [2006/95/EC]

### *Protection against electric shock - Common aspects for installation and equipment*

UR robots are constructed in compliance with this standard to provide protection against electrical shock. A protective earth/ground connection is mandatory, as defined in the [Part I Hardware Installation Manual on page 5](#).

IEC 60068-2-1:2007  
IEC 60068-2-2:2007  
IEC 60068-2-27:2008  
IEC 60068-2-64:2008  
EN 60068-2-1:2007  
EN 60068-2-2:2007  
EN 60068-2-27:2009  
EN 60068-2-64:2008

*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*

UR robots are tested according to the test methods defined in these standards.

IEC 61784-3:2010  
EN 61784-3:2010 [SIL 2]

*Industrial communication networks - Profiles*

*Part 3: Functional safety fieldbuses - General rules and profile definitions*

This standard defines requirements for safety-rated communication buses.

IEC 61784-3:2010  
EN 61784-3:2010 [SIL 2]

*Safety of machinery - Electrical equipment of machines*

*Part 1: General requirements*

The general principles of this standard are applied.

IEC 60664-1:2007  
IEC 60664-5:2007  
EN 60664-1:2007 [2006/95/EC]  
EN 60664-5:2007

*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*

The electrical circuitry of UR robots is designed in compliance with this standard.

## EUROMAP 67:2015, V1.11

*Electrical Interface between Injection Molding Machine and Handling Device / Robot*

UR robots equipped with the E67 accessory module to interface injection molding machines comply with this standard.

# 15. Technical Specifications

Robot type	UR5e
Weight	20.7 kg / 45.7 lb
Maximum payload	5 kg / 11 lb ( <a href="#">5. Mechanical Interface on page 27</a> )
Reach	850 mm / 33.5 in
Joint ranges	± 360 ° for all joints
Speed	Joints: Max 180 °/s . Tool: Approx. 1 m/s / Approx. 39.4 in/s.
System Update Frequency	500 Hz
Force Torque Sensor Accuracy	4 N
Pose Repeatability	± 0.03 mm / ± 0.0011 in (1.1 mils)per ISO 9283
Footprint	Ø149 mm / 5.9 in
Degrees of freedom	6 rotating joints
Control Box Size (W × H × D)	460 mm × 449 mm × 254 mm / 18.2 in × 17.6 in × 10 in
Control box I/O Ports	16 digital in, 16 digital out, 2 analog in, 2 analog out
Tool I/O Ports	2 digital in, 2 digital out, 2 analogue in
Tool Communication	RS
Tool I/O Power Supply	12 V/24 V 1.5 A (Dual pin) 1 A (Single pin)
I/O Power Supply	24 V 2 A in Control Box
Communication	TCP/IP 1000 Mbit: IEEE 802.3ab, 1000BASE-T Ethernet socket, MODBUS TCP & EtherNet/IP Adapter, Profinet
Programming	PolyScope graphical user interface on 12" touchscreen
Noise	Robot Arm: Less than 65dB(A) Control Box: Less than 50dB(A)
IP classification	IP54
Cleanroom classification	Robot Arm: ISO Class 6, Control Box: ISO Class 6
Maximum Average Power	570 W
Power consumption	Approx. 250 W using a typical program
Short-Circuit Current Rating (SCCR)	200A
Collaboration operation	17 advanced safety functions. In compliance with: EN ISO 13849-1, PLd, Cat.3 and EN ISO 10218-1
Materials	Aluminium, PP plastic
Temperature	The robot can work in an ambient temperature range of 0-50 °C

Power supply	100-240 VAC, 47-440 Hz
TP Cable: Teach Pendant to Control Box	4.5 m / 177 in
Robot Cable: Robot Arm to Control Box (options)	Standard (PVC) 6 m/236 in x 13.4 mm Standard (PVC) 12 m/472.4 in x 13.4 mm HiFlex (PUR) 6 m/236 in x 12.1 mm HiFlex (PUR) 12 m/472.4 in x 12.1 mm

# 16. Safety Functions Tables

## 16.1. Table 1



### NOTE

The Safety Functions tables presented in this chapter are simplified. You can find the comprehensive versions of them here: <https://www.universal-robots.com/support>

Universal Robots e-Series Safety Functions and Safety I/O are PLd, Category 3 (ISO 13849-1), with certification by TÜV NORD (certificate # 44 207 14097610).

Safety Function (SF) Descriptions (see Chapter 2 of manual: For safety I/O, the resulting safety function including the external device or equipment is determined by the overall architecture and the sum of all PFHds, including the UR robot safety function PFHd. All safety functions are individual safety functions.



### WARNING

If any safety function limit is exceeded, or a fault is detected in a safety function or safety-related part of the control system, the result is a Category 0 stop (immediate removal of power) according to IEC 60204-1.

SF# and Safety Function	Description	What happens?	Tolerance	PFHd	Affects
SF1 1,2,3,4 Emergency Stop (according to ISO 13850)	Pressing the Estop PB on the pendant1 or the External Estop (if using the Estop Safety Input) results in a Cat 1 stop with power removed from the robot actuators and the tool I/O. Command1 all joints to stop and upon all joints coming to a monitored standstill state, power is removed. See Stop Time and Stop Distance Safety Functions. ONLY USE FOR EMERGENCY PURPOSES, not safeguarding.	Category 1 stop (IEC 60204-1)	-	1.30E-07	Robot including robot tool I/O

SF# and Safety Function	Description	What happens?	Toler- ance	PFHd	Affects
SF2 Safeguard Stop4 (Protective Stop according to ISO 10218-1)	<p>This safety function is initiated by an external protective device using safety inputs which will initiate a Cat 2 stop3. The tool I/O are unaffected by the safeguard stop. Various configurations are provided. If an enabling device is connected, it is possible to configure the safeguard stop to function in automatic mode ONLY. See the Stop Time and Stop Distance Safety Functions4 . For the functional safety of the complete integrated safety function, add the PFHd of the external protective device to the PFHd of the Safeguard Stop.</p>	Category 2 stop (IEC 60204-1) SS2 stop (as described in IEC 61800-5-2)	-	1.20E-07	Robot
SF3 Joint Position Limit (soft axis limiting)	<p>Sets upper and lower limits for the allowed joint positions. Stopping time and distance is not a considered as the limit(s) will not be violated. Each joint can have its own limits. Directly limits the set of allowed joint positions that the joints can move within. It is set in the safety part of the User Interface. It is a means of safety-rated soft axis limiting and space limiting, according to ISO 10218-1:2011, 5.12.3.</p>	Will not allow motion to exceed any limit settings. Speed could be reduced so motion will not exceed any limit. A protective stop will be initiated to prevent exceeding any limit.	5°	1.20E-07	Joint (each)
SF4 Joint Speed Limit	<p>Sets an upper limit for the joint speed. Each joint can have its own limit. This safety function has the most influence on energy transfer upon contact (clamping or transient). Directly limits the set of allowed joint speeds which the joints are allowed to perform. It is set in the safety setup part of the User Interface. Used to limit fast joint movements, e.g. risks related to singularities.</p>	Will not allow motion to exceed any limit settings. Speed could be reduced so motion will not exceed any limit. A protective stop will be initiated to prevent exceeding any limit.	1.15 °/s	1.20E-07	Joint (each)

SF# and Safety Function	Description	What happens?	Tolerance	PFHd	Affects
Joint Torque Limit	Exceeding the internal joint torque limit (each joint) results in a Cat 0 stop3. This is shown as SF #5 in the Generation 3 (CB3) UR robots. This is not accessible to the user; it is a factory setting. It is NOT shown as a safety function because there are no user settings and no user configuration possibilities.	-	-	-	-
SF5 Called various names: Pose Limit, Tool Limit, Orientation Limit, Safety Planes, Safety Boundaries	Monitors the TCP Pose (position and orientation) and will prevent exceeding a safety plane or TCP Pose Limit. Multiple pose limits are possible (tool flange, elbow, and up to 2 configurable tool offset points with a radius) Orientation restricted by the deviation from the feature Z direction of the tool flange OR the TCP. This safety function consists of two parts. One is the safety planes for limiting the possible TCP positions. The second is the TCP orientation limit, which is entered as an allowed direction and a tolerance. This provides TCP and wrist inclusion/exclusion zones due to the safety planes.	Will not allow motion to exceed any limit settings. Speed or torques could be reduced so motion will not exceed any limit. A protective stop will be initiated to prevent exceeding any limit. Will not allow motion to exceed any limit settings.	3 ° 40 mm	1.20E-07	TCP Tool flange Elbow
SF6 Speed Limit TCP & Elbow	Monitors the TCP and elbow speed to prevent exceeding a speed limit.		50 mm/s	1.20E-07	TCP

SF# and Safety Function	Description	What happens?	Tolerance	PFHd	Affects
SF7 Force Limit (TCP)	<p>The Force Limit is the force exerted by the robot at the TCP (tool center point) and “elbow”. The safety function continuously calculates the torques allowed for each joint to stay within the defined force limit for both the TCP &amp; the elbow. The joints control their torque output to stay within the allowed torque range. This means that the forces at the TCP or elbow will stay within the defined force limit. When a monitored stop is initiated by the Force Limit SF, the robot will stop, then “back-off” to a position where the force limit was not exceeded. Then it will stop again.</p>	<p>Will not allow motion to exceed any limit settings. Speed or torques could be reduced so motion will not exceed any limit. A protective stop will be initiated to prevent exceeding any limit. Will not allow motion to exceed any limit settings.</p>	25N	1.50E-07	TCP
SF8 Momentum Limit	<p>The momentum limit is very useful for limiting transient impacts. The Momentum Limit affects the entire robot.</p>		3 kg m/s	1.20E-07	Robot
SF9 Power Limit	<p>This function monitors the mechanical work (sum of joint torques times joint angular speeds) performed by the robot, which also affects the current to the robot arm as well as the robot speed. This safety function dynamically limits the current/ torque but maintains the speed.</p>	<p>Dynamic limiting of the current/torque</p>	10 W	1.50E-07	Robot

SF# and Safety Function	Description	PFHd	Affects
SF10 UR Robot Estop Output	<p>When configured for Estop output and there is an Estop condition (see SF1), the dual outputs are LOW. If there is no Estop condition, dual outputs are high. Pulses are not used but they are tolerated. For the integrated functional safety rating with an external Estop device, add the PFHd of the UR Estop function (SF0 or SF1) to the PFHd of the external logic (if any) and its components (e.g. Estop pushbutton).</p> <p>For the Estop Output, validation is performed at the external equipment, as the UR output is an input to this external equipment.</p>	4.70E-08	External connection to logic and/or equipment
SF11 UR Robot Moving: Digital Output	<p>Whenever the robot is moving (motion underway), the dual digital outputs are LOW. Outputs are HIGH when no movement.</p> <p>The functional safety rating is for what is within the UR robot.</p> <p>The integrated functional safety performance requires adding this PFHd to the PFHd of the external logic (if any) and its components.</p>	1.20E-07	External connection to logic and/or equipment
SF12 UR Robot Not stopping: Digital Output	<p>Whenever the robot is STOPPING (in process of stopping or in a stand-still condition) the dual digital outputs are HIGH. When outputs are LOW, robot is NOT in the process of stopping and NOT in a stand-still condition. The functional safety rating is for what is within the UR robot.</p> <p>The integrated functional safety performance requires adding this PFHd to the PFHd of the external logic (if any) and its components.</p>	1.20E-07	External connection to logic and/or equipment
SF13 UR Robot Reduced Mode: Digital Output	<p>Whenever the robot is in reduced mode (or reduced mode is initiated), the dual digital outputs are LOW. See below. The functional safety rating is for what is within the UR robot.</p> <p>The integrated functional safety performance requires adding this PFHd to the PFHd of the external logic (if any) and its components.</p>	1.20E-07	External connection to logic &/or equipment
SF14 UR Robot Not Reduced Mode: Digital Output	<p>Whenever the robot is NOT in reduced mode (or the reduced mode is not initiated), the dual digital outputs are LOW. The functional safety rating is for what is within the UR robot.</p> <p>The integrated functional safety performance requires adding this PFHd to the PFHd of the external logic (if any) and its components.</p>	1.20E-07	External connection to logic &/or equipment

SF# and Safety Function	Description	What happens?	Tolerance	PFHd	Affects
SF15 Stopping Time Limit	Real time monitoring of conditions such that the stopping time limit will not be exceeded. Robot speed is limited to ensure that the stop time limit is not exceeded. The control SW continuously calculates the stopping capability of the robot in the given motion. If the time needed to stop the robot is at risk of exceeding the time limit, the speed of motion is reduced to ensure the limit is not exceeded. The safety function performs the same calculation of the stopping time and initiates a cat 0 stop if they are exceeded.	Will not allow the actual stopping time to exceed the limit setting. Causes decrease in speed or a protective stop so as NOT to exceed the limit	50 ms	1.20E-07	Robot
SF16 Stopping Distance Limit	Real time monitoring of conditions such that the stopping distance limit will not be exceeded. Robot speed is limited to ensure that the stop distance limit will not be exceeded. The control SW continuously calculates the stopping capability of the robot in the given motion. If the distance needed to stop the robot is at risk of exceeding the distance limit, the speed of motion is reduced to ensure the limit is not exceeded. The safety function performs the same calculation of the stopping distance and initiates a cat 0 stop if they are exceeded.	Will not allow the actual stopping time to exceed the limit setting. Causes decrease in speed or a protective stop so as NOT to exceed the limit	40 mm	1.20E-07	Robot
SF17 Safe Home Position	Safety function which monitors a safety rated output, such that it ensures that the output can only be activated when the robot is in the configured “safe home position”. A cat 0 stop is initiated if the output is activated when the robot is not in the configured position.	The “safe home output” can only be activated when the robot is in the configured “safe home position”	1.7 °	1.20E-07	External connection to logic and/or equipment

Safety Function	Description	PFHd	Affects
Reduced Mode SF settings change	<p>Reduced Mode can be initiated by a safety plane/ boundary (starts when at 2cm of the plane and reduced mode settings are achieved within 2cm of the plane) or by use of an input to initiate (will achieve reduced settings within 500ms).</p> <p>When the external connections are Low, Reduced Mode is initiated. Reduced Mode means that ALL reduced mode limits are ACTIVE.</p> <p>Reduced mode is not a safety function, rather it is a state affecting the settings of the following safety function limits: joint position, joint speed, TCP pose limit, TCP speed, TCP force, momentum, power, stopping time, and stopping distance.</p>	PFHd is either 1.20E-07 or 1.50E-07	Robot depending on the safety function
Safeguard Reset	<p>When configured for Safeguard Reset and the external connections transition from low to high, the safeguard stop RESETS. Safety input to initiate a reset of safeguard stop safety function.</p>	Input to SF2 (See SF2)	Robot
3 Position Enabling Device INPUT	<p>When the external Enabling Device connections are Low, a Safeguard Stop (SF2) is initiated. Recommendation: Use with a mode switch as a safety input. If a mode switch is not used and connected to the safety inputs, then the robot mode will be determined by the User Interface. If the User Interface is in</p> <ul style="list-style-type: none"> <li>“run mode”, the enabling device will not be active.</li> <li>“programming mode”, the enabling device will be active. It is possible to use password protection for changing the mode by the User Interface.</li> </ul>	Input to SF2 (See SF2)	Robot safeguard stop
Mode switch INPUT	<p>When the external connections are Low, Operation Mode (running/ automatic operation in automatic mode) is in effect. When High, mode is programming/ teach. Recommendation: Use with a 3-position enabling device. When in teach/program, initially the TCP speed will be limited to 250mm/s. The speed can manually be increased by using the pendant user interface “speed-slider”, but upon activation of the enabling device, the speed limitation will reset to 250mm/s.</p>	Input to SF2 (See SF2)	Robot safeguard stop

## 16.2. Table 2

UR e-Series robots comply with ISO 10218-1:2011 and the applicable portions of ISO/TS 15066. It is important to note that most of ISO/TS 15066 is directed towards the integrator and not the robot manufacturer. ISO 10218-1:2011, clause 5.10 collaborative operation details 4 collaborative operation techniques as explained below. It is very important to understand that collaborative operation is of the APPLICATION when in AUTOMATIC mode.

#	ISO 10218-1	Technique	Explanation	UR e-Series
1	Collaborative Operation 2011 edition, clause 5.10.2	Safety-rated monitored stop	<p>Stop condition where position is held at a standstill and is monitored as a safety function. Category 2 stop is permitted to auto reset. In the case of resetting and restarting operation after a safety -rated monitored stop, see ISO 10218-2 and ISO/TS 15066 as resumption shall not cause hazardous conditions.</p> <p>This is essentially individual and direct personal control while the robot is in automatic mode.</p> <p>Hand guiding equipment shall be located close to the end-effector and shall have:</p> <ul style="list-style-type: none"> <li>• an Emergency Stop pushbutton</li> <li>• a 3-position enabling device</li> <li>• a safety-rated monitored stop function</li> <li>• a settable safety-rated monitored speed function</li> </ul>	<p>UR robots' safeguard stop is a safety-rated monitored stop, See SF2 on page 1. It is likely, in the future, that "safety-rated monitored stop" will not be called a form of collaborative operation.</p>
2	Collaborative Operation 2011 edition, clause 5.10.3	Hand-guiding		<p>UR robots do not provide hand-guiding for collaborative operation. Hand-guided teach (free drive) is provided with UR robots but this is for programming in manual mode and not for collaborative operation in automatic mode.</p>

#	ISO 10218-1	Technique	Explanation	UR e-Series
3	Collaborative Operation 2011 edition, clause 5.10.4	Speed and separation monitoring (SSM) safety functions	<p>SSM is the robot maintaining a separation distance from any operator (human). This is done by monitoring of the distance between the robot system and intrusions to ensure that the MINIMUM PROTECTIVE DISTANCE is assured. Usually, this is accomplished using Sensitive Protective Equipment (SPE), where typically a safety laser scanner detects intrusion(s) towards the robot system.</p> <p>This SPE causes:</p> <ol style="list-style-type: none"> <li>1. dynamic changing of the parameters for the limiting safety functions; or</li> <li>2. a safety-rated monitored stop condition.</li> </ol> <p>Upon detection of the intrusion exiting the protective device's detection zone, the robot is permitted to:</p> <ol style="list-style-type: none"> <li>1. resume the "higher" normal safety function limits in the case of 1) above</li> <li>2. resume operation in the case of 2) above</li> </ol> <p>In the case of 2) 2), restarting operation after a safety -rated monitored stop, see ISO 10218-2 and ISO/TS 15066 for requirements.</p>	<p>To facilitate SSM, UR robots have the capability of switching between two sets of parameters for safety functions with configurable limits (normal and reduced). See Reduced Mode on page 4. Normal operation can be when no intrusion is detected. It can also be caused by safety planes/safety boundaries. Multiple safety zones can be readily used with UR robots. For example, one safety zone can be used for "reduced settings" and another zone boundary is used as a safeguard stop input to the UR robot. Reduced limits can also include a reduced setting for the stop time and stop distance limits - to reduce the work area and floorspace.</p>

#	ISO 10218-1	Technique	Explanation	UR e-Series
4	Collaborative Operation 2011 edition, clause 5.10.5	Power and force limiting (PFL) by inherent design or control	<p>How to accomplish PFL is left to the robot manufacturer. The robot design and/or safety functions will limit the energy transfer from the robot to a person. If any parameter limit is exceeded, a protective stop happens. PFL applications require considering the ROBOT APPLICATION (including the end-effector and workpiece(s)), so that any contact will not cause injury. The study performed evaluated pressures to the ONSET of pain, not injury. See Annex A. See ISO/TR 20218-1 End-effectors.</p>	<p>UR robots are power and force limiting robots specifically designed to enable collaborative applications where the robot could contact a person and cause no injury. UR robots have safety functions that can be used to limit motion, speed, momentum, force, power and more of the robot. These safety functions are used in the robot application to thereby lessen pressures and forces caused by the end-effector and workpiece(s).</p>



# Part II

# PolyScope Manual



# 17. Introduction

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



## NOTE

Before powering on the robot for the first time, your designated Universal Robots robot integrator must:

1. Read the safety information in the [Hardware Installation Manual](#).
2. Set the safety configuration parameters defined by the risk assessment (see [21. Safety Configuration on page 115](#)).

## 17.1. Robot Arm Basics

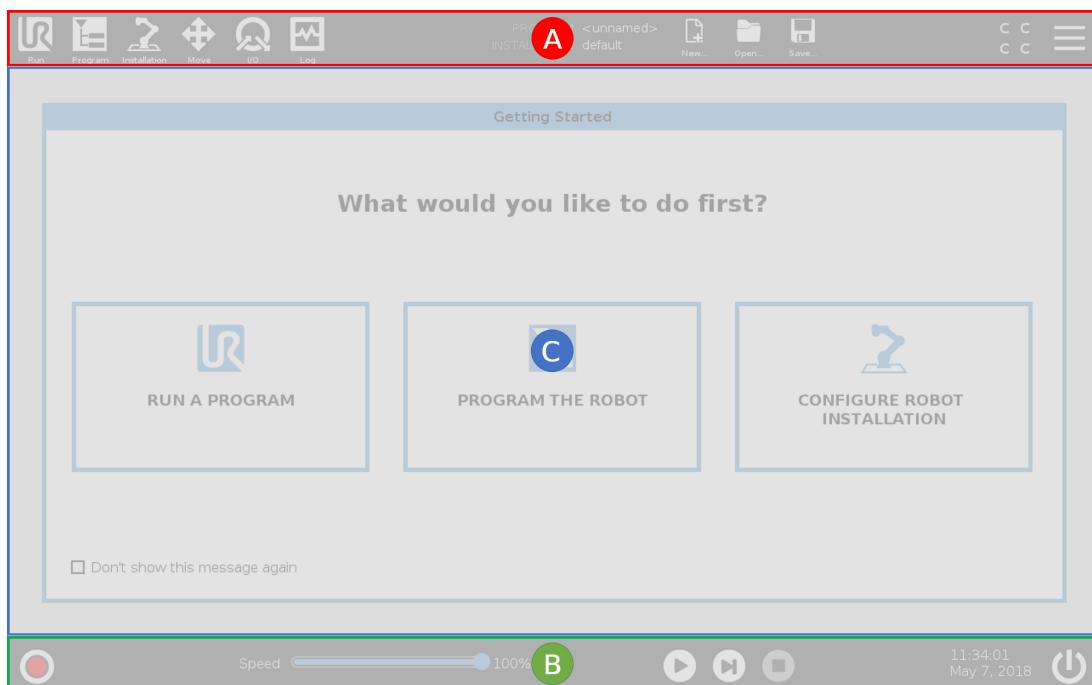
The Universal Robots robot arm is composed of tubes and joints. You use the PolyScope to coordinate the motion of these joints to move the robot arm. You attach tools to end of the robot arm, or Tool Flange. Moving the robot arm positions the tool. You cannot position the tool directly above, or directly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

## 17.2. PolyScope Basics

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.



## 17.2.1. Touch Screen

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.

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

### Using the Touch Screen

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.

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

## 17.2.2. Header Icons/Tabs



**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 (see [28.4. File manager on page 246](#)). 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.



**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.

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



**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.

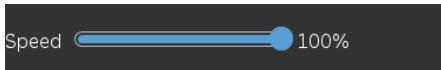


Hamburger Menu accesses PolyScope Help, About and Settings.

### 17.2.3. Footer Buttons



**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.



**250mm/s** **Manual High Speed** allows the tool speed to temporarily exceed 250mm/s. This hold-to-run function is only available in Manual mode when a Three-Position Enabling Device is configured.



**Play** starts current loaded robot Program.



**Step** allows a Program to be run single-stepped.



**Stop** halts current loaded robot Program.

## 17.3. Installation

### 17.3.1. Installing the Robot Arm and Control Box

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

See [Hardware Installation Manual](#) for detailed installation instructions.

1. Unpack the **robot arm** and the **Control Box**.
2. Mount the robot arm on a sturdy, vibration-free surface.

Verify the surface can withstand at least 10 times the full torque of the base joint and at least 5 times the weight of the robot arm.

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 plug of the control box.



#### WARNING

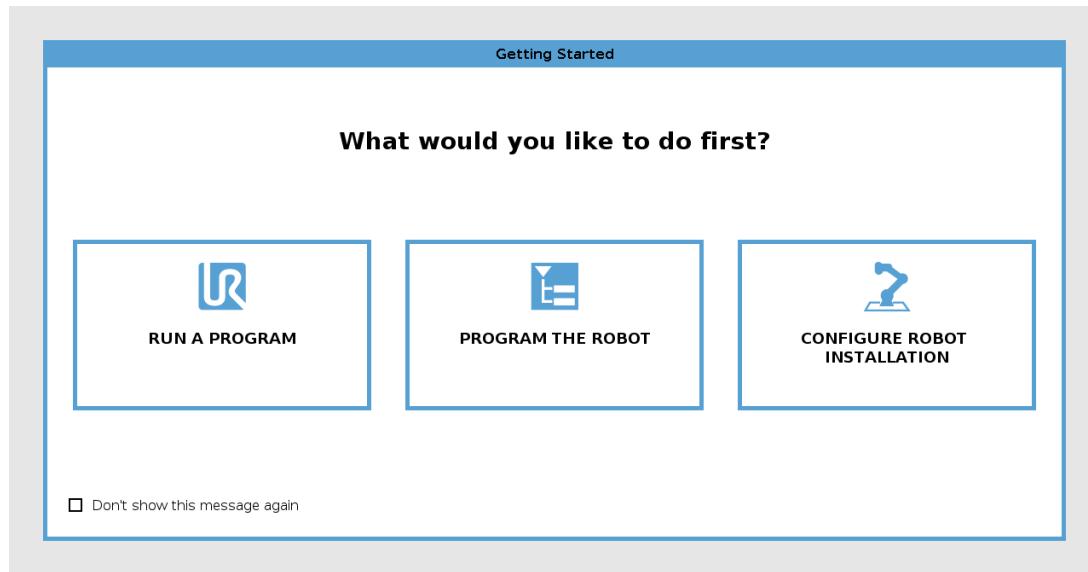
Tipping hazard. If the robot is not securely placed on a sturdy surface, the robot can fall over and cause injury.

### 17.3.2. Turning the Control Box On/Off

The Control Box mainly contains the physical electrical Input/Output that connects the robot arm, the Teach Pendant and any peripherals. You must turn on the Control Box to be able to power on the robot arm.

1. On your Teach Pendant, press the power button to turn on the control box.
2. Wait as text from the underlying operating system, followed by buttons, appear on the screen.

A Getting Started screen can appear, prompting you to begin programming the robot.

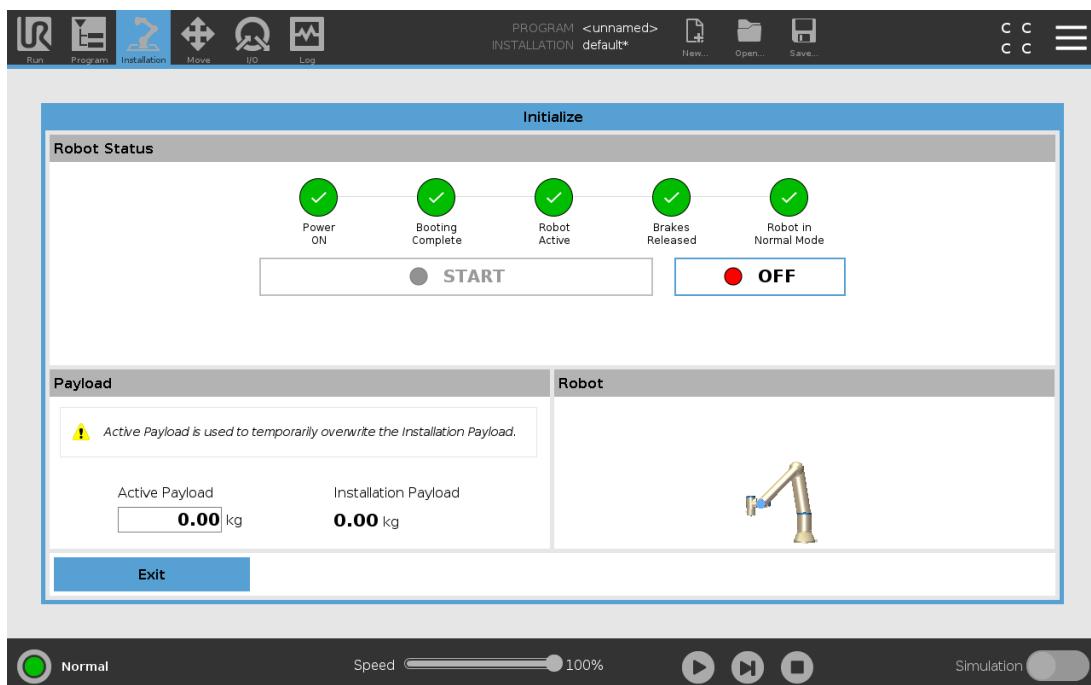


### 17.4. Initialization

On your first start up a Cannot Proceed dialog box can appear. Select **Go to initialization screen** to access the Initialize screen.

In the Footer, to the left, the Initialize button indicates the status of the robot arm using colors:

- **Red** Power off. The robot arm is in a stopped state.
- **Yellow** Idle. The robot arm is on, but not ready for normal operation.
- **Green** Normal. The robot arm is on and ready for normal operation.



### 17.4.1. Starting up the Robot Arm



#### WARNING

Always verify the actual payload and installation are correct before starting up the robot arm. If these settings are incorrect, the robot arm and Control Box will not function correctly and may become dangerous to people or equipment.



#### CAUTION

Ensure the robot arm is not touching an object (e.g., a table) because a collision between the robot arm and an obstacle might damage a joint gearbox.

To start the robot:

1. Tap the ON button with the green LED to start the initialization process. Then, the LED turns yellow to indicate the power is on and in **Idle**.
2. Tap the START button to release the breaks.
3. Tap the OFF button with the red LED to power off the robot arm.
  - When the PolyScope starts, tap the ON button once to power the robot arm. Then, the status changes to yellow to indicate the robot is on and idle.
  - When the robot arm state is **Idle**, tap the START button to start robot arm. At this point, sensor data is checked against the configured mounting of the robot arm.

If a mismatch is found (with a tolerance of 30°), the button is disabled and an error message is displayed below it.

- If the mounting is verified, tap Start to release all joint brakes and the robot arm is ready for normal operation.

Robot arm start up is accompanied by sound and slight movements as joint brakes are released.

## 17.5. Quick System Start-up

Before using the PolyScope, verify that the Robot Arm and Control Box are correctly installed.

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 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 robot state to change to **Idle**.
8. In the **Payload** field, in **Active Payload**, verify the payload mass. You can also verify 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.



### NOTE

Learn to program your Universal Robots robot on [www.universal-robots.com/academy/](http://www.universal-robots.com/academy/)

## 17.6. The First Program

A program is a list of commands telling the robot what to do. For most tasks, programming is done entirely using the PolyScope. 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 (see [25. Move Tab on page 229](#)) 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 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.

The following is a simple program that allows a Robot Arm to move between two waypoints.

To create a simple program

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.



#### NOTE

1. Do not drive the robot into itself or anything else as this may cause damage to the robot.
2. This is only a quick start guide 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.

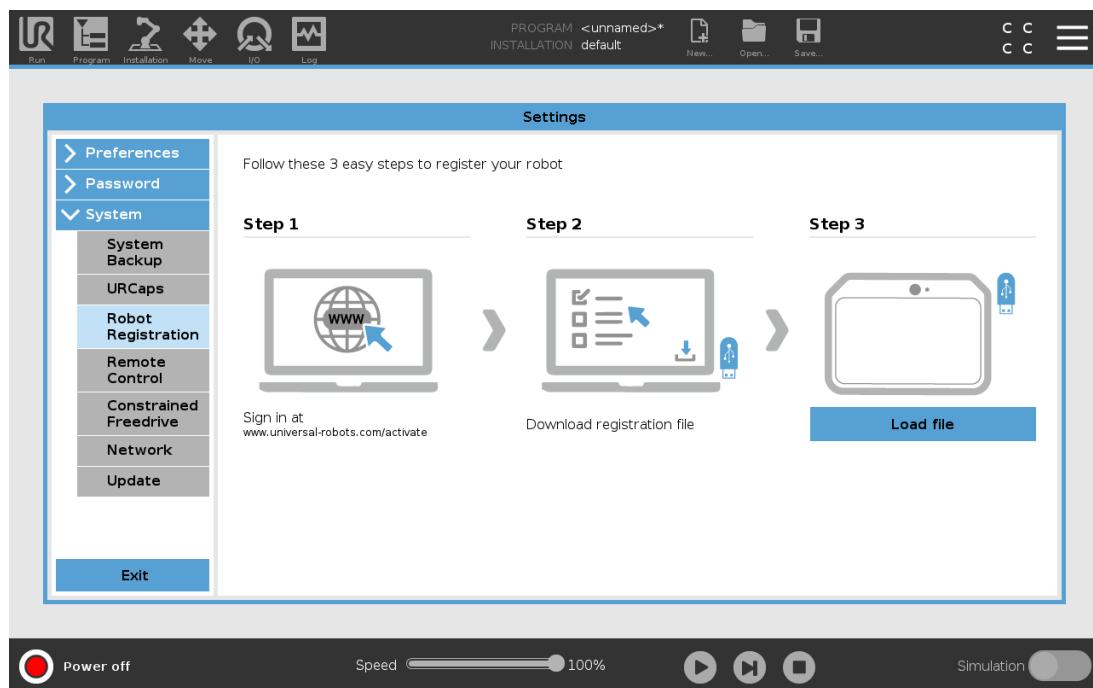


#### WARNING

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

## 17.7. Robot Registration and URCap License files

Before using the Remote TCP URCap, register the robot and download and install the URCap License File (see [23.13. URCaps on page 186](#)).

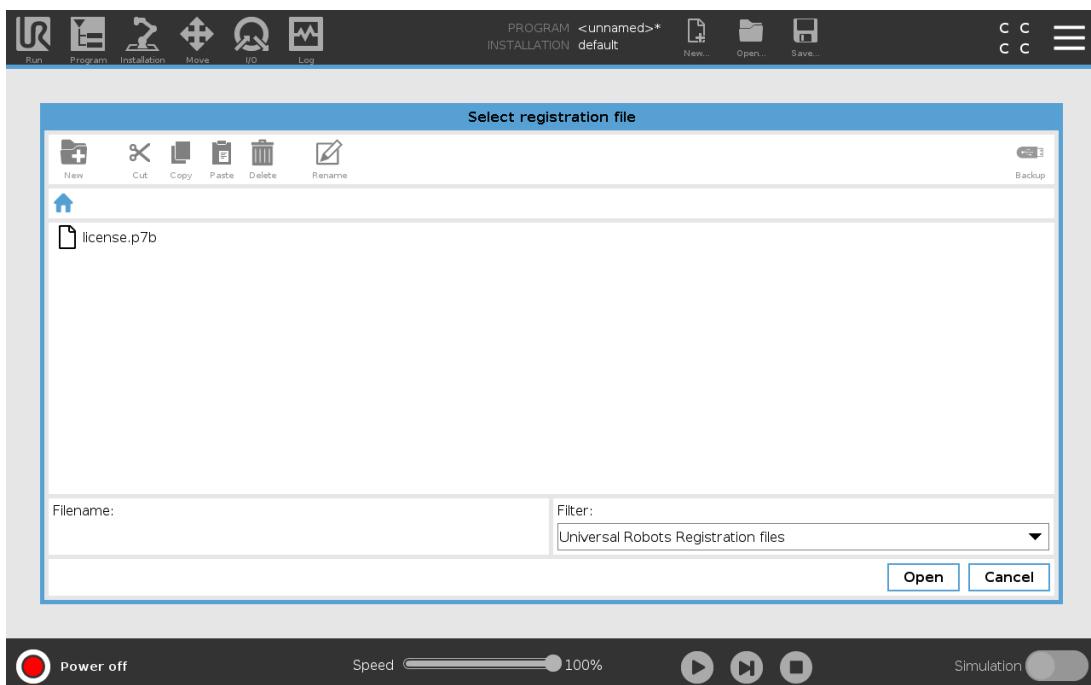


### 17.7.1. Registering the Robot from Your Current Screen

1. In the Header, tap the **Hamburger Menu** and select **Settings**.
2. In the menu on the left, tap **System** and select **Robot Registration** for the Settings screen to appear.
3. Follow steps 1 and 2 on the screen to register your robot.

### 17.7.2. Downloading the URCap License File

1. Fill in the required fields online and 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 registration 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**.



### 17.7.3. Unregistering Robot

A new license is required if the robot changes owners. In this case, the robot must first be unregistered.

1. In the Header, tap the **Hamburger Menu** and select **Settings**.
2. In the menu on the left, tap **System** and select **Robot Registration**.
3. On the bottom right of the Settings screen, tap **Unregister**.

## 17.8. Robot Cyber Security

Before implementing cyber security, you must conduct a risk assessment to:

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

### 17.8.1. Cyber Security Pre-requisites

Before your system can reach a secure state of operation, ensure the following:

- You have a thorough understanding of general cyber security principles and advanced technologies as used in your Universal Robots robot.
- You take physical security measures to allow only trusted personnel physical access to the robot.
- You only connect your robot to a trusted network, behind a firewall restricting both inbound and outbound access to/from the Internet.

## 17.8.2. Hardening Cyber Security

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

- Always set an Admin password (see [Admin on page 248](#)) before connecting your robot to any network.
- 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 connections made. In some applications this is a security liability.
- Use local port forwarding (see [Enabling SSH access on page 253](#)) to setup an authenticated and secure connection, if you require remote access to the motion controlling interfaces of the robot. For example: the Dashboard Server and the Primary/Secondary/Realtime Client Interfaces.
- Remove sensitive data from the robot before it is decommissioned. Pay particular attention to the URCaps (see [23.13. URCaps on page 186](#)) and data in the program folder.



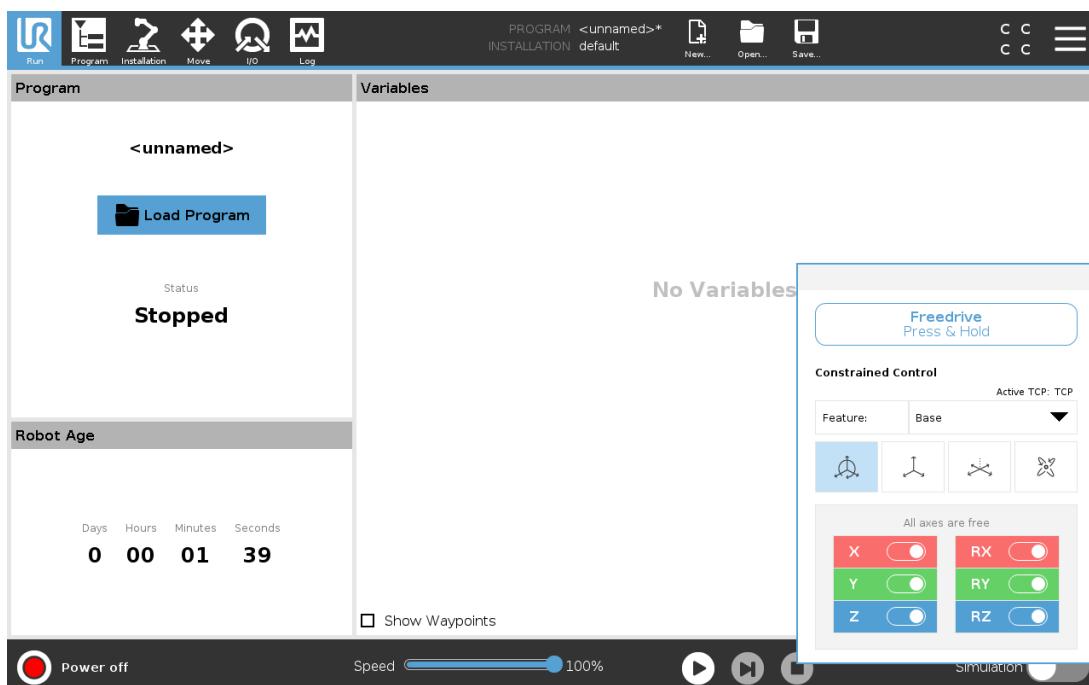
# 18. Freedrive

This chapter describes how to start using the robot arm in Freedrive.

Freedrive allows the robot arm to be manually pulled into desired positions and poses.

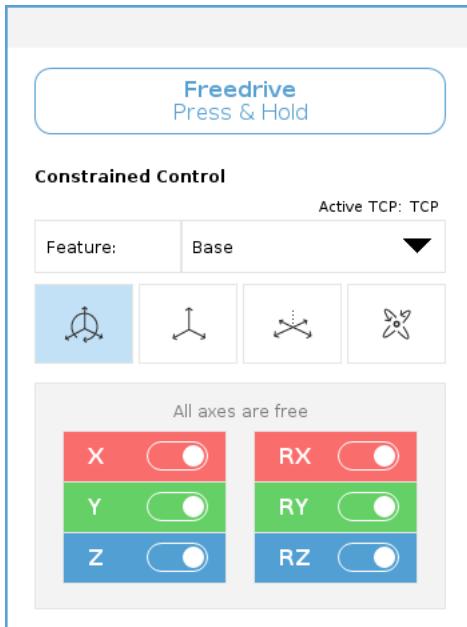
As you use Freedrive, the robot arm creates a repelling force if it approaches a joint limit or a safety plane (see [21.11. Planes on page 121](#)).

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



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.



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

	Movement is allowed through all axes.
	Movement is only allowed through the X-axis and Y-axis.
	Movement is allowed through all axes, without rotation.
	Movement is allowed through all axes, in a spherical motion, around the TCP.



### WARNING

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

## 18.1. Enabling Freedrive: Standard Teach Pendant

You can enable Freedrive in the following ways:

- Use the Freedrive button.
- Use the Freedrive button on the Move Tab screen (see [25.2.4. Freedrive on page 230](#)).
- Use I/O Actions (see ).

**WARNING**

- Do not enable Freedrive when you are pushing or touching the robot, as this may cause the robot to drift.
- Do not switch axes while moving the robot in Freedrive mode, as this may cause the robot to drift.

### 18.1.1. Using the Freedrive Button

1. Press and hold down the Freedrive button located on the Teach Pendant.
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.

### 18.1.2. Using the Freedrive Button on the Move Tab Screen

1. In the Header, tap **Move**.
2. In the Robot panel, tap **Freedrive**.
3. When the Freedrive panel appears, select the desired movement type for the robot arm's joints. Or use the list of axes to customize the movement type.
4. 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.
5. On the Freedrive panel, press and hold **Freedrive** to move the robot arm.

## 18.2. Enabling Freedrive: 3PE Teach Pendant

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

1. Rapidly light-press, then light-press-and-hold, the 3PE button.

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

# 19. Backdrive

Backdrive is 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.

## 19.1. Enabling Backdrive: Standard Teach Pendant

1. On the **Initialize** screen, tap **ON** to start the power up sequence.
2. When the robot state is **Idle**, press and hold the **Freedrive** button. The robot state changes to **Backdrive**.
3. Now you can apply significant pressure to release the brake in a desired joint.

As long as the Freedrive button is engaged/pressed, Backdrive is enabled, making the robot joints feel heavy to move.

## 19.2. Enabling Backdrive: 3PE Teach Pendant

To use the 3PE 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 **Idle**, light-press, then light-press-and-hold, the 3PE button. The robot state changes to **Backdrive**.
  3. Now you can apply significant pressure to release the brake in a desired joint.
- As long as light-press is maintained on the 3PE button, Backdrive is enabled. The robot joints feel heavy to move.

# 20. Operational Mode Selection

## 20.1. Operational Modes

Operational Modes are enabled when you configure a Three-Position Enabling Device, set a password, define an Operational Mode Configurable I/O, or via the Dashboard Server (see [Using the Dashboard Server on page 113](#)).

**Automatic Mode** Once activated, the robot can only perform pre-defined tasks. The Move Tab and Freedrive Mode are unavailable if a Three-Position Enabling Device is configured. You cannot modify or save programs and installations.



### WARNING

Automatic Mode Safeguard Stop can only be activated in Automatic Mode, so the safeguard's protective function is only active in Automatic Mode.

**Manual Mode** Once activated, you can program the robot using the Move Tab, Freedrive Mode and Speed Slider. You can modify and save programs and installations.

Operational mode	Manual	Automatic
Freedrive	x	*
Move robot with arrows on Move Tab	x	*
Speed Slider	x	x**
Edit & save program & installation	x	
Execute Programs	Reduced speed***	x
Start program from selected node	x	

\*Only when no Three-Position Enabling Device is configured.

\*\* The Speed Slider on the Run Screen can be enabled in the PolyScope Settings.

\*\*\* If a Three-Position Enabling Device is configured, the robot operates at Manual Reduced Speed unless Manual High Speed is activated.



### NOTE

- A Universal Robots robot may not be equipped with a Three-Position Enabling Device. If the risk assessment requires the device, it must be attached before the robot is used.
- If a Three-Position Enabling Device is not configured, speed is not reduced in Manual Mode.



### WARNING

- Any suspended safeguards must be returned to full functionality before selecting Automatic Mode.
- Wherever possible, the Manual Mode of operation shall be performed with all persons outside the safeguard space.
- The device used to switch between Operational Modes must be placed outside the safeguarded space.
- The user must not enter the safeguarded space when robot is in Automatic Mode, unless an Automatic Mode Safeguard input is configured.

The methods for configuring Operational Mode selection are described in the following subsections. Each method is exclusive, meaning that using one method, makes the other two methods inactive.

## Using Operational Mode Safety Input

1. Tap the Installation Tab, and select Safety I/O.
2. Configure the Operational Mode Input. The option to configure appears in the drop-down menu.
  - The robot is in Automatic Mode when the Operational Mode Input is low.
  - The robot is in Manual Mode when the Operational Mode Input is high.



### NOTE

The physical mode selector, if used, must completely adhere to ISO 10218-1: article 5.7.1 for selection.

## Switching Modes

1. To switch between modes, 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.

PolyScope is automatically in Manual Mode when the Safety I/O configuration with Three-Position Enabling Device is enabled.

## 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

See <http://universal-robots.com/support/> for more on using the Dashboard server.

## 20.2. Three-Position Enabling Device

When a Three-Position Enabling Device is configured and the **Operational Mode** is in Manual Mode, the robot can only be moved by pressing the Three-Position Enabling Device. Connecting and configuring a Three-Position Enabling Device makes Automatic Mode Safeguard Stop accessible. The Three-Position Enabling Device has no effect in Automatic Mode.

### 20.2.1. Manual High Speed

The hold-to-run function, **Manual High Speed**, allows the tool speed to temporarily exceed 250mm/s. It is only available when your robot is in Manual mode and a Three-Position Enabling Device is configured. The robot performs a Safeguard Stop in Manual mode, if a Three-Position Enabling Device is configured, but not pressed. 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 Manual High Speed, use safety joint limits (see [21.10. Joint Limits on page 120](#)) or safety planes (see [21.11. Planes on page 121](#)) to restrict the robot's moving space.



# 21. Safety Configuration

## 21.1. Safety Settings Basics

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

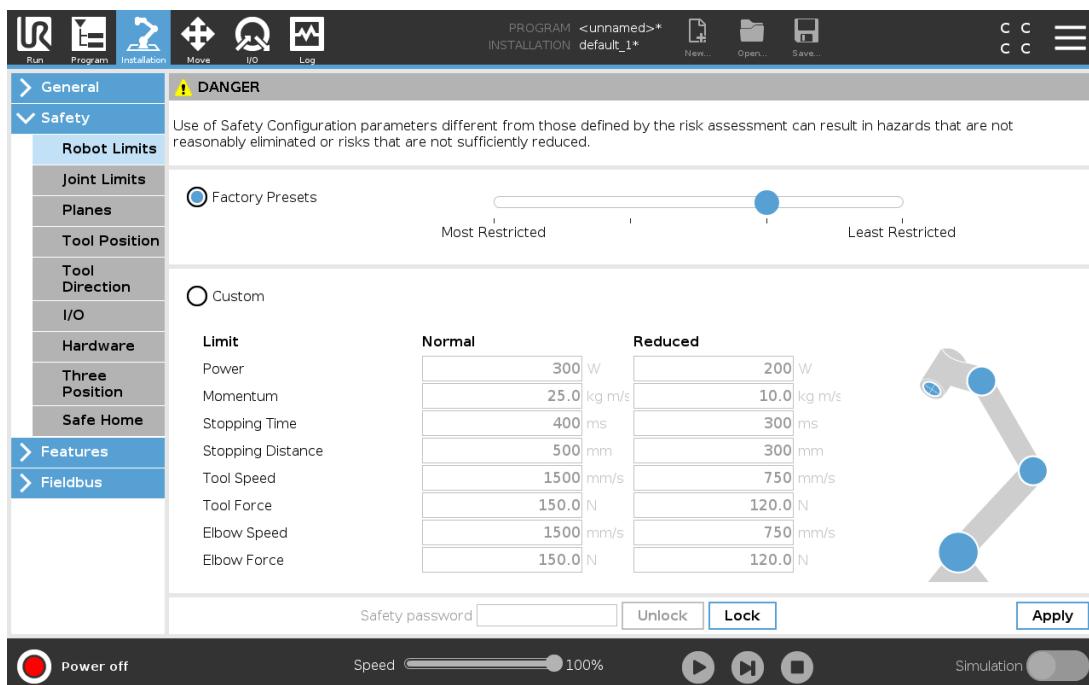
Before you configure your robot safety settings, your integrator must conduct a risk assessment to guarantee the safety of personnel and equipment around the robot. A risk assessment is an evaluation of all work procedures throughout the robot lifetime, conducted in order to apply correct safety configuration settings (see [Hardware Installation Manual](#) ). You must set the following in accordance with the integrator's 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 (see [Hardware Installation Manual](#) ).
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.

### 21.1.1. Accessing Safety Configuration

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

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.



You can find more safety system information in the [Hardware Installation Manual](#).

## 21.2. Setting a Safety 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.

Safety password  **Unlock** **Lock**

## 21.3. Changing the Safety Configuration

Changes to the Safety Configuration settings must comply with the risk assessment conducted by the integrator (see [Hardware Installation Manual](#) ).

Recommended procedure:

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.”

## 21.4. Applying New Safety Configuration

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

## 21.5. Safety Checksum



The **Safety Checksum** icon displays your applied robot safety configuration and is read from top to bottom, left to right e.g. BF4B. 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.

## 21.6. Safety Menu Settings

This section defines Safety menu settings that make up your robot Safety configuration.

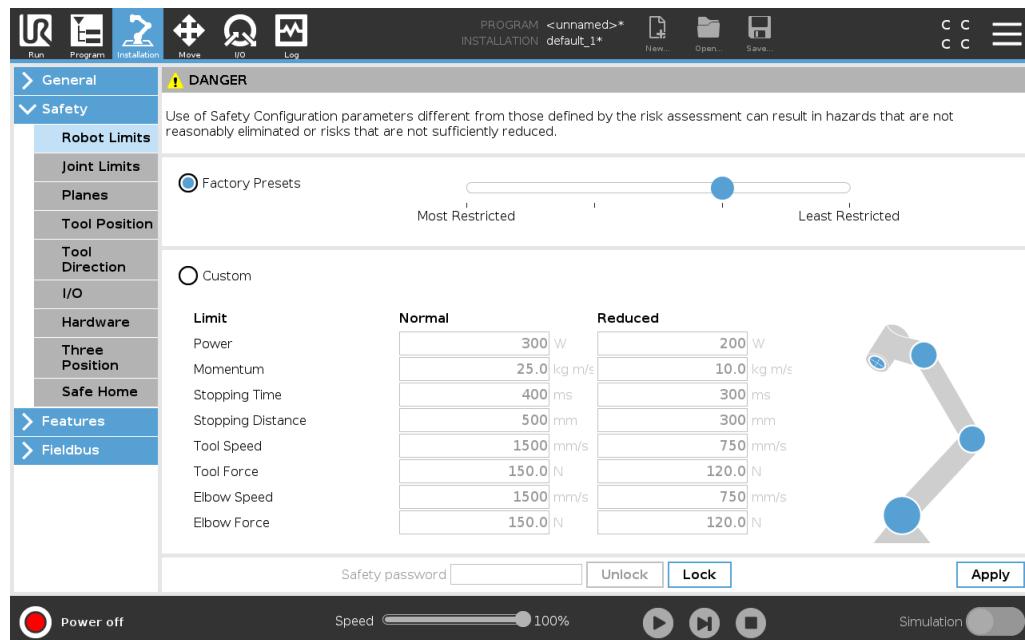
## 21.7. Robot Limits

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

1. **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**


**NOTE**

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



2. 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.


**NOTE**

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 the maximum force exerted by the robot tool in clamping situations.

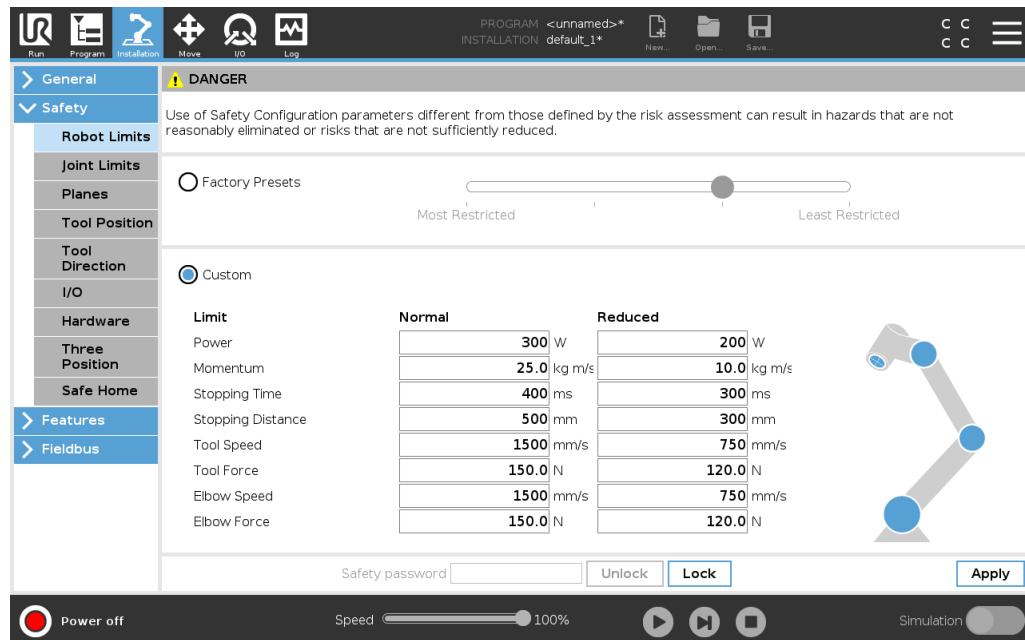
### Elbow Speed

limits maximum robot elbow speed.

### Elbow Force

limits maximum force that the elbow exerts on the environment.

The tool speed and force are limited at the tool flange and the center of the two user-defined tool positions, see [21.14. Tool Position on page 125](#).



#### NOTE

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

## 21.8. Safety Modes

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 mode** is the safety mode that is active by default

**Reduced mode** is active when the robot **Tool Center Point** (TCP) is positioned beyond a Trigger Reduced mode plane (see [21.11. Planes on page 121](#)), or when triggered using a configurable input (see [21.16. I/O on page 128](#))

**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 violated already when the robot arm is powered on, it starts up in **Recovery** mode. This makes it possible to move the robot arm back within the safety limits. While in Recovery mode, the movement of the robot arm is restricted by a fixed limit that you cannot customize. For details about Recovery mode limits (see [Hardware Installation Manual](#)).

**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 Normal and Reduced mode. For the tool and joints, Reduced mode limits for speed and momentum are required to be more restrictive than their Normal mode counterparts.

## 21.9. Tolerances

In the Safety Configuration the safety system limits are specified. The *Safety System* receives the values from the input fields and detects any violation if any these values are exceeded. The robot controller attempts to prevent any violations by making a protective stop or by reducing the speed. This means that a program might not be able to perform motions very close to a limit.

**WARNING**

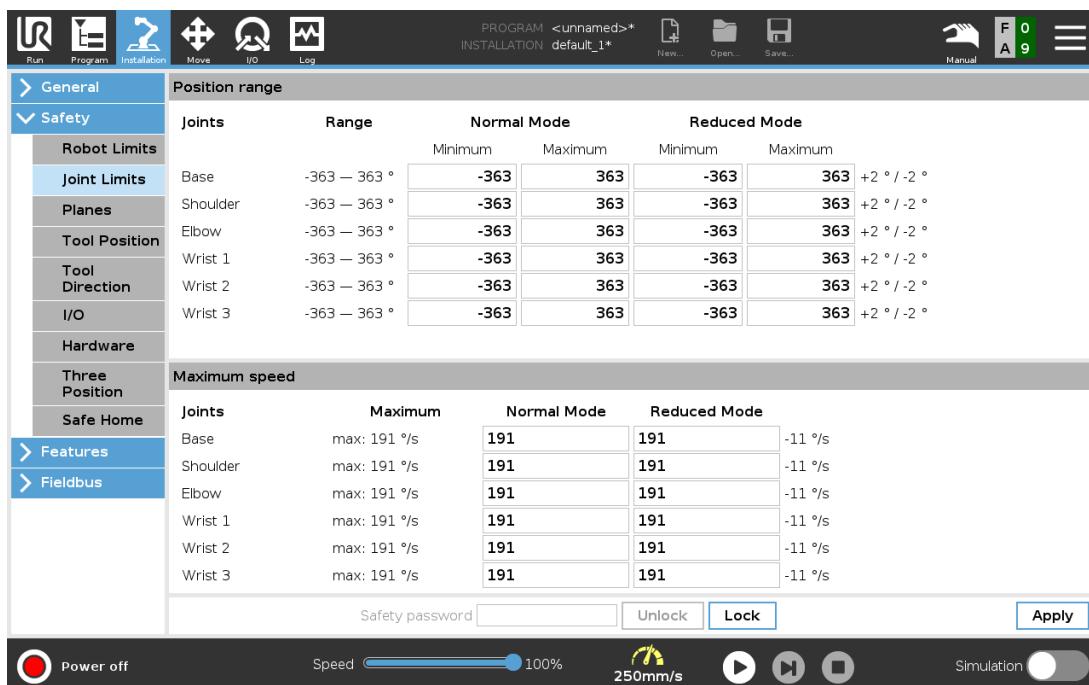
Tolerances are specific to Software version. Updating software may change tolerances. Consult the release notes for information about Software version changes.

## 21.10. Joint Limits

Joint Limits allow you to restrict individual robot joint movements in joint space i.e. joint rotational position and joint rotational speed. There are two Joint Limits options: **Maximum speed** and **Position range**.

The Wrist 3 position range is unlimited by default. When using cables attached to the robot, you must first disable the **Unrestricted Range for Wrist 3** checkbox to avoid cable tension and protective stops.

1. Maximum speed is where you define the maximum angular velocity for each joint.
2. Position range is where you define the position range for each joint. Again, the input fields for Reduced mode are disabled if there is no safety plane or configurable input set to trigger it. This limit enables safety-rated soft axis limiting of the robot.



## 21.11. Planes



### NOTE

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

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 by deselecting the checkbox. Before configuring safety planes, you must define a feature in the robot installation (see ). 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.

### 21.11.1. 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 in Normal mode, a normal plane is active and it acts as a strict limit on the position.

#### *Reduced*

When the safety system is in Reduced mode, a reduced mode plane is active and it acts as a strict limit on the position.

#### *Normal & Reduced*

When the safety system is either in Normal or Reduced mode, a normal and reduced mode plane is active and acts as a strict limit on the position.

#### *Trigger Reduced Mode*

The safety plane causes the safety system to switch to Reduced mode 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*

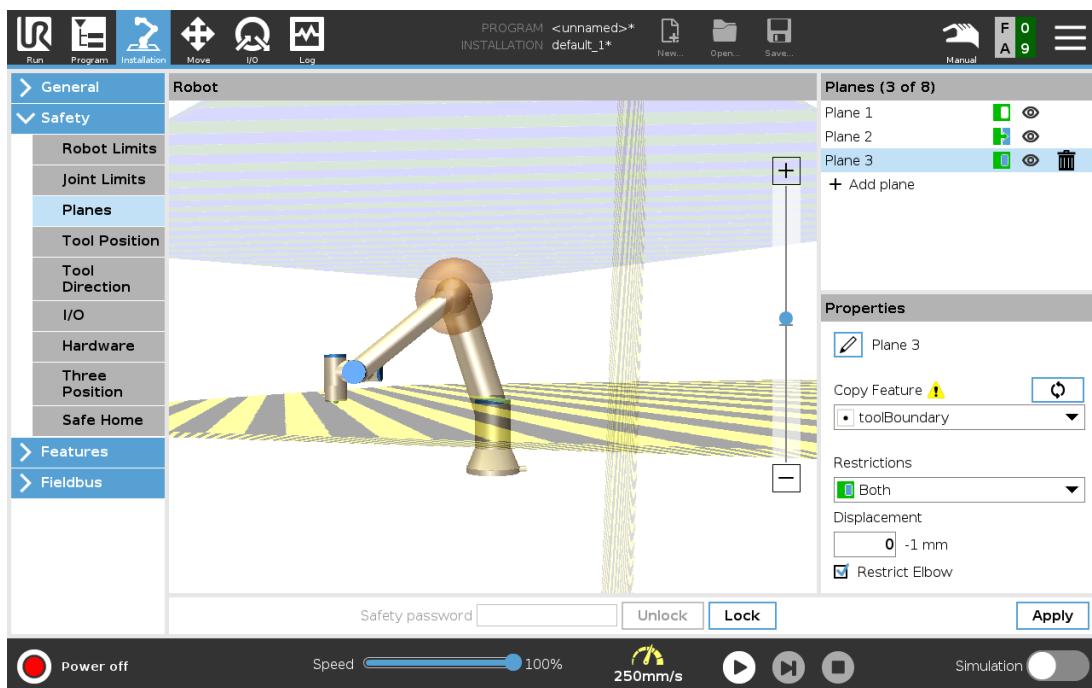
Pressing this icon allows you to rename the plane.

## 21.11.2. 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.

In **Copy Feature**, only **Undefined** and **Base** are available. You can reset a configured safety plane by selecting **Undefined**

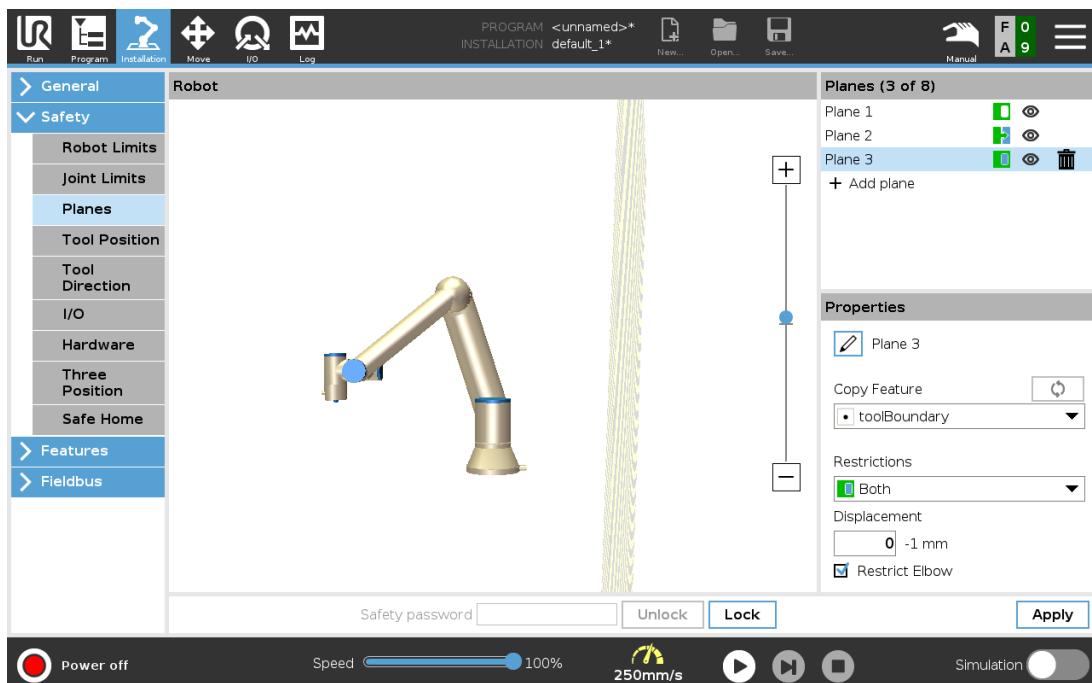
If the copied feature is modified in the Features screen, a warning icon appears to the right of the Copy Feature text. This indicates that the feature is out of sync i.e. the information in the properties card is not updated to reflect the modifications that may have been made to the Feature.



### 21.11.3. Elbow

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.

### 21.11.4. 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)

## 21.12. Freedrive

Freedrive allows the robot arm to be manually pulled into desired positions and/or poses. The joints move with little resistance because the brakes are released. While the robot arm is being moved manually, it is in Freedrive mode (see [20.1. Operational Modes on page 111](#)). As the robot arm in Freedrive Mode approaches a predefined limit or plane (see [21.11. Planes on page 121](#)), resistance increases. This makes pulling the robot into position feel heavy. You can enable Freedrive in the following ways:

- Use the Freedrive Button
- Use I/O Actions (see )
- Use the Freedrive button on the Move Tab screen (see [25.2.4. Freedrive on page 230](#))

### 21.12.1. Using the Freedrive Button

1. Press and hold down the **Freedrive Button** located on **Teach Pendant**.
2. Move the robot arm as desired.

## 21.13. 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 tremors are undesirable and the **Backdrive** feature can be used to forcefully move specific joints to a desired position without releasing all brakes in the robot arm.

### 21.13.1. Enabling Backdrive

1. On the **Initialize** screen, tap **ON** to start the power up sequence.
2. When the robot state is **Idle**, press and hold the **Freedrive** button. The robot state changes to **Backdrive**.

3. Brakes are only released in the joints to which significant pressure is applied. As long as the **Freedrive** button is engaged/pressed. While using **Backdrive**, the robot arm feels heavy to move around.

## 21.14. Tool Position

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

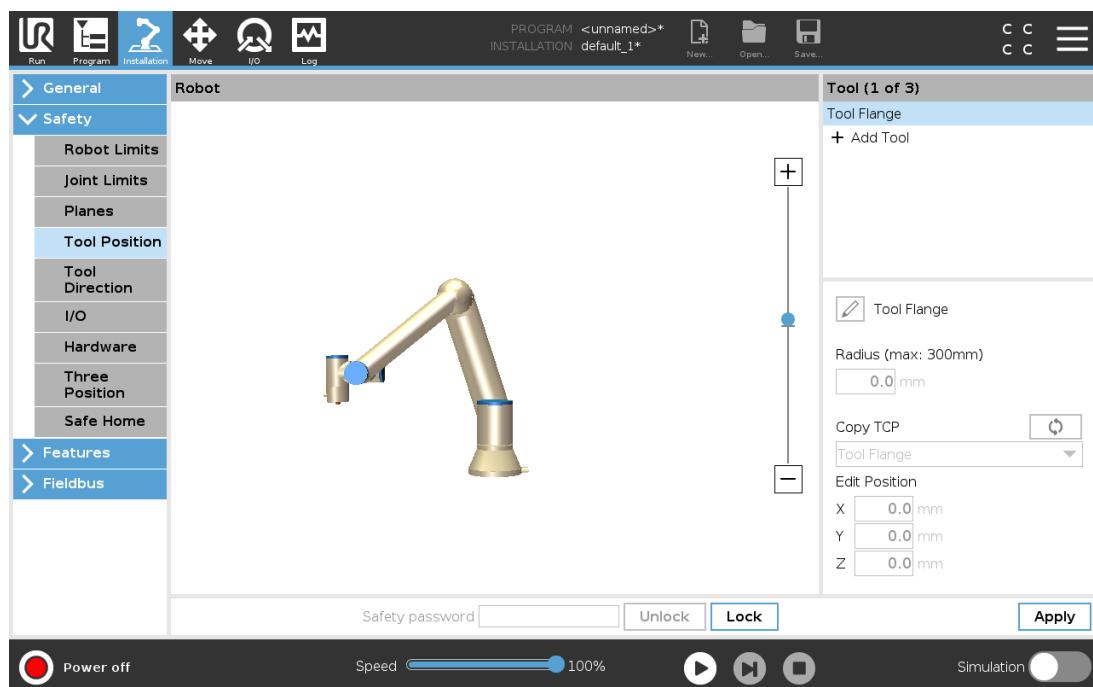
**Robot** is where you can visualize your modifications.

**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.



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 mode trigger plane, the robot switches to *Reduced* mode. The safety system prevents any point on the sphere from passing a safety plane (see [21.11. Planes on page 121](#)).

**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.

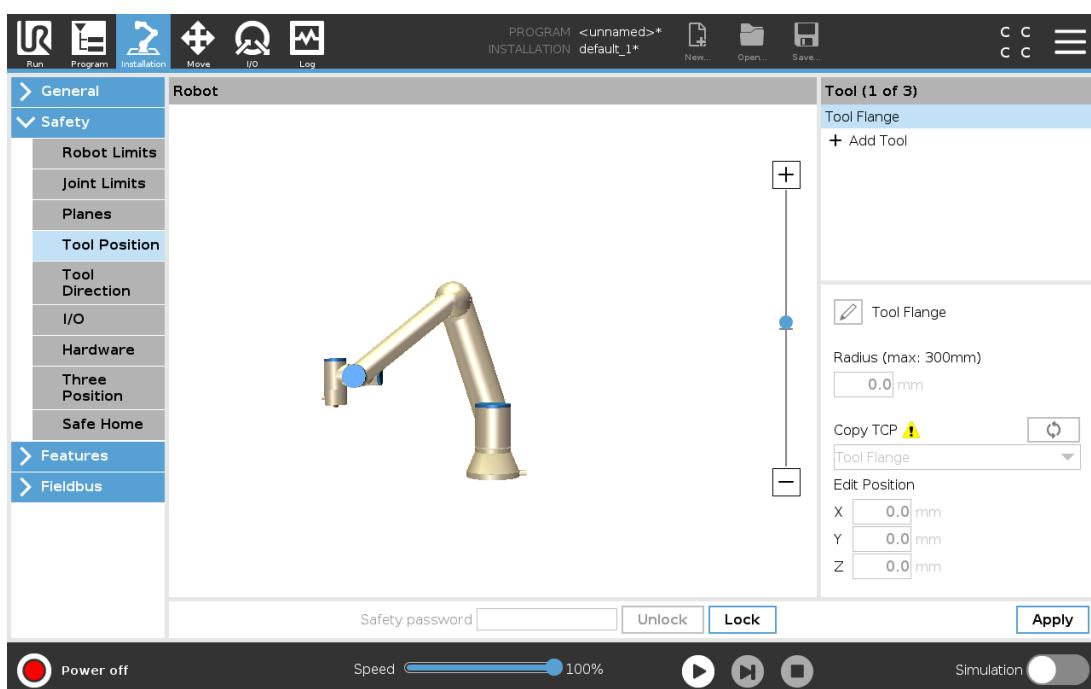
When you edit or adjust the values in the **Edit Position** input fields, the name of the TCP visible in the

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 (see [24.2. TCP Configuration on page 197](#)).

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.



## 21.15. Tool Direction

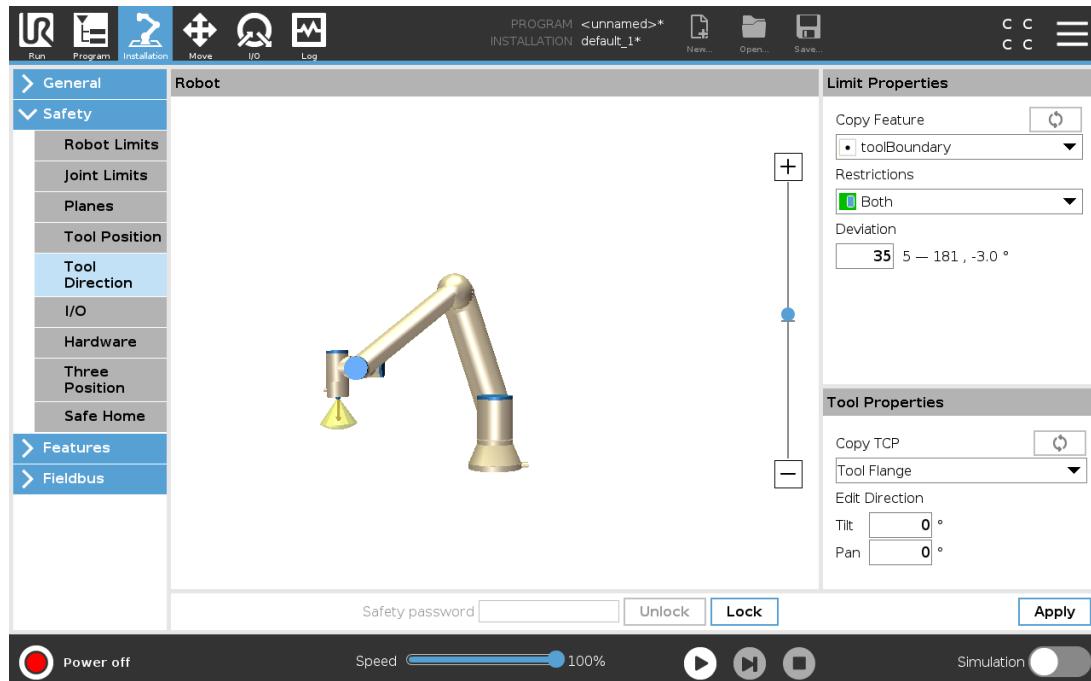
The Tool Direction screen can be used to restrict the angle in which the tool is pointing. The limit is defined by a cone that has a fixed orientation with respect to the robot arm Base. As the robot arm moves around, tool direction is restricted so it remains within the defined cone. The default direction 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 (see [24.17. Features on page 215](#)). The feature can then be copied and its Z axis used as the center of the cone defining the limit.



### NOTE

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.



## 21.15.1. 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*

is never active

*Normal Tool direction limit*

is active only when safety system is in **Normal mode**.

*Reduced Tool direction limit*

is active only when the safety system is in **Reduced mode**.

*Normal & Reduced Tool direction limit*

is 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".

## 21.15.2. 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:

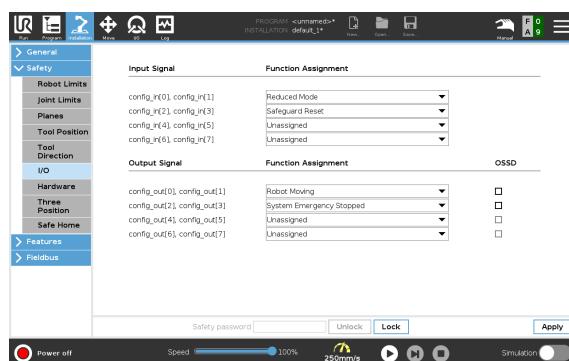
**Tilt angle:** How much to tilt the Z axis of the output flange towards the X axis of the output flange

**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.

## 21.16. I/O

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.



### 21.16.1. Input Signals

The following Safety Functions can be used with the input signals:

#### *System Emergency Stop*

This is an emergency stop button alternative to the one on the Teach Pendant, providing the same functionality if the device complies with ISO 13850.

#### *Reduced Mode*

All safety limits can be applied in either Normal mode or Reduced mode (see [21.8. Safety Modes on page 119](#)). When configured, a low signal sent to the inputs causes the safety system to transition to Reduced mode. The robot arm decelerates to satisfy the Reduced mode limit set. The safety system guarantees that the robot is within Reduced mode limits less than 0.5s after the input is triggered.

If the robot arm continues to violate any of the Reduced mode limits, it performs a Stop Category 0. Transition to Normal mode occurs in the same way. The trigger planes can also cause a transition to Reduced mode.

#### *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.

#### *Operational Mode*

When defined, this input can be used to switch between **Automatic Mode** and **Manual Mode** (see [20.1. Operational Modes on page 111](#)).

#### *Safeguard Reset*

When a Safeguard Stop occurs, this output ensures that the Safeguard Stop state continues until a reset is triggered.

#### *Automatic Mode Safeguard Stop*

Once configured, an **Automatic Mode Safeguard Stop** performs a Safeguard Stop when the input pins are low and ONLY when the robot is in Automatic mode.

#### *Automatic Mode Safeguard Reset*

When an Automatic Mode Safeguard Stop occurs, the robot remains safeguard stopped in Automatic Mode until a rising edge on the input pins trigger a reset.



#### **WARNING**

- If you disable the default Safeguard Reset input, the Robot Arm is no longer Safeguard Stop stopped as soon as the input is high. A program paused only by the Safeguard stop resumes.
- Similar to the Safeguard Reset, if the default Automatic Mode Safeguard Reset is disabled, the Robot Arm is no longer Safeguard Stop stopped once the Automatic Mode Safeguard Stop input is high. A program paused only by the Automatic Mode Safeguard Stop resumes.

## 21.16.2. Output Signals

You can apply the following Safety functions for output signals. All signals return to low when the state which triggered the high signal has ended:

#### *System Emergency Stop*

Signal is *Low* when the safety system has been triggered into an Emergency Stopped state by the Robot Emergency Stop input or the Emergency Stop Button. To avoid deadlocks, if the Emergency Stopped state is triggered by the System Emergency 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 Mode*

Signal is *Low* when the robot arm is placed in Reduced mode or if the safety input is configured with a Reduced Mode input and the signal is currently low. Otherwise the signal is high.

#### *Not Reduced Mode*

This is the inverse of the Reduced Mode defined above.

#### Safe Home

Signal is *High* if the Robot Arm is stopped in the configured Safe Home Position. Otherwise, the signal is *Low*.



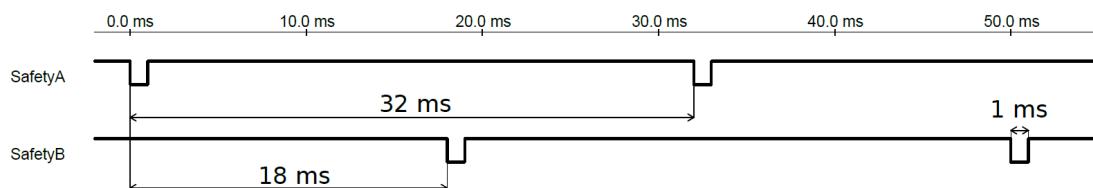
#### NOTE

Any external machinery receiving its Emergency Stop state from the robot through the System Emergency 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 Emergency 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.

### 21.16.3. OSSD Safety Signals

You can configure the Control Box to output OSSD pulses when a safety output is inactive/high. OSSD pulses detect the ability of the Control Box to make safety outputs active/low. When OSSD pulses are enabled for an output, a 1ms low pulse is generated on the safety output once every 32ms. The safety system detects when an output is connected to a supply and shuts down the robot.

The illustration below shows: the time between pulses on a channel (32ms), the pulse length (1ms) and the time from a pulse on one channel to a pulse on the other channel (18ms)



#### Enabling OSSD for Safety Output

1. In the Header, tap **Installation** and select **Safety**.
2. Under **Safety**, select **I/O**.
3. On the I/O screen, under Output Signal, select the desired OSSD checkbox. You must assign the output signal to enable the OSSD checkboxes.

## 21.17. Hardware

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.

## 21.17.1. Selecting Available Hardware

The robot can be used without PolyScope as the programming interface.

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.

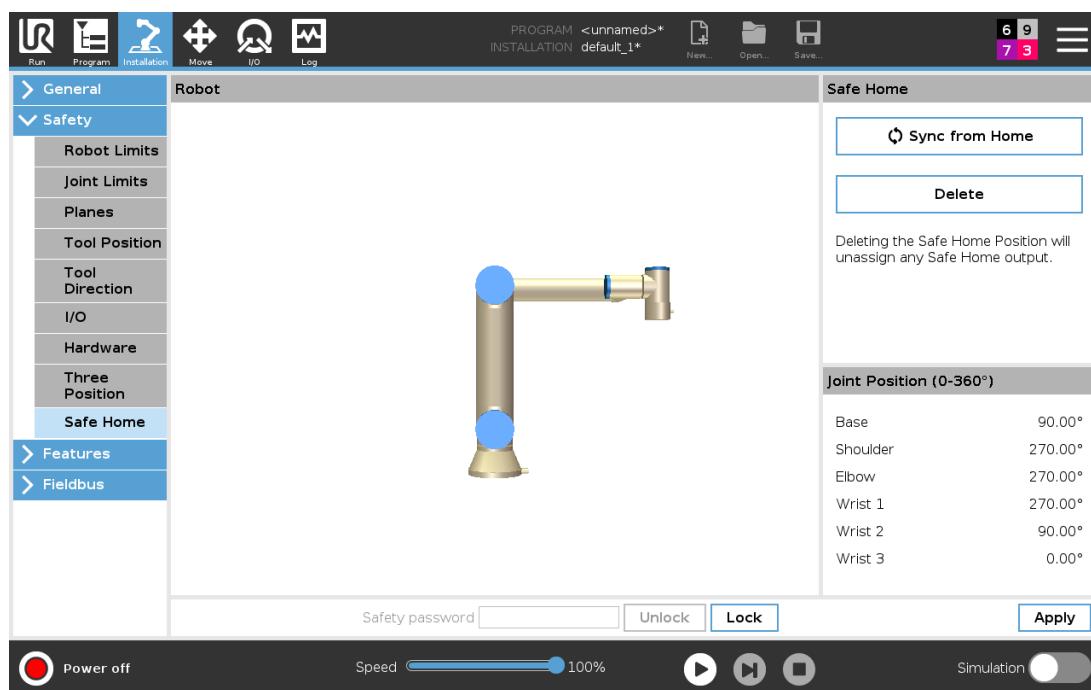


### CAUTION

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.

## 21.18. Safe Home Position

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.



## 21.18.1. Syncing 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**.

## 21.19. Safe Home Output

The Safe Home Position must be defined before the Safe Home Output (see [21.16.2. Output Signals on page 129](#)).

### 21.19.1. Defining 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**.

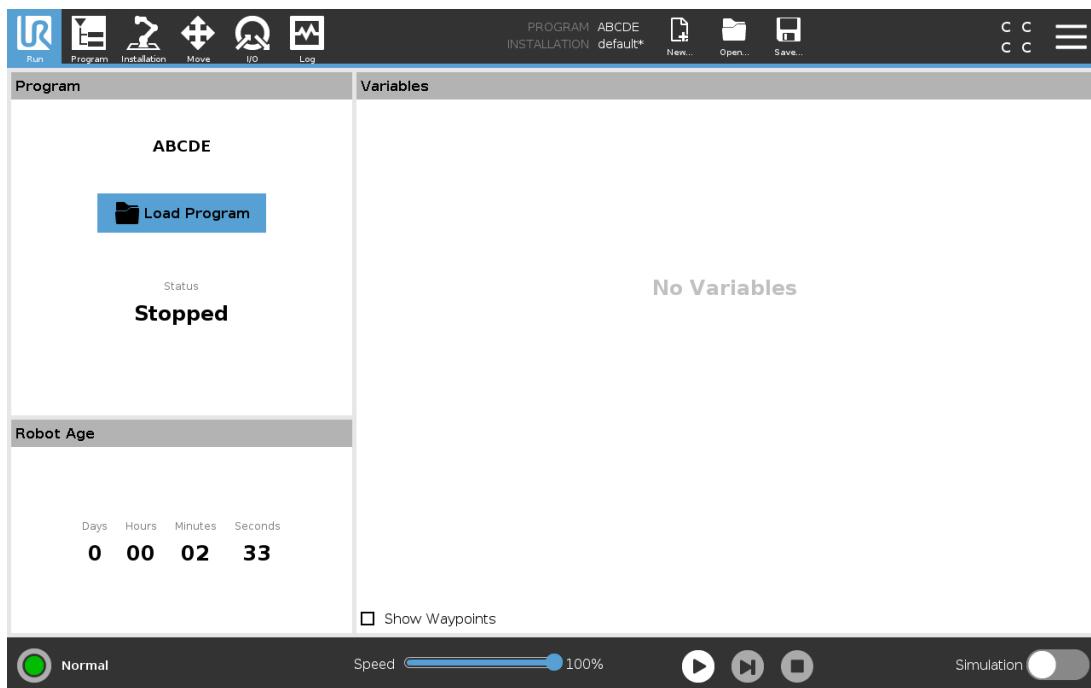
## 21.20. Edit Safe Home

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

### 21.20.1. Editing Safe Home

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** (See [21.2. Setting a Safety Password on page 116](#)).
5. Under **Safe Home**, tap **Sync from Home**

# 22. Run Tab



The **Run** tab allows you to simply operate the robot arm and control box, using as few buttons and options as possible. You can combine simple operation with password protecting the programming part of PolyScope (see [29.3. Settings on page 247](#)), to make the robot into a tool that can run exclusively pre-written programs.

On this screen you can automatically load and start default a program based on an external input signal edge transition (see [24.7. Startup on page 207](#)).

The combination of auto loading and starting of a default program and auto initialization on power up can be used to integrate the robot arm into other machinery.

## 22.1. Program

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.

## 22.2. Variables

A robot program can make use of variables to store and update various values during runtime. Two kinds of variables are available:

### *Installation variables*

These can be used by multiple programs and their names and values are persisted together with the robot installation (see [24.6. Installation Variables on page 206](#)). 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

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.

#### Variable types

<i>bool</i>	A boolean variable whose value is either <code>True</code> or <code>False</code> .
<i>int</i>	A whole number in the range from – 2147483648 to 2147483647 (32 bit).
<i>float</i>	A floating point number (decimal) (32 bit).
<i>string</i>	A sequence of characters.
<i>pose</i>	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 as $[x, y, z, rx, ry, rz]$ .
<i>list</i>	A sequence of variables.

## 22.3. Robot Age

This field represents the length of time since the robot was initially turned on. The numbers in the field are not associated with program run time

## 22.4. Move Robot into Position

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.

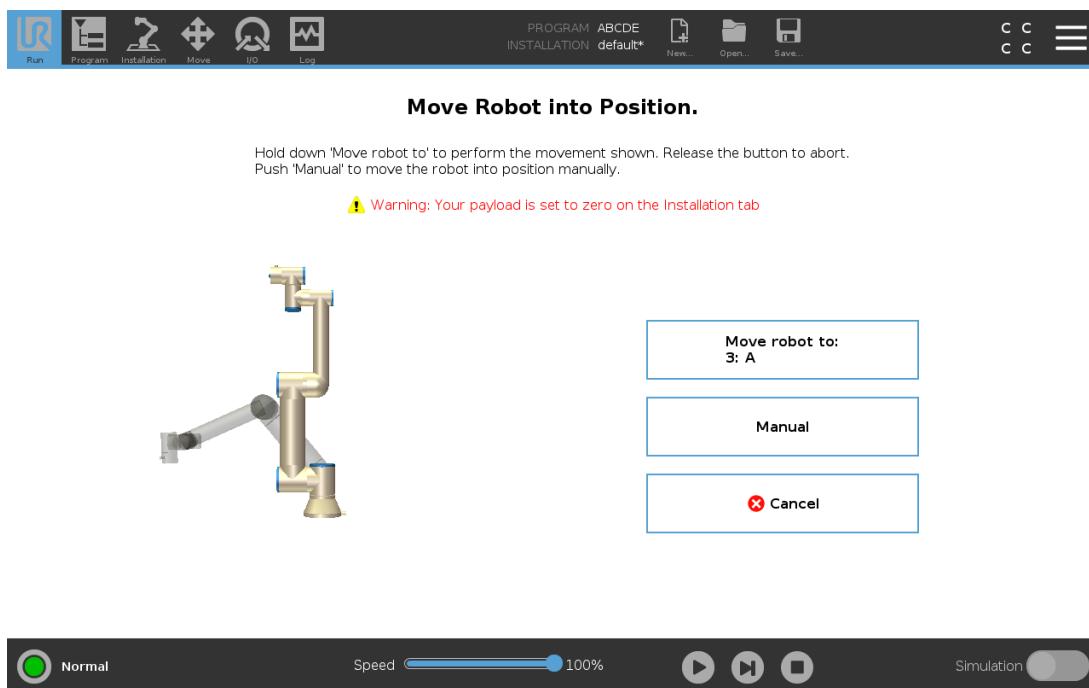
In cases where the **Move Robot into Position** screen cannot move the Robot Arm to the program start position, it moves to the first waypoint in the program tree.

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.

### 22.4.1. Accessing the Move Robot into Position Screen

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



## 22.4.2. 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.



### NOTE

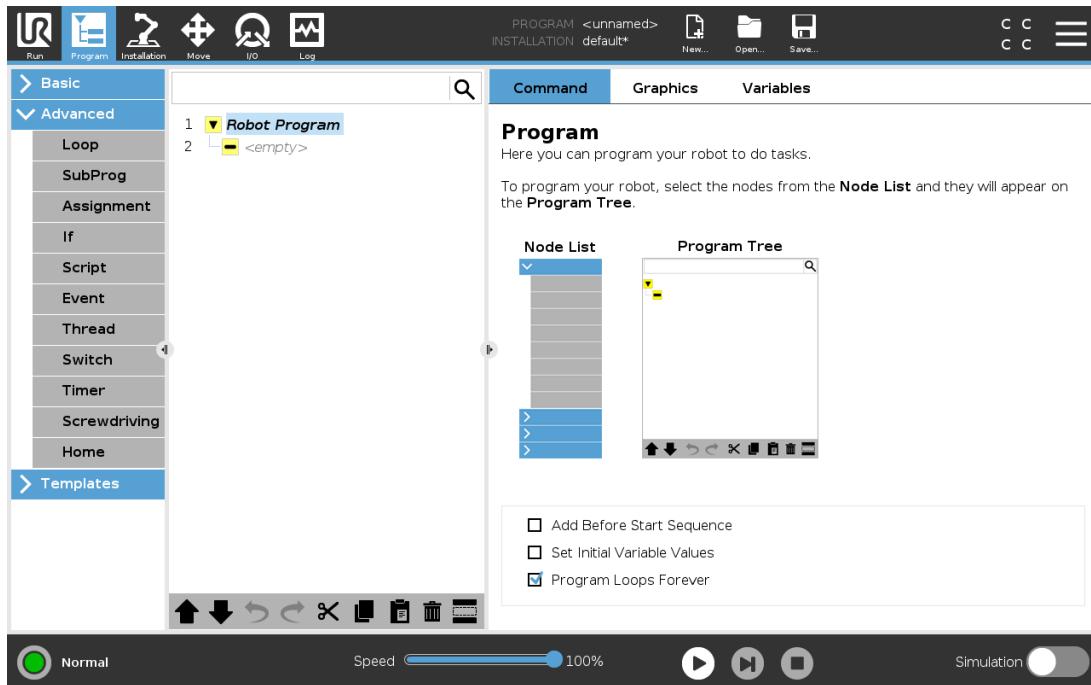
Collision can damage the robot or other equipment. Compare the animation with the position of the real Robot Arm to ensure the Robot Arm can safely perform the movement without colliding with any obstacles.

## 22.4.3. 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.



# 23. Program Tab



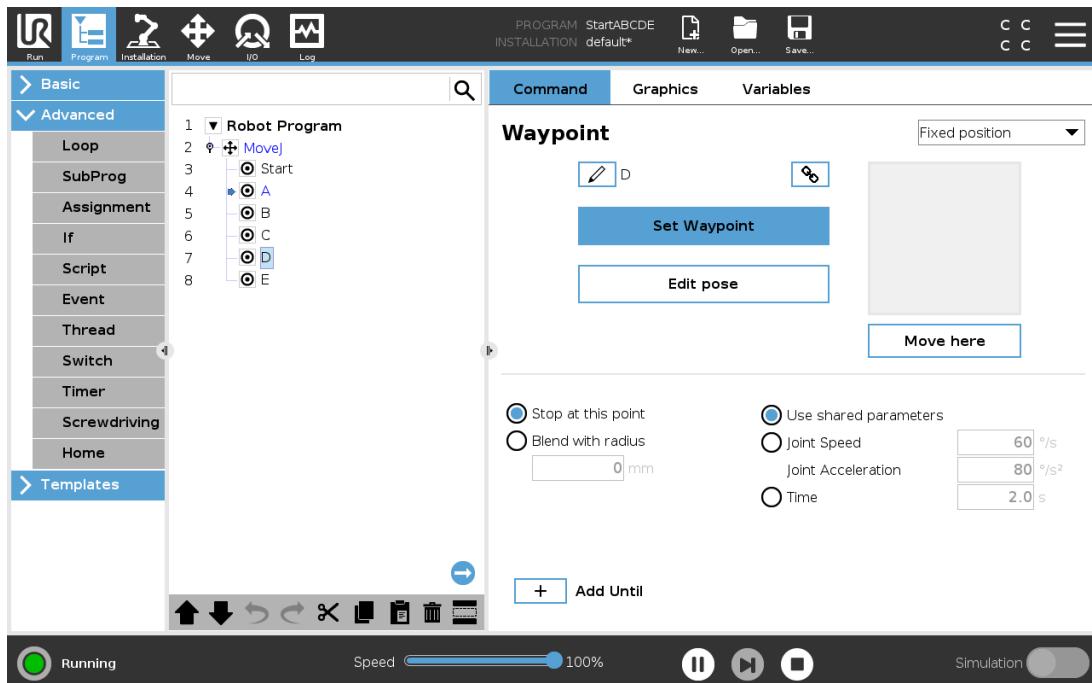
The program tab shows the current program being edited.

## 23.1. Program Tree

By tapping **Command** you add program nodes to the Program Tree. Configure the functionality of the added program nodes on the right side of the screen.

An empty Program Tree is not allowed to run. Programs containing mis-configured program nodes are also not allowed to run. Invalid program nodes are highlighted in yellow to indicate what should be fixed before the program is allowed to run.

## 23.1.1. Program Execution Indication



When the program is running, the program node currently being executed is indicated by a small icon next to the node. Furthermore, the path of execution is highlighted using a blue color.

Pressing the icon at the corner of the program will make it track the command being executed.

## 23.1.2. Search Button

Tap the to search in the Program Tree. Press the to exit search.

## 23.2. Program Tree Toolbar

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

### 23.2.1. Undo/Redo Buttons

The and buttons serve to undo and redo changes to commands.

### 23.2.2. Move Up & Down

The and buttons change the position of a node.

### 23.2.3. Cut

The button cuts a node and allows it to be used for other actions (e.g., paste it on other place on the Program Tree).

## 23.2.4. Copy

The  button allows copies a node and allows it to be used for other actions (e.g., paste it on other place on the Program Tree).

## 23.2.5. Paste

The  button allows you to paste a node that was previously cut or copied.

## 23.2.6. Delete

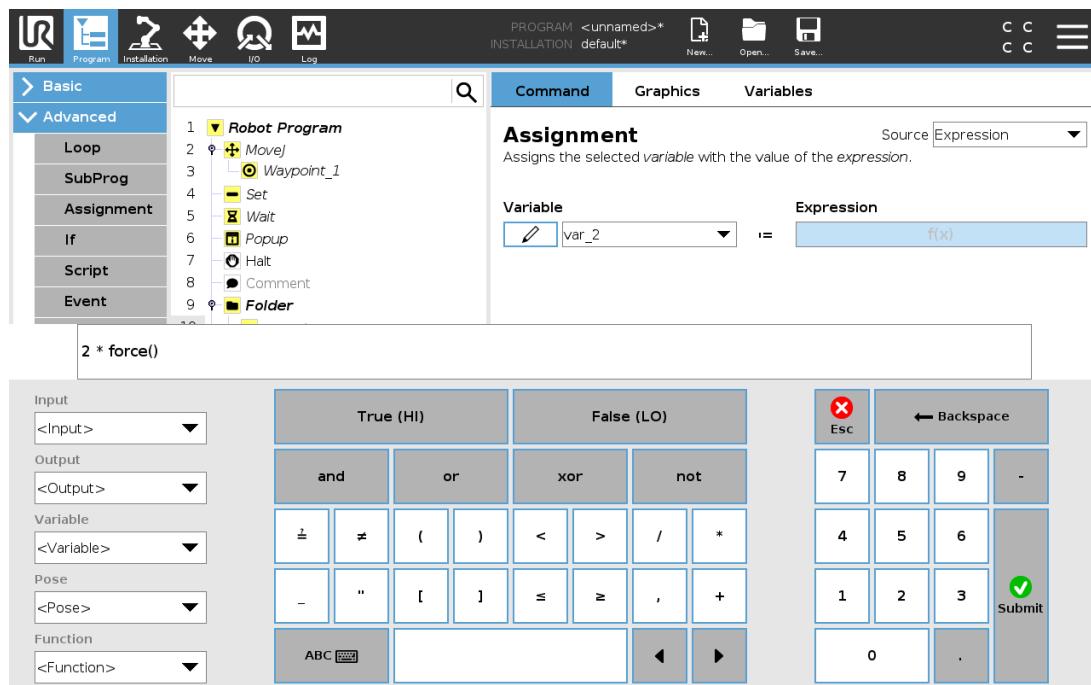
Tap the  button to remove a node from the Program Tree.

## 23.2.7. Suppress

Tap the  button to suppress specific nodes on the Program Tree.

Suppressed program lines are simply skipped when the program is run. A suppressed line can be unsuppressed again at a later time. This is a quick way to make changes to a program without destroying the original contents.

## 23.3. Expression Editor



While the expression itself is edited as text, the expression editor has a number of buttons and functions for inserting the special expression symbols, such as  $*$  for multiplication and  $\leq$  for less than or equal to. The keyboard symbol button in the top left of the screen switches to text-editing of the expression. 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**.

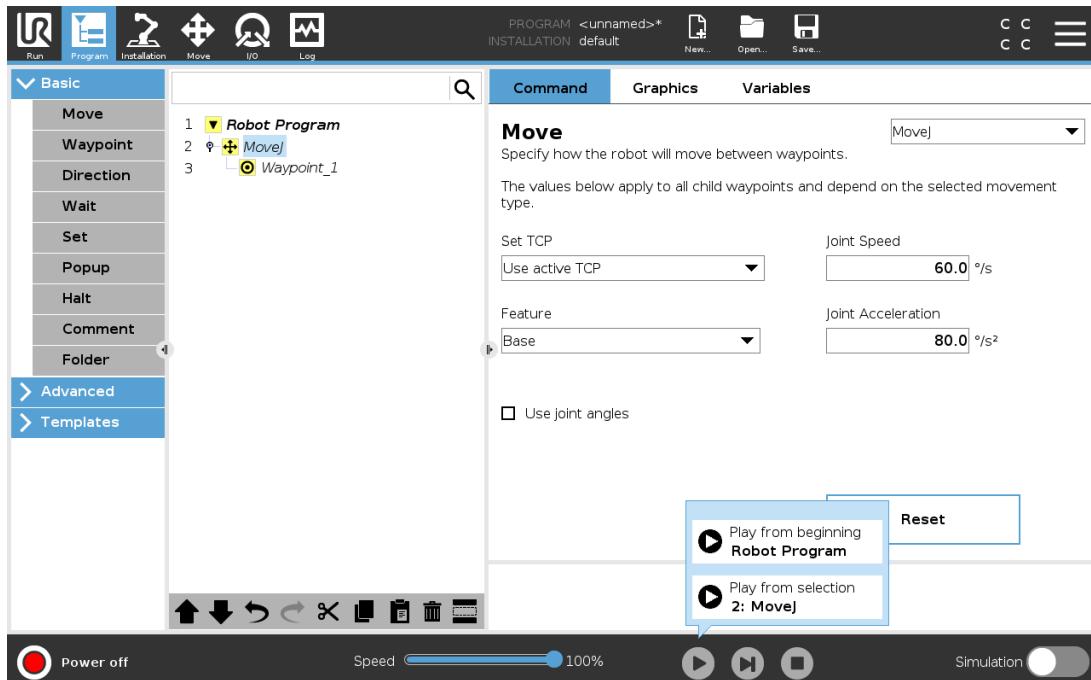
The expression is checked for grammatical errors when the **OK** button is pressed. The **Cancel** button leaves the screen, discarding all changes.

An expression can look like this:

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

## 23.4. Starting a Program from a Selected Node

When the robot is in Manual Mode (see [20.1. Operational Modes on page 111](#)), **Play from Selection** allows a program to start from a selected node. **Play from Beginning** runs a program normally. The **Play from Selection** option is disabled if a program cannot be run from a particular node. Play from Selection cannot be enabled with a thread because threads always start from the beginning.



### 23.4.1. Using Play from Selection

Tap **Play** and select **Play from Selection** to run a program from a node in the program tree.



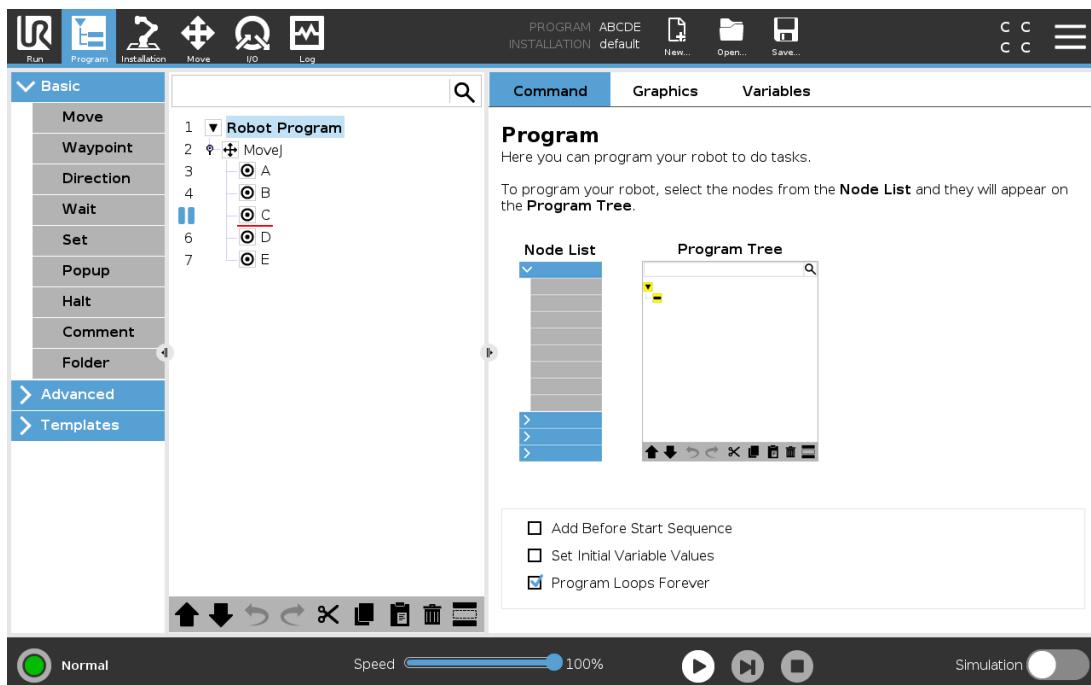
#### NOTE

- The **Before start** section, if used, is always executed for both **Play from Selection** and **Play from Beginning**.
- The program stops and displays an error message if an unassigned variable is encountered.
- A program can only start from a node in the Robot Program.
- **Play from Selection** can be used within a subprogram. The program execution halts when the subprogram ends.

## 23.5. Using Breakpoints in a Program

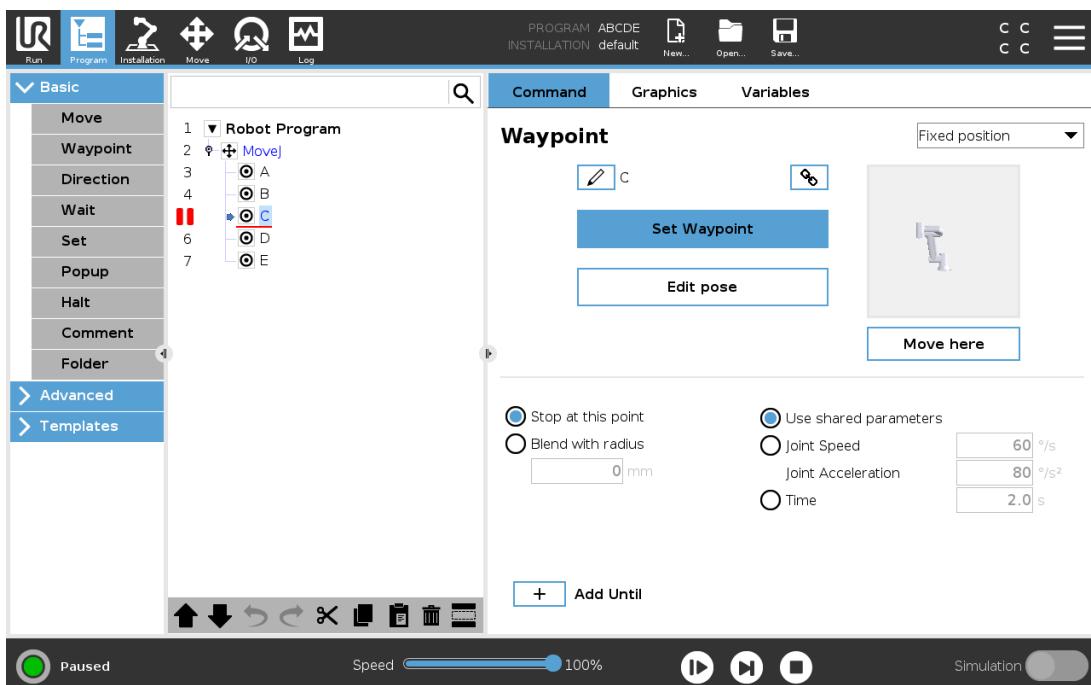
A breakpoint pauses program execution. You can use breakpoints to pause and resume a program at a specific point to inspect robot position, variables etc. See [20.1. Operational Modes on page 111](#).

1. In a Program tree, tap a line number to set or clear a breakpoint.



A red line either above or below a node, indicates when a breakpoint is set, pausing execution. Most nodes pause before they are executed, with the following exceptions:

- Waypoints: A breakpoint on a waypoint node, ignores the blend and pauses the program when the robot reaches this waypoint.
- Until nodes: A breakpoint on an until node pauses the program once the until condition is met. Blends used in the until node are not ignored. They are paused when the robot reaches the blend radius.

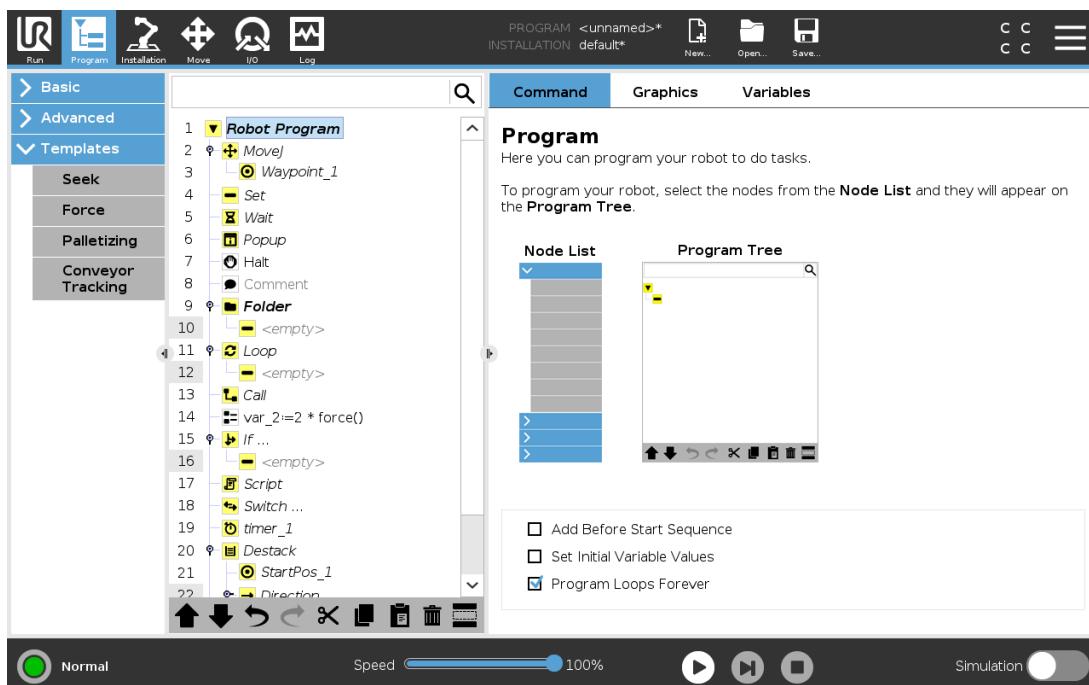


## 23.6. Single Step in a Program

The Single Step button allows one node to be executed at a time, when the robot is in Manual mode. You can use the Single Step button when the program is paused. Tap the Single Step button for the program to continue execution, and pause when reaching the following node in the program. If a node does not support Breakpoints, program execution initiated by tapping the Single Step button does not pause on that node. Instead, the execution continues until the program reaches a node that supports Breakpoints.

## 23.7. Command Tab

This manual does not cover all the details about every type of Program Node. The Robot Program Node includes three check-boxes controlling the overall behavior of the program.



## Add Before Start Sequence

Select this check-box to add a special section to the program which is once when the program starts.

## Set Initial Variables Values

Select this to set initial values of program variables.

1. Select a variable from the dropdown list, or by 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 from previous run** to initialize the variable to the value found on the **Variables** tab (see [23.9. Variables Tab on page 145](#)).

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).

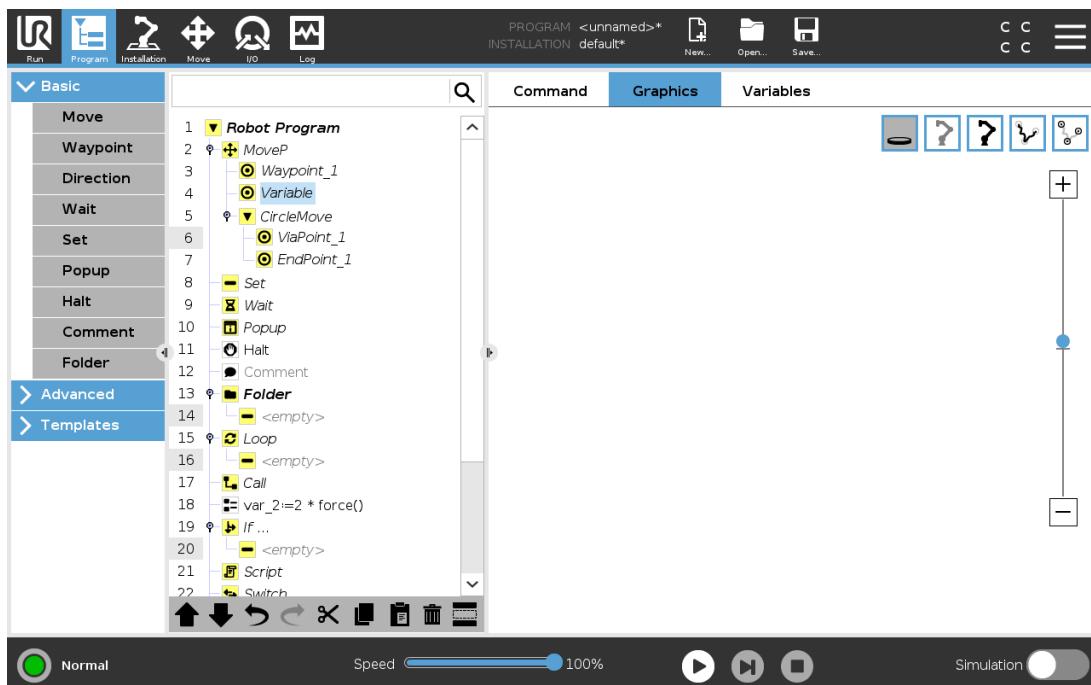
## Program Loops Forever

Select this to make the program continuous.

## 23.8. Graphics Tab

Graphical representation of the current robot program. The path of the TCP is shown in 3D view, with motion segments in black, and blend segments (transitions between motion segments) shown in green. The green dots specify the positions of the TCP at each of the waypoints in the program. The 3D

drawing of the robot arm shows the current position of the robot arm, and the *shadow* of the robot arm shows how the robot arm intends to reach the waypoint selected in the left hand side of the screen.



If the current position of the robot TCP comes close to a safety or trigger plane, or the orientation of robot tool is near the tool orientation boundary limit (see [21.11. Planes on page 121](#)), a 3D representation of the boundary limit is shown.

When the robot is running a program, the visualization of boundary limits will be disabled.

Safety planes are visualized in yellow and black with a small arrow representing the plane normal, which indicates the side of the plane on which the robot TCP is allowed to be positioned.

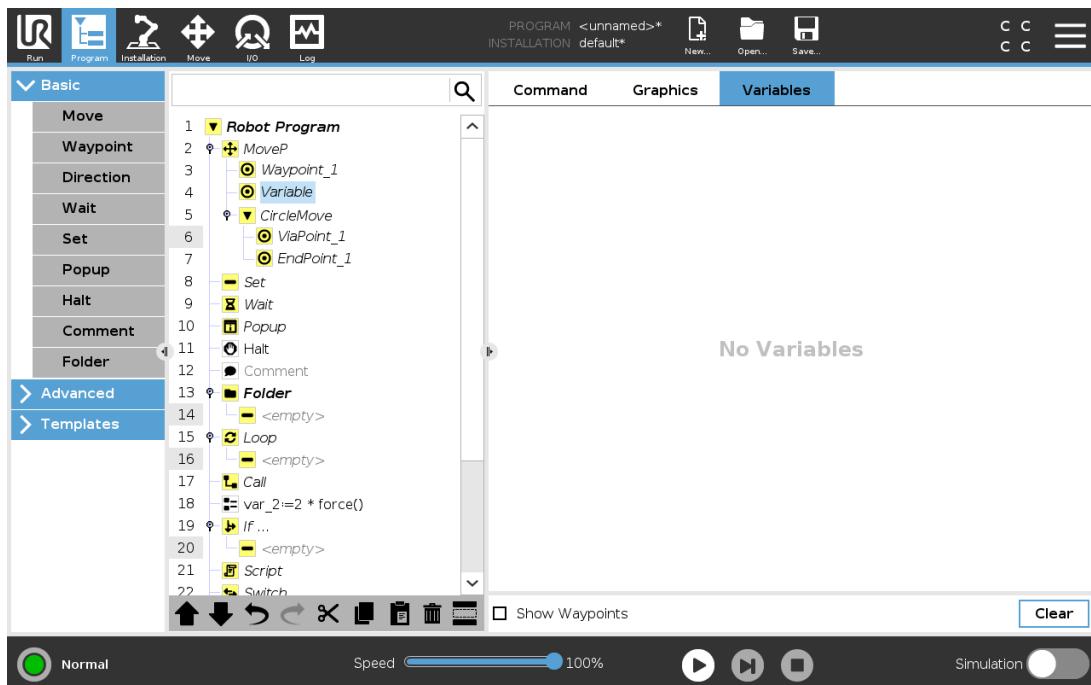
Trigger planes are displayed in blue and green and a small arrow pointing to the side of the plane, where the Normal mode limits (see [21.8. Safety Modes on page 119](#)) are active.

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 target robot TCP no longer is in the 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.

The 3D view can be zoomed and rotated to get a better view of the robot arm. The buttons in the top-right side of the screen can disable the various graphical components in 3D view. The bottom button switches on/off the visualization of proximate boundary limits.

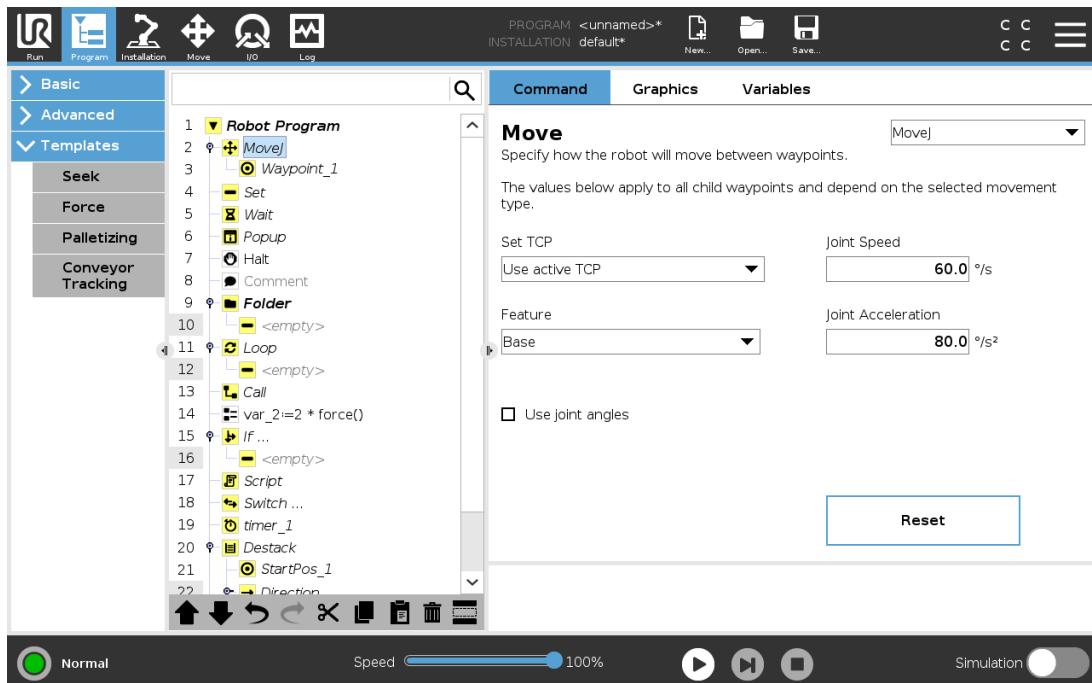
## 23.9. Variables Tab



The **Variables** tab shows the live values of variables in the running program, and keeps a list of variables and values between program runs. It only appears when it has information to display. Waypoint variables will be shown in the list if Show Waypoints is enabled.

## 23.10. Basic program nodes

### 23.10.1. Move



The **Move** command controls the robot motion through the underlying waypoints. Waypoints have to be under a Move command. The Move command defines the acceleration and the speed at which the robot arm will move between those waypoints.

### Movement Types

You can select one of three types of movements: **MoveJ**, **MoveL** and **MoveP**. Each movement type is explained below.

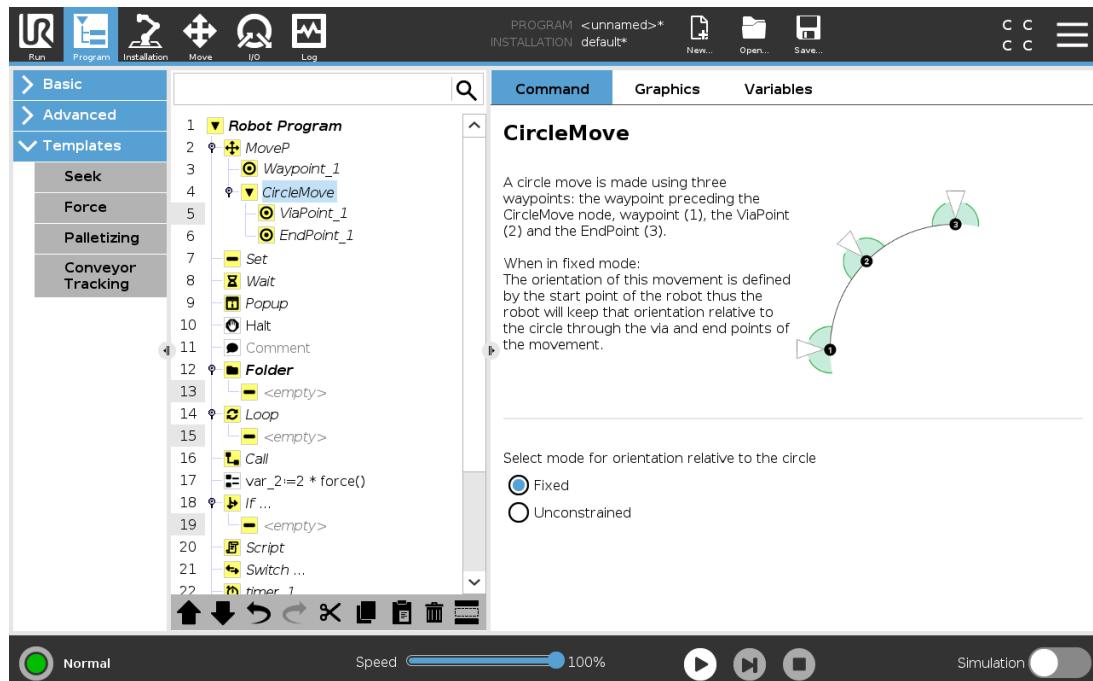
- **moveJ** makes movements that are calculated in the robot arm **joint space**. Joints are controlled to finish their movements at the same time. This movement type results in a curved path for the tool. The shared parameters that apply to this movement type are the maximum joint speed and joint acceleration, specified in  $deg/s$  and  $deg/s^2$ , respectively. If it is desired to have the robot arm move fast between waypoints, disregarding the path of the tool between those waypoints, this movement type is the preferable choice.
- **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. The shared parameters that can be set for this movement type are the desired tool speed and tool acceleration specified in  $mm/s$  and  $mm/s^2$ , respectively, and also a feature.
- **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 protective stop.

- **Circle move** can be added to a **moveP** to make a circular movement. The robot starts the movement from its current position or start point, moves through a **ViaPoint** specified on the circular arc, and 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 tool orientation
- Unconstrained: the start point transforms to the **EndPoint** to define tool orientation



## Shared parameters

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.

## TCP selection

The way the robot moves between waypoints is adjusted depending on whether the TCP is set using a user defined TCP or an active TCP. **Ignore Active TCP** allows this movement to be adjusted in relation to the Tool Flange.

## Setting 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**.

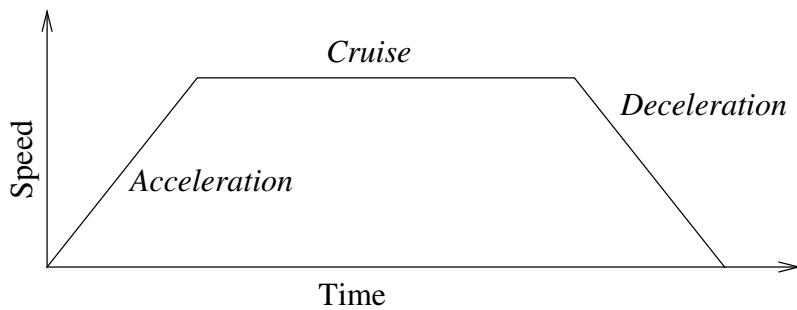
## Feature selection

The feature spaces the waypoints under the Move command, that should be represented when specifying these waypoints (see section [24.17. Features on page 215](#)). This means that when setting a waypoint, the program will remember the tool coordinates in the feature space of the selected feature. There are a few circumstances that need detailed explanation:

The selected 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.

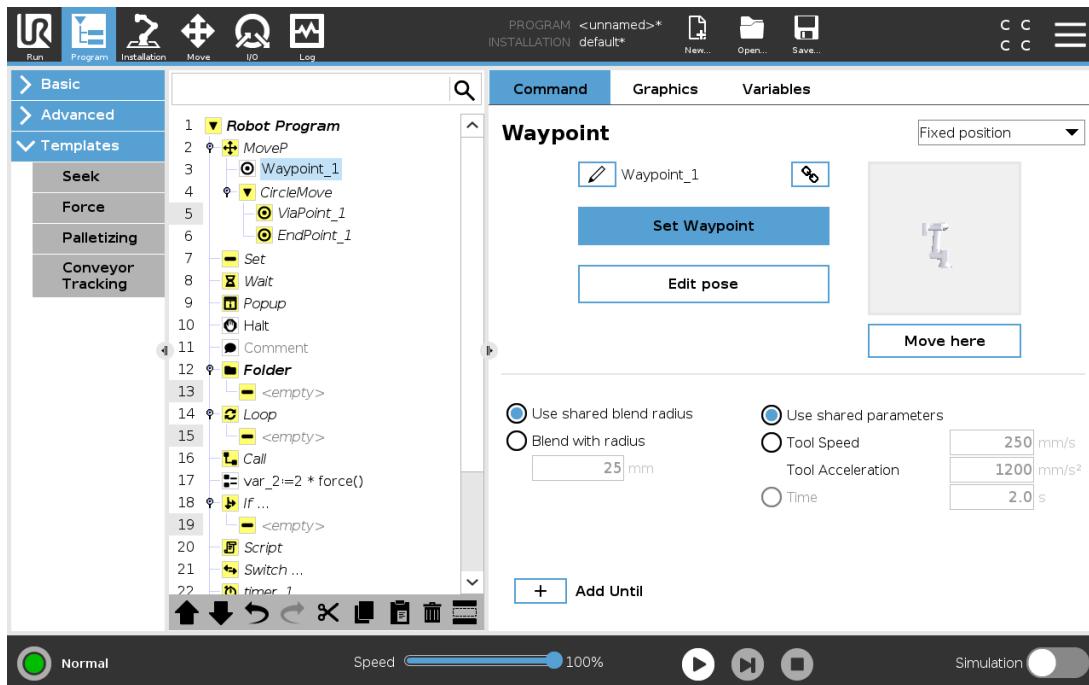


**12.1:** Speed profile for a motion. The curve is divided into three segments: acceleration, cruise and deceleration. The level of the cruise phase is given by the speed setting of the motion, while the steepness of the acceleration and deceleration phases is given by the acceleration parameter.

## Use joint angles

As an alternative to the 3D pose, you can select the **Use joint angles** checkbox when using the MoveJ to define waypoints using the robot joint angles. If **Use joint angles** is enabled, TCP and feature options are unavailable. Waypoints defined using **Use joint angles** are not adjusted when the program is moved between robots.

## 23.10.2. Fixed Waypoint



A point on the robot path. Waypoints are the most central part of a robot program, telling the robot arm where to be. A fixed position waypoint is taught by physically moving the robot arm to the position.

## Teaching Waypoints

Teaching is the term used to show the robot how to position the TCP in relation to a feature for an application. To teach the robot a waypoint, follow the instructions below:

1. In the Program Tab, insert a **Move Node**.
2. On the Move Node, use the **Set TCP** drop-down menu to set the TCP.
3. On the Move Node, use the **Feature** drop-down menu to select a feature.
4. On the Waypoint Node, use **Teach Mode** or **Jog** to position the robot in a desired configuration.

## Using Waypoints

Using a waypoint means applying the taught relation between the feature and the TCP in the present situation. The relation between the feature and the TCP, applied to the current selected feature, achieves the desired TCP location. Then the robot figures out how to position itself to let the currently active TCP reach that TCP position. To use a waypoint, follow the instructions below:

1. Use an existing waypoint in a Move Node, or insert the waypoint into a different Move Node (e.g. by copy and paste or use the "Link" button on the waypoint).
2. Set the desired TCP.
3. Set the desired feature.

## Setting the waypoint

### Waypoint names

Waypoints automatically get a unique name. The name can be changed by the user. By selecting the link icon, waypoints are linked and share position information. 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.

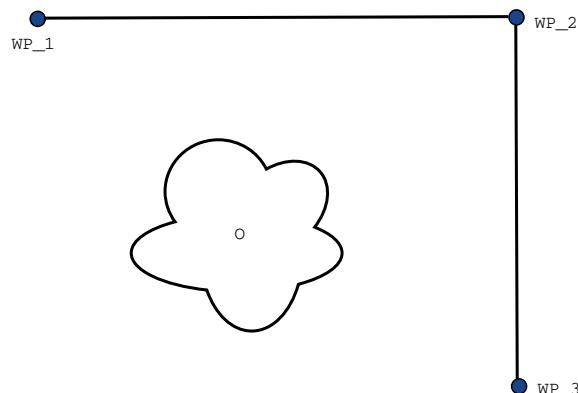
### Blending

Blending enables the robot to smoothly transition between two trajectories, without stopping at the waypoint between them.

### Example

Consider a pick and place application as an example (see figure 12.2), where the robot is currently 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 (O), the robot must approach (WP\_3) in the direction coming from Waypoint 2 (WP\_2).

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



12.2: (WP\_1) : initial position, (WP\_2) : via point, (WP\_3) : pick up position, (O) : obstacle.

Without configuring other settings, the robot will make a 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 happens near this position.

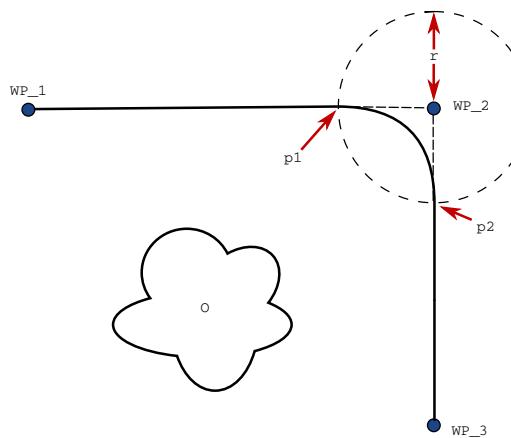
The stop at (WP\_2) can be avoided by configuring a blend for the waypoint, allowing the robot to calculate a smooth transition into the next trajectory. The primary parameter for the blend is a radius.

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.

## Blend parameters

Apart from the waypoints, multiple parameters will influence the blend trajectory (see figure 12.3):

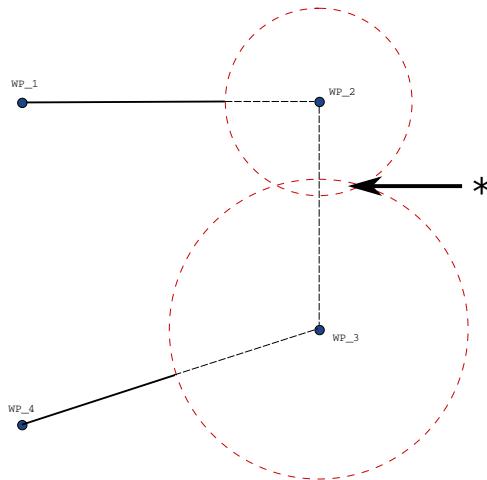
- the blend radius ( $r$ )
- the initial and final speed of the robot (at positions  $p_1$  and  $p_2$ , respectively)
- the movement time (e.g. if setting a specific time for a trajectory this will influence the initial/final speed of the robot)
- the trajectory types to blend from and to (`MoveL`, `MoveJ`)



12.3: Blend over ( $WP_2$ ) with radius  $r$ , initial blend position at  $p_1$  and final blend position at  $p_2$ . ( $O$ ) is an obstacle.

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 as shown in figure 12.4.



12.4: Blend radius overlap not allowed (\*).

## Conditional blend trajectories

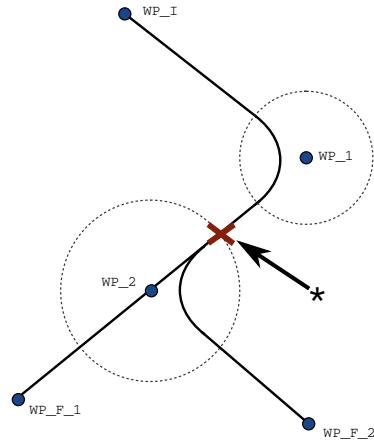
The blend trajectory is affected both by the waypoint where the blend radius is set and the following one in the program tree. That is, in the program in figure 12.5 the blend around (WP\_1) is affected by (WP\_2). The consequence of this becomes more apparent when blending around (WP\_2) in this example.

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 (or other necessary statements to determine the following waypoint, e.g. variable waypoints) is evaluated before we actually reach (WP\_2) 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.

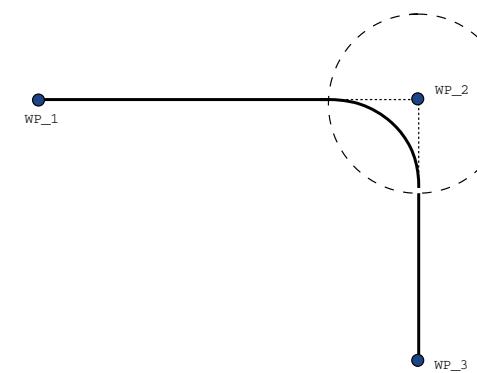
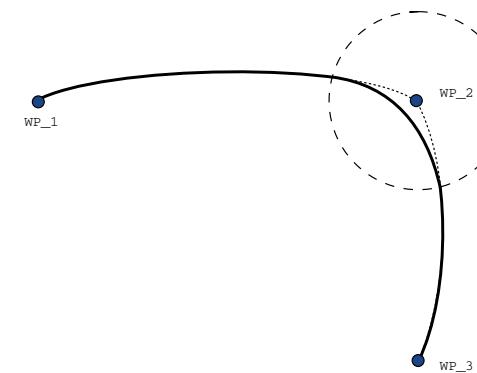
```

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



**12.5:** *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 (\*).

## Blend Trajectories

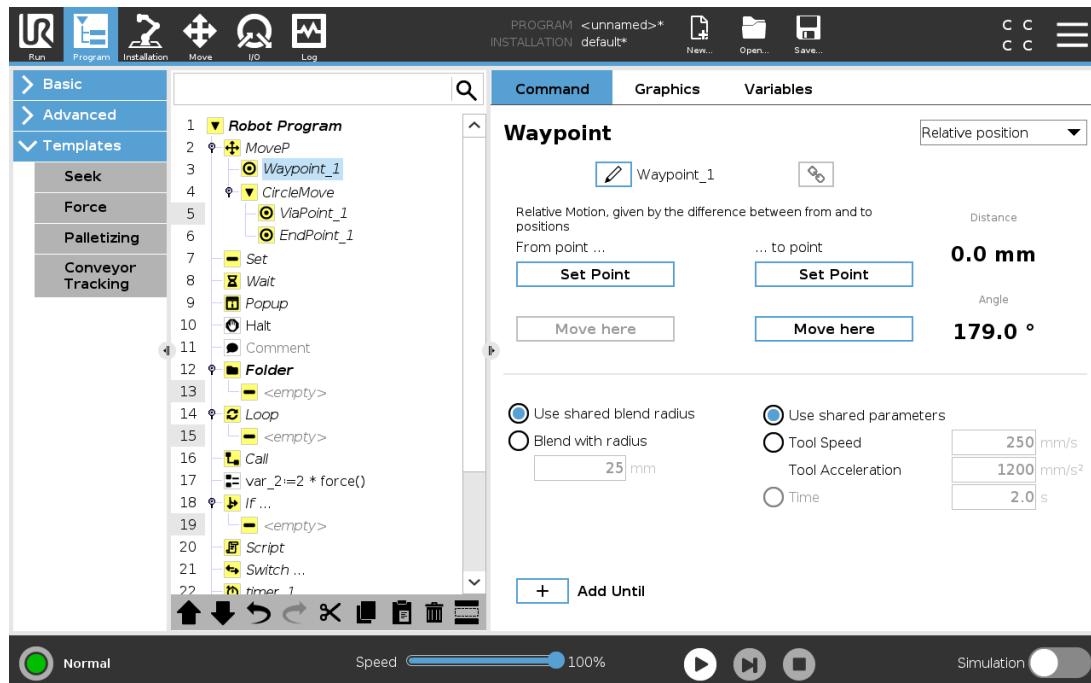


**12.6:** *Joint space (MoveJ) vs. cartesian space (MoveL) movement and blend.*

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. The blend produces a smoother and faster trajectory than the movements without a blend (see Figure 15.6). If velocity and acceleration are used for specifying the velocity profile, the blend stays within the blend radius during the blend. If using *time* instead of *velocity* and *acceleration* for specifying the velocity profile of both motions, the blend trajectory follows the trajectory of the original MoveJ. When both motions are time constrained, using blends does not save time.
- **Blends in MoveL** When blending in MoveL, the position of the blend follows a circle arc at constant speed. The orientation blends with a smooth interpolation between the two trajectories. The robot may decelerate on the trajectory before following the circular arc to avoid very high accelerations (e.g., if the angle between the two trajectories are close to 180 degrees).

### 23.10.3. Relative Waypoint

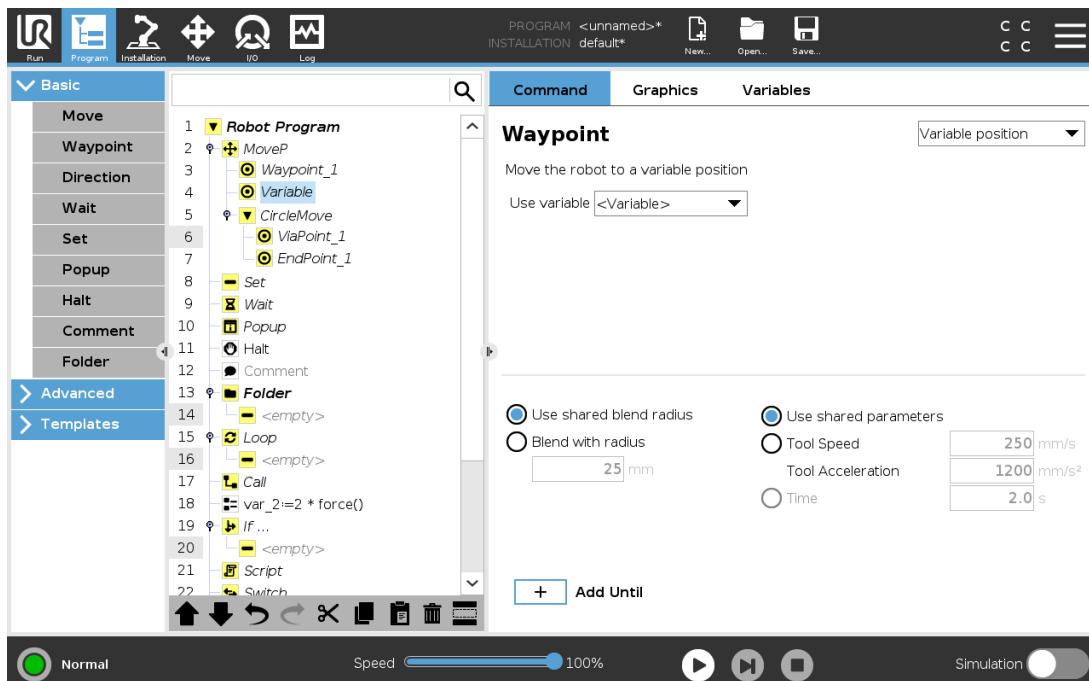


A waypoint with the position given relative to the robot arm's previous position, such as "two centimeters to the left". The relative position is defined as the difference between the two given positions (left to right).

Note: repeated relative positions can move the robot arm out of its workspace.

The distance here is the Cartesian distance between the TCP in the two positions. The angle states how much the TCP orientation changes between the two positions. More precisely, the length of the rotation vector describing the change in orientation.

## 23.10.4. Variable Waypoint



A waypoint with the position given by a variable, in this case `calculated_pos`. The variable has to be a *pose* such as

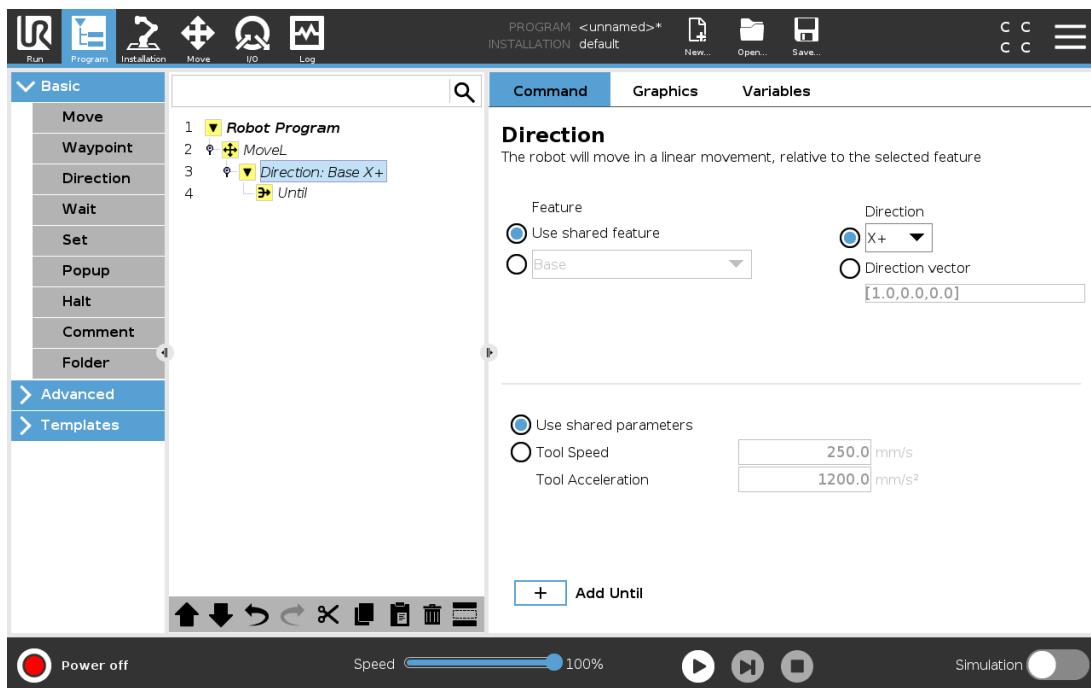
`var=p[0.5,0.0,0.0,0.3,14,0.0,0.0]`. The first three are *x,y,z* and the last three are the orientation given as a *rotation vector* given by the vector *rx,ry,rz*. The length of the axis is the angle to be rotated in radians, and the vector itself gives the axis about which to rotate. The position is 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 [Blend parameters on page 151](#)). If, when running the program, the blend radius overlaps a point, the robot will ignore it and move to the next one.

For example, to move the robot 20 *mm* along the *z*-axis of the tool:

```
var_1=p[0,0,0.02,0,0,0]
Move1
Waypoint_1 (variable position):
  Use variable=var_1, Feature=Tool
```

## 23.10.5. Direction

The program node **Direction** specifies a motion relative to feature axes or TCPs. The robot moves in along the path specified by the Direction Program Node until that movement is stopped by an **Until** condition.



### Adding a Direction Movement

1. Under Basic, tap **Direction** to add a linear movement to your Program Tree.
2. In the Direction field, under Feature, define the linear movement.

### Stopping a Direction Movement

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

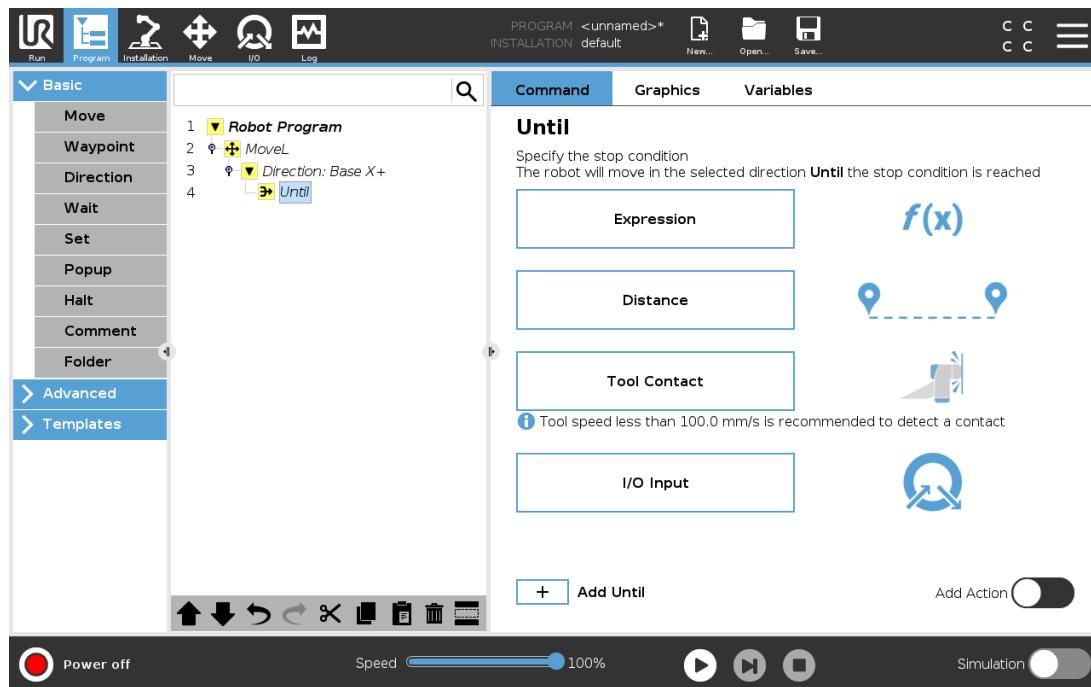
You can add Direction Vector settings, for **Tool Speed** and **Tool Acceleration**, to define the vector direction for linear motion, allowing for advanced uses as:

- defining linear motion relative to multiple feature axes
- computing the direction as a mathematical expression

The Direction Vectors defines a custom code expression that is resolved to a unit vector. For example, Direction vectors of [100,0,0] and [1,0,0] have the exact 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.

## 23.10.6. Until

The program node **Until** defines a stop criterion for a motion. The robot moves along a path and stops when contact is detected. In the Program Tree, you can add Until Nodes under Direction Nodes and Waypoint Nodes. You can add several stop criteria to a single movement. The motion stops when the first **Until** condition is met.



In the **Until** field, you can define the following stop criteria:

- **Distance** This node can be used to stop a Direction move when the robot has moved a certain distance. The velocity is ramped down so the robot stops exactly at the distance.
- **Tool Contact** (see [23.10.7. Until-Tool Contact below](#)) You can use this node to stop a movement when the robot tool detects a contact.
- **Expression** This node can be used to stop the motion due to a custom program expression. You can use I/Os, variables or script functions to specify the stop condition.
- **I/O Input** You can use this node to stop a signal controlled motion on an I/O Input.

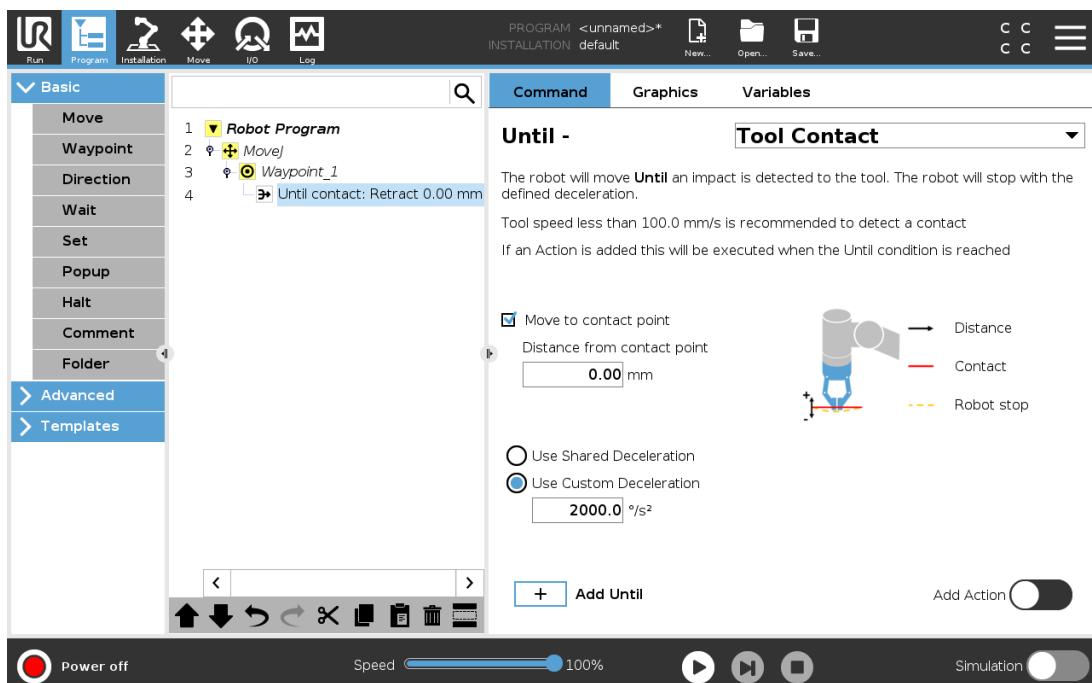
## 23.10.7. Until-Tool Contact

The program node **Until Tool Contact** allows the robot to stop its motion when contact with the tool is established. You can define the deceleration of the stop and the retraction of the tool.



### CAUTION

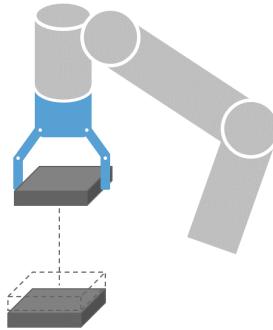
The default speed of motion is too high for contact detection. A faster speed of motion triggers a Protective Stop, before the Tool Contact condition can take effect. To avoid triggering a Protective Stop, lower the speed of motion. For example: 100m/s.



### NOTE

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

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



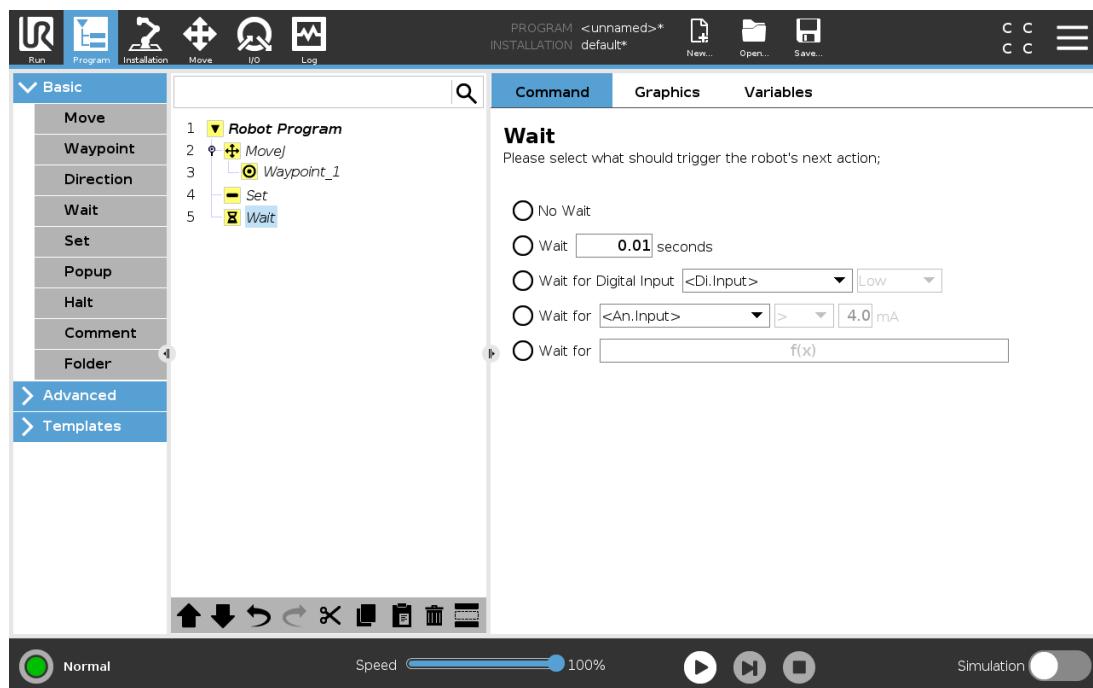
## Retract to Contact

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.

## Action

Add an **Action** allows a program node to be added if a specific **Until** condition is met. For 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.

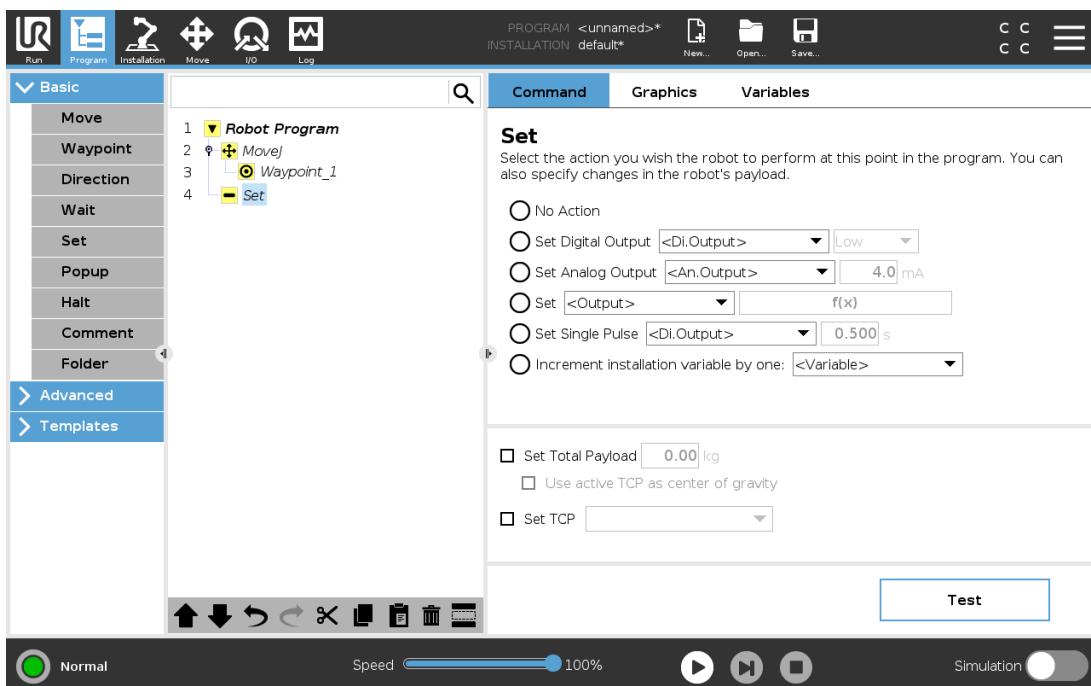
### 23.10.8. Wait



**Wait** pauses I/O signal, or expression, for a given amount of time. If **No Wait** is selected, nothing is done.

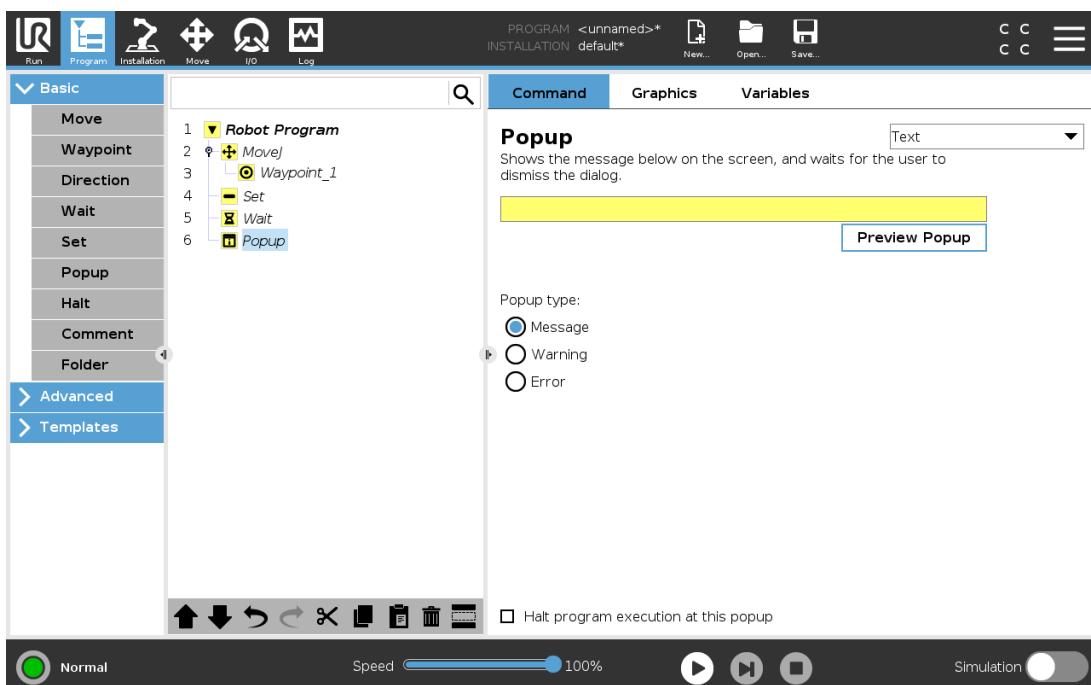
When the Tool Communication Interface TCI is enabled, the tool analog input is unavailable for **Wait For** selection and expressions.

## 23.10.9. Set



If the active TCP for a particular motion is known at the time of writing of the program, you can use the TCP selection by tapping **Move** in the Side Menu on the left, (see [23.10.1. Move](#) on page 146). For further information about configuring named TCPs (see [24.2. TCP Configuration](#) on page 197).

## 23.10.10. Popup



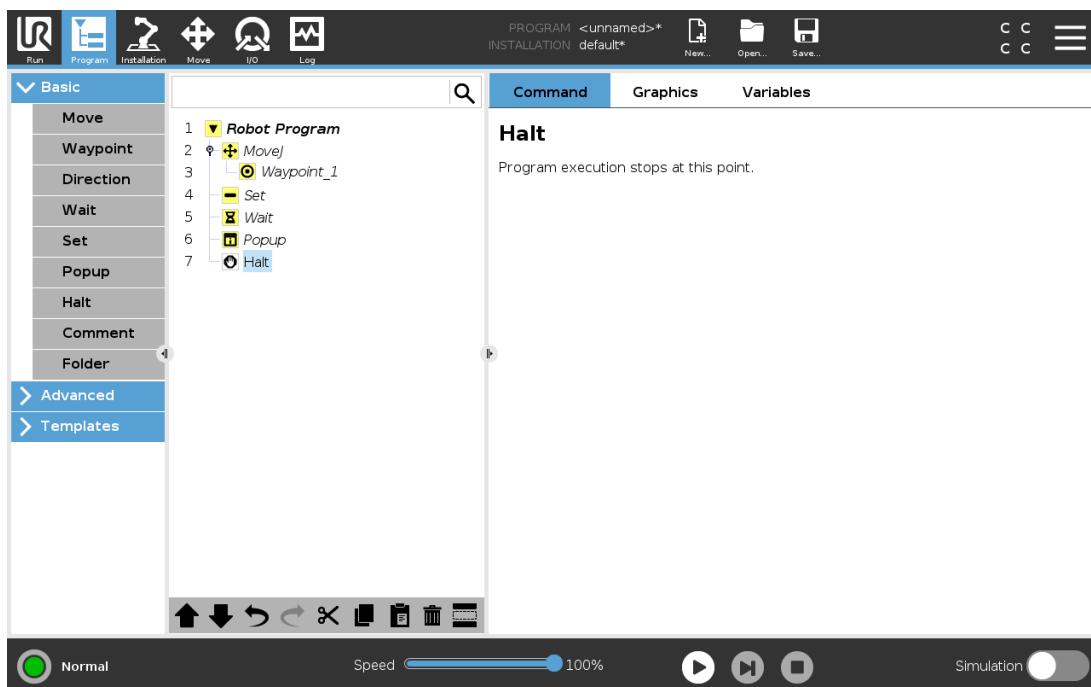
The Popup is a message that appears on the screen when the program reaches the Popup node in the program tree. Under the Command tab, tap the empty field and use the on-screen keyboard to create text content for the popup message. Messages are limited to a maximum of 255 characters.

You can select the **Text** dropdown, if you prefer to have a variable displayed in your popup message instead of text.

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

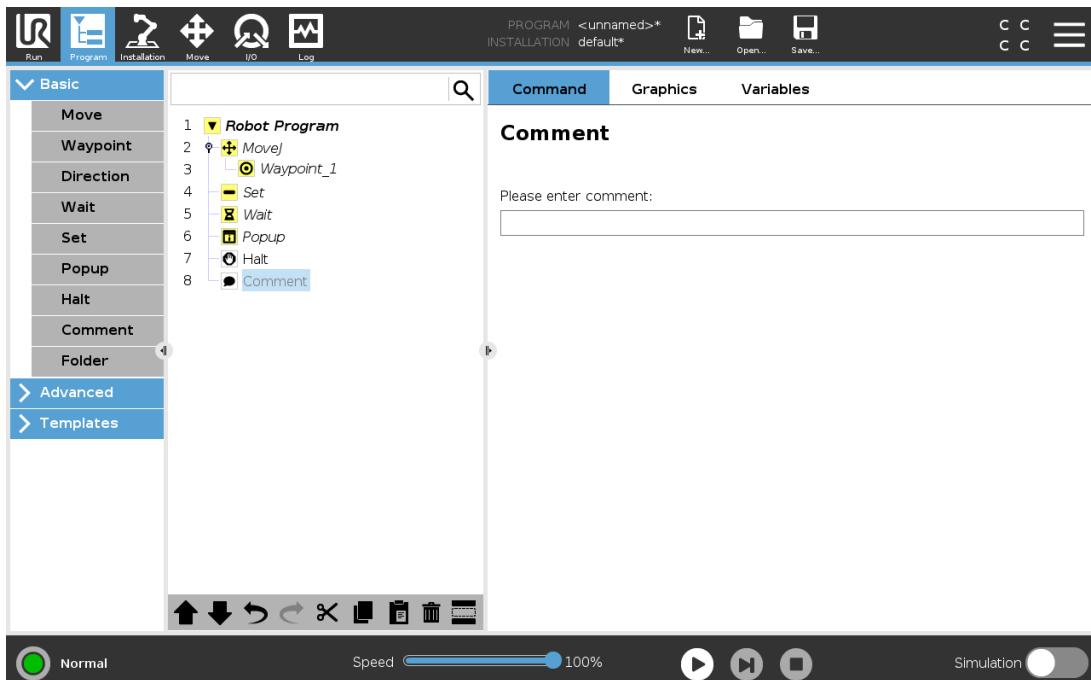
During program execution, when the popup message appears, tap **OK** in the popup dialog box to continue the program.

### 23.10.11. Halt



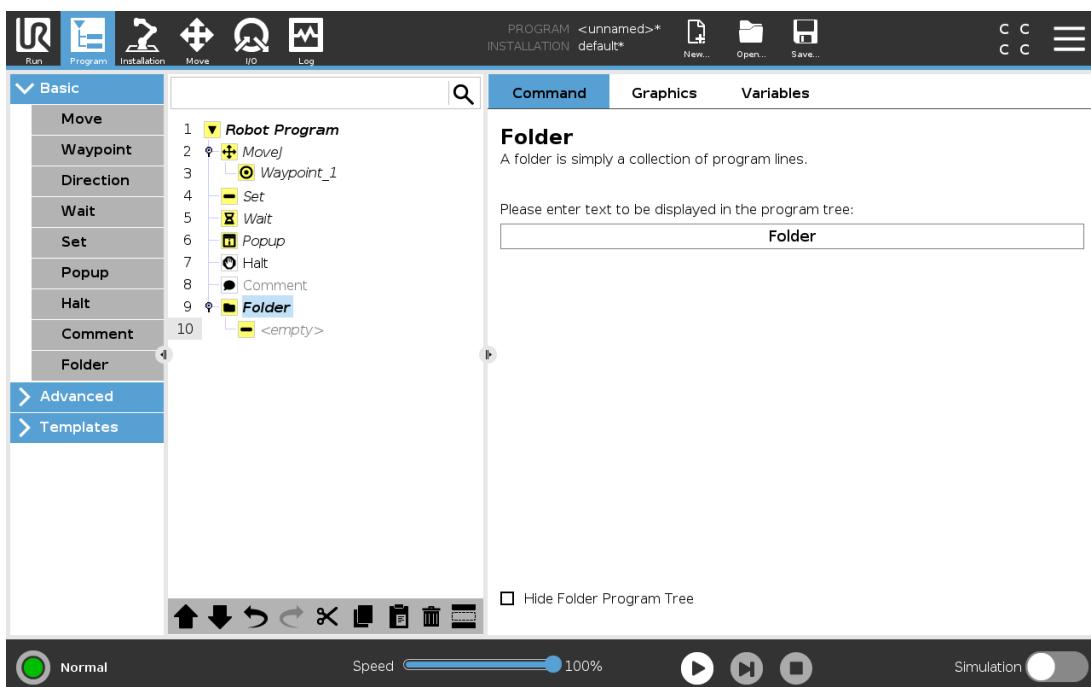
The program execution stops at this point.

## 23.10.12. Comment



Gives the programmer an option to add a line of text to the program. This line of text does not do anything during program execution.

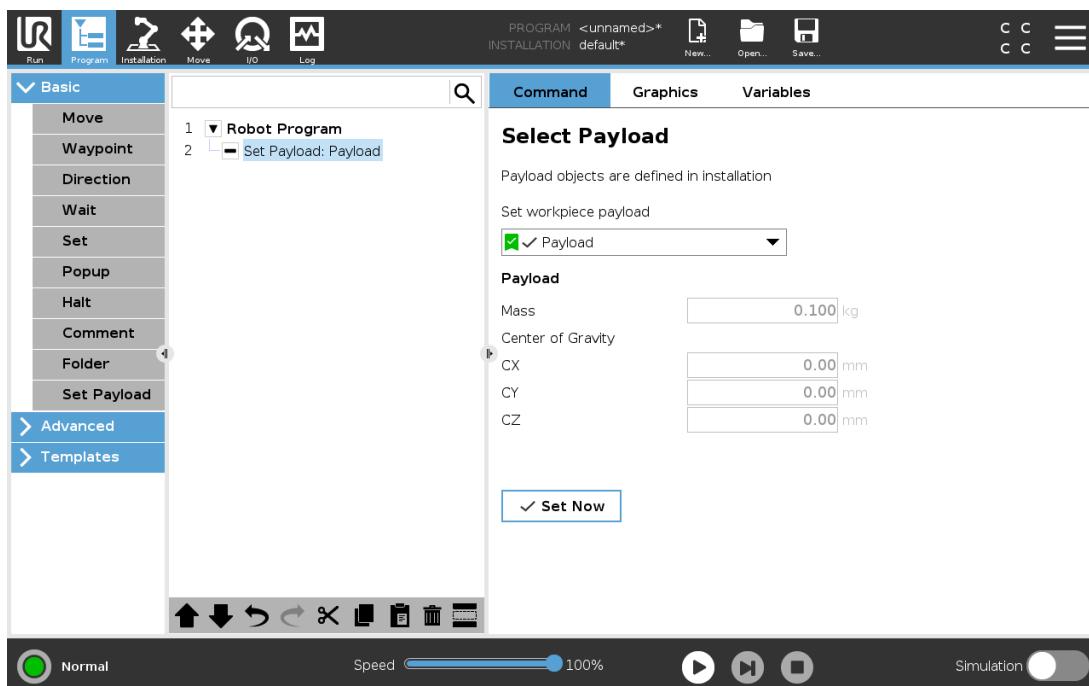
## 23.10.13. Folder



A **Folder** is used to organize and label specific parts of a program, to clean up the program tree, and to make the program easier to read and navigate.

**Folders** have no impact on the program and its execution.

## 23.10.14. Set Payload



Use the **Set Payload** command to set the payload of the robot arm. You can adjust the payload weight to prevent the robot from triggering a protective stop, when the weight at the tool differs from the expected payload.



### NOTICE

Setting the payload correctly ensures optimal motion performance and avoids protective stops.

- Set the payload when picking up- and releasing objects with a gripper.

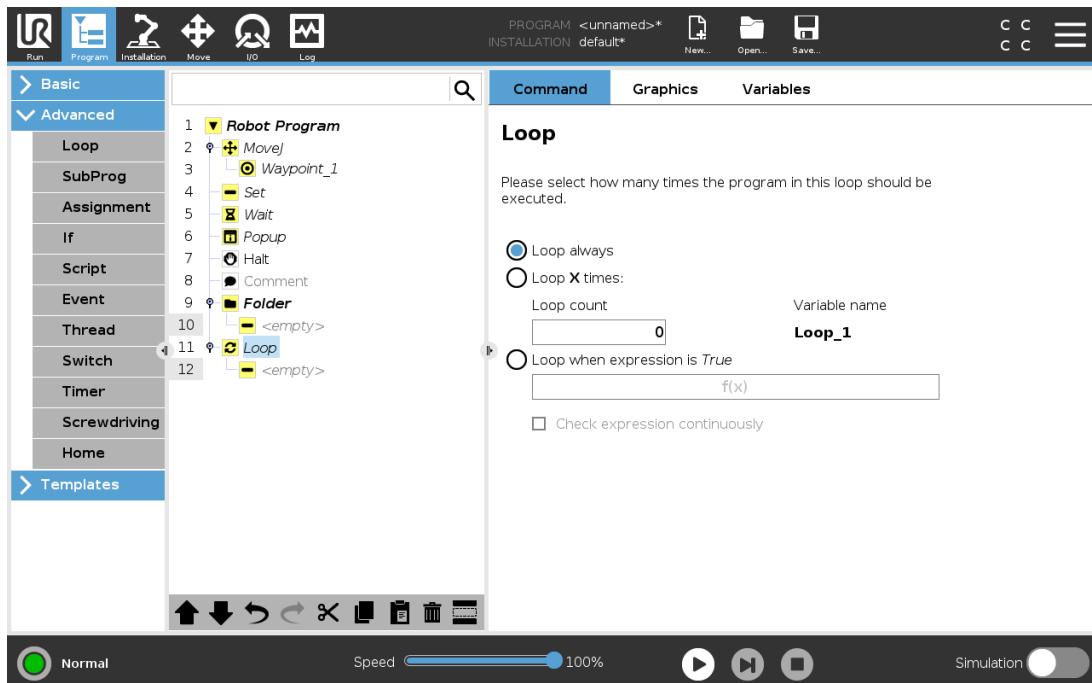
To use the Set Payload command you can:

- Use the drop-down, under **Select Payload**, to select one of the payloads already configured in the installation.
- Or, use the drop-down to configure a new payload by selecting **Custom Payload** and completing the mass and CoG fields.

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

## 23.11. Advanced program nodes

### 23.11.1. Loop

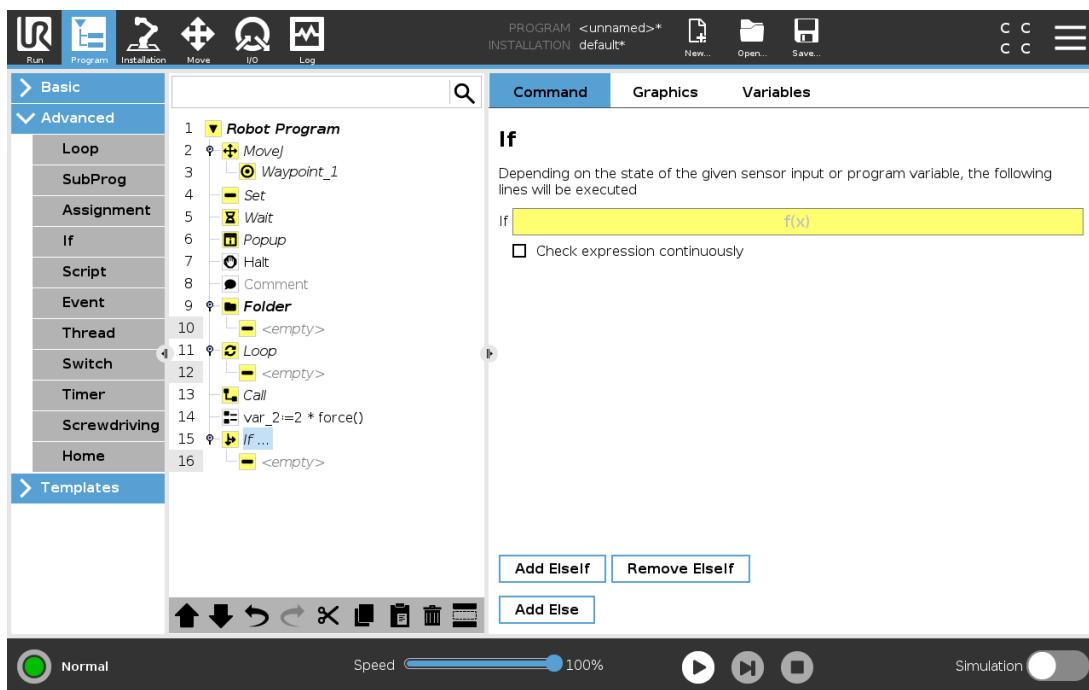


Loops the underlying program commands. Depending on the selection, the underlying program commands are either looped infinitely, a certain number of times or as long as the given condition is true. When looping a certain number of times, a dedicated loop variable (called `loop_1` in the screenshot above)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

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.

### 23.11.2. If

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



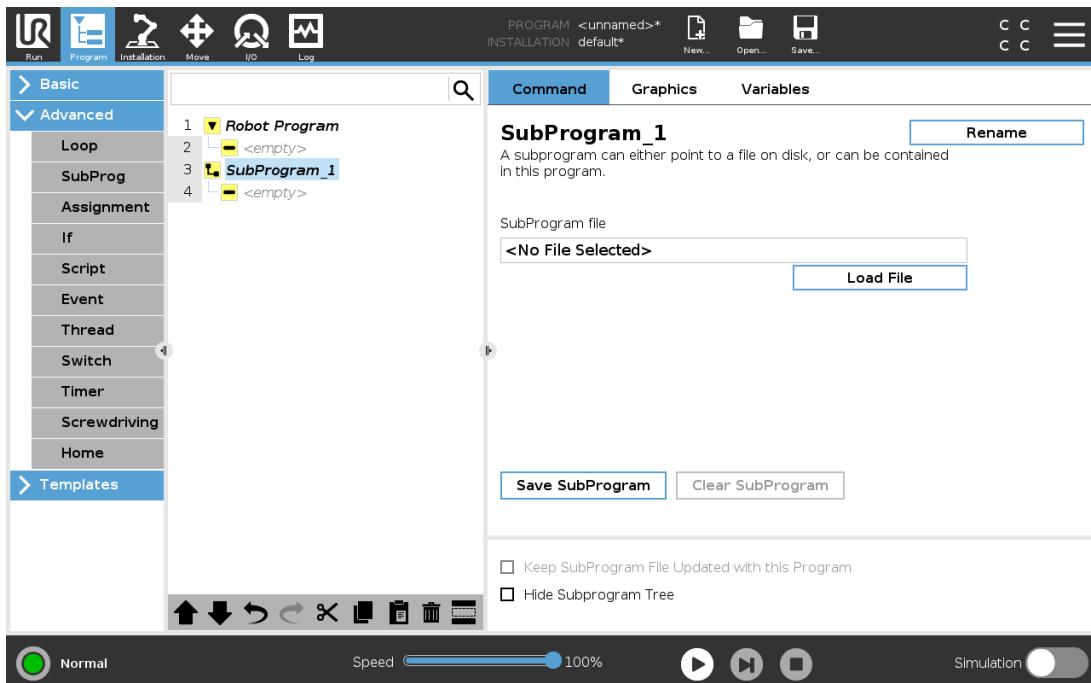
Select conditions in the Expression Editor that make up expressions using an **If** statement. If a condition is evaluated as True, the statements within this **If** command are executed. An **If** statement can have only one **Else** statement. Use **Add Elseif** and **Remove Elseif** to add and remove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



### NOTE

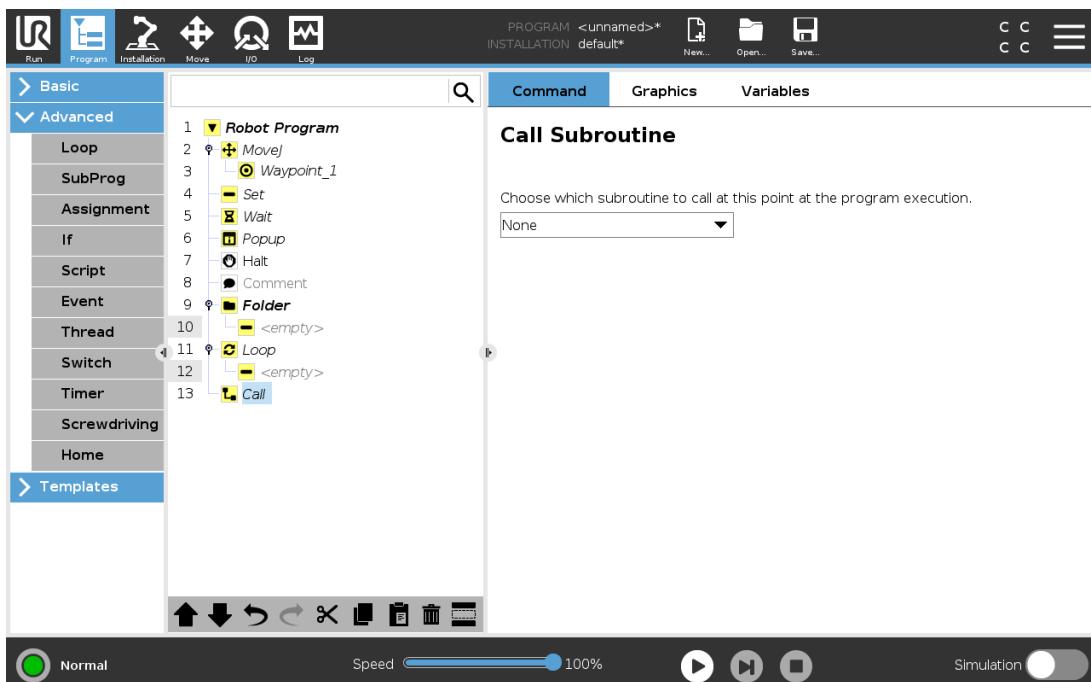
If there are waypoints inside an **If** expression or inside a **Loop** expression with the **Check Expression Continuously** option, you can add a **stopj()** or a **stopl()** after the expression to gently decelerate the robot arm. This is valid for both **If** and **Loop** Commands (see section [23.11.1. Loop](#) on the previous page).

### 23.11.3. SubProgram



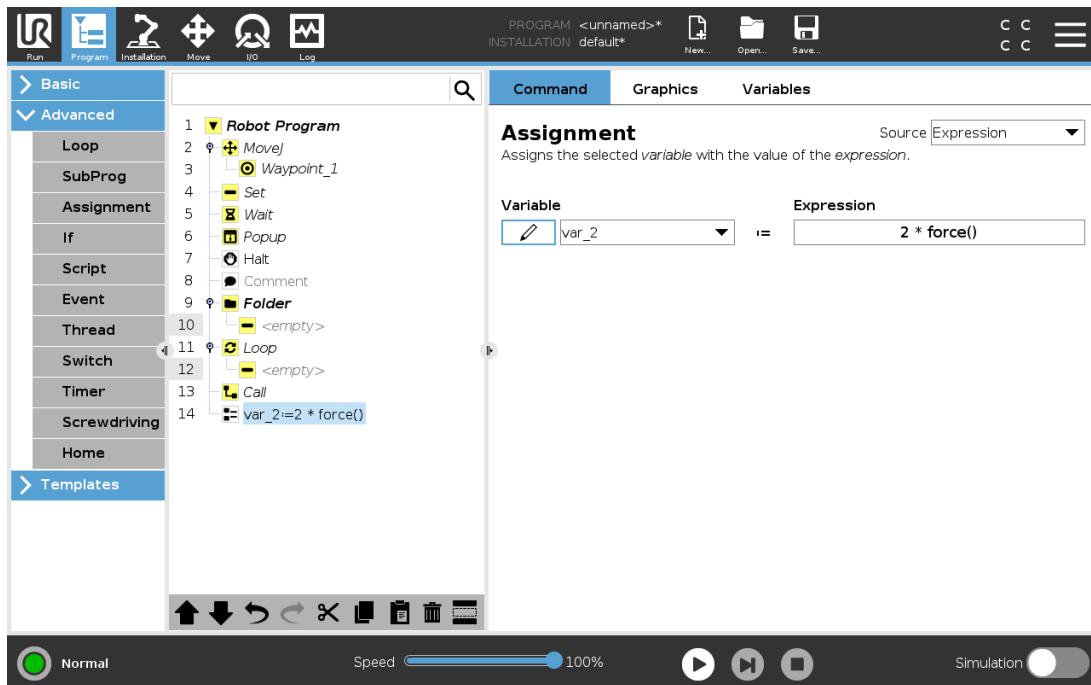
A SubProgram can hold program parts that are needed several places. A SubProgram can be a separate file on the disk, and can also be hidden to protect against accidental changes to the SubProgram.

### Call Subroutine



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

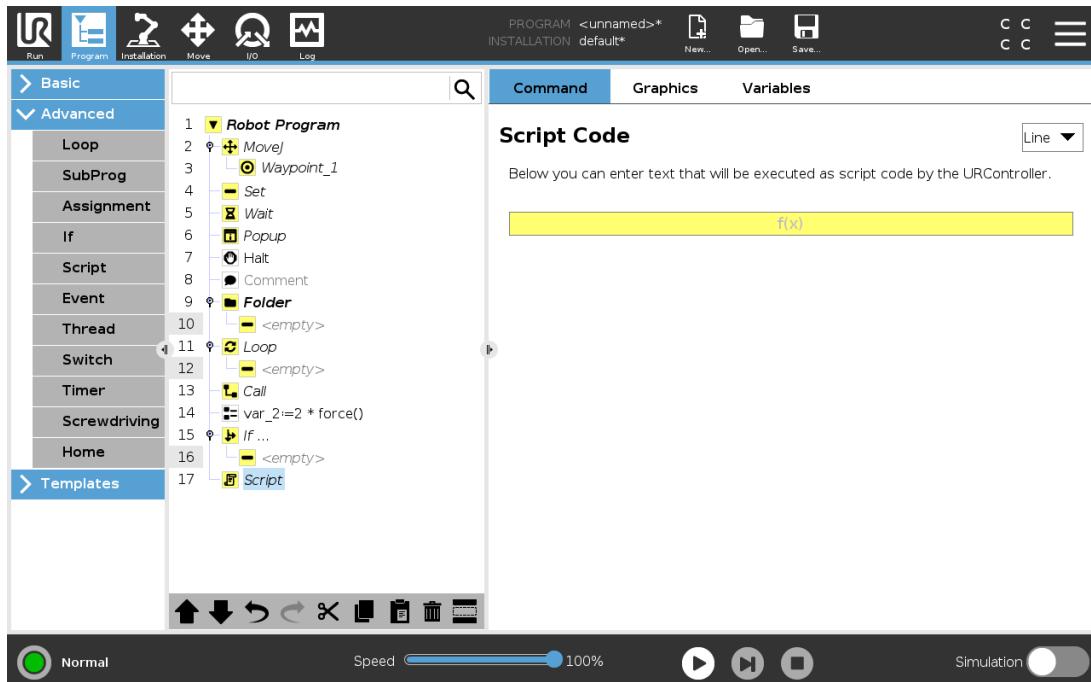
## 23.11.4. Assignment



Assigns values to the variables. The variable value can be the result of expressions created in the Expression Editor (see section [23.3. Expression Editor on page 139](#)).

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.

## 23.11.5. Script



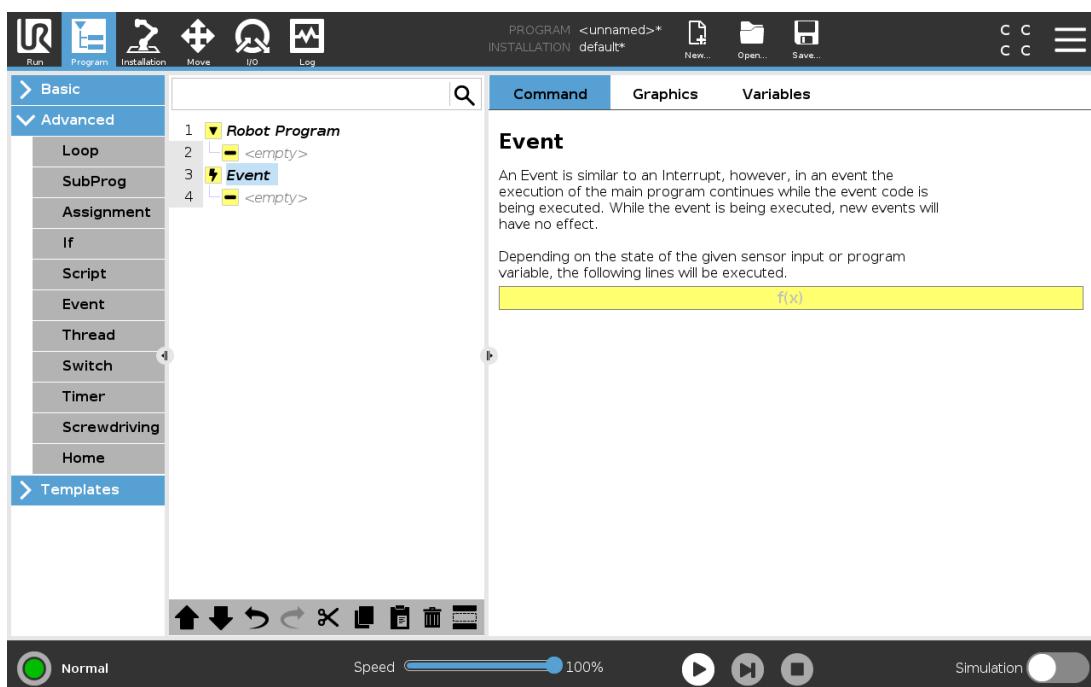
The following options are available in the drop down list under Command:

- **Line** allows you to write a single line of URscript code, using the Expression Editor ( [23.3. Expression Editor on page 139](#) )
- **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>).

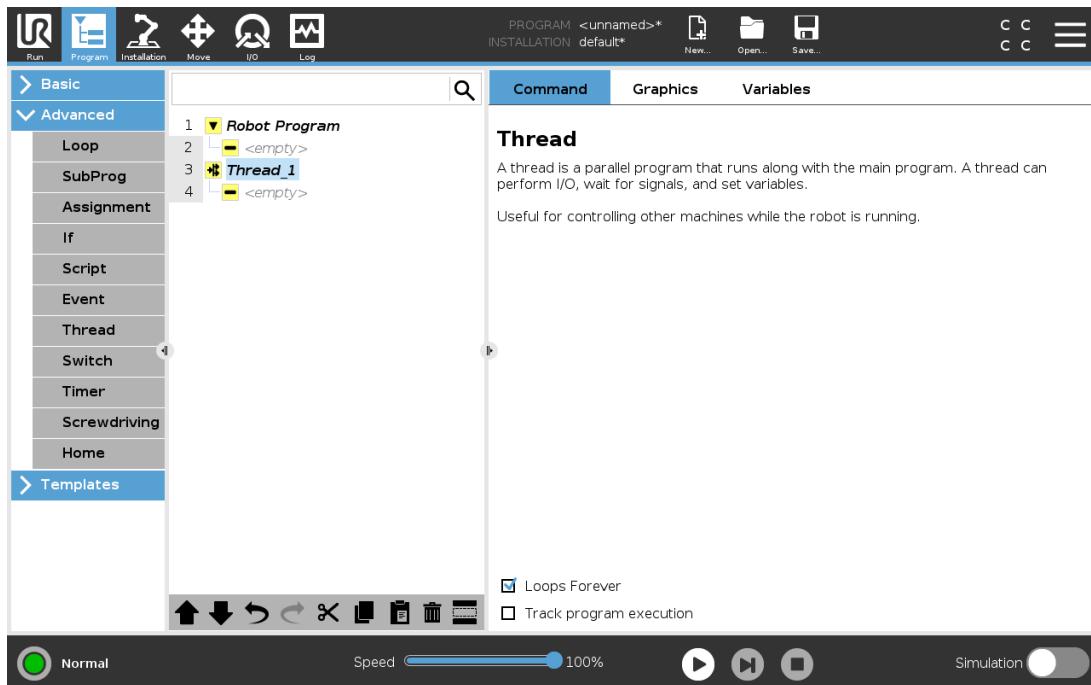
Functions and variables declared in a URscript file are available for use throughout the program in the PolyScope.

## 23.11.6. Event



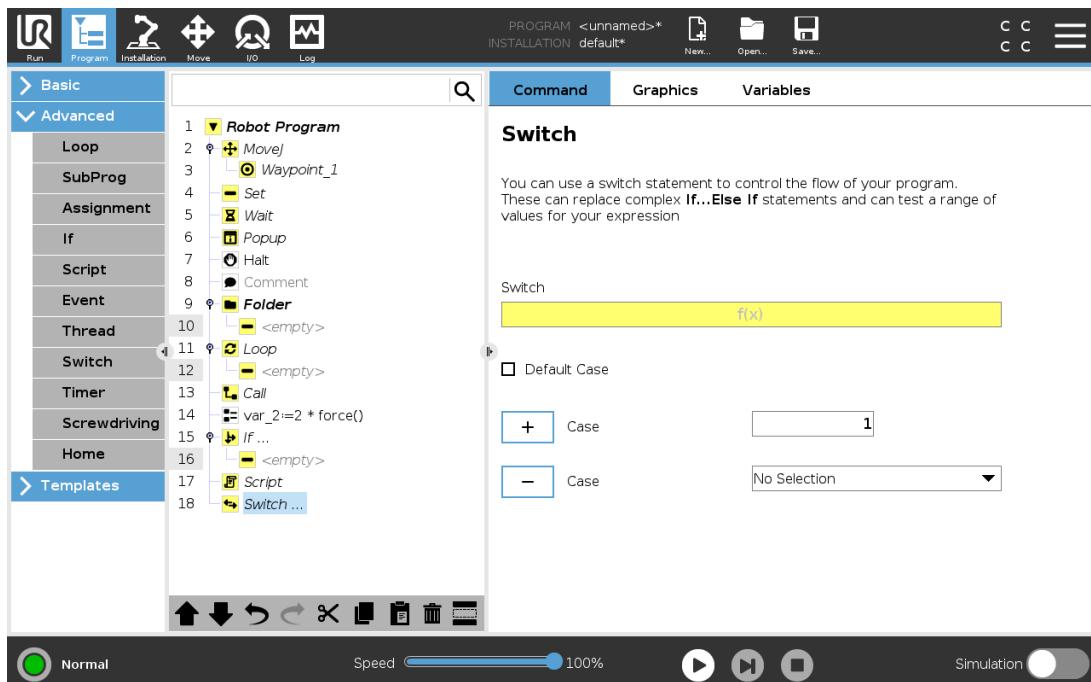
An event can be used to monitor an input signal, and perform some action or set a variable when that input signal goes high. For example, in the event that an output signal goes high, the event program can wait for 200ms and then set it back to low again. This can make the main program code a lot simpler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

## 23.11.7. Thread



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.

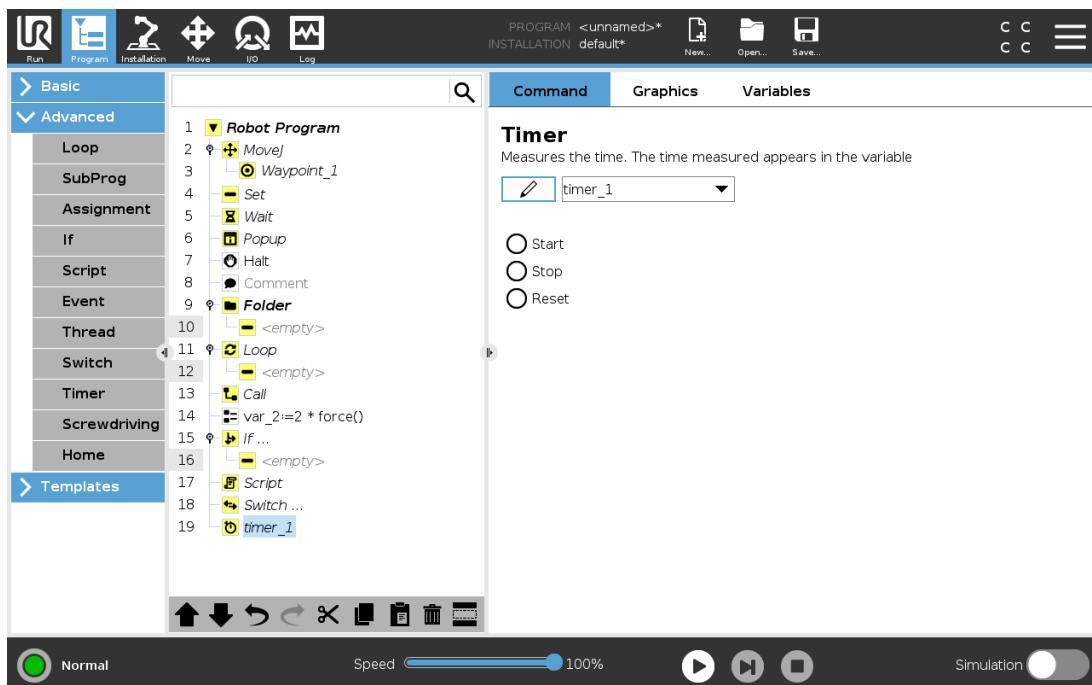
## 23.11.8. Switch



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.

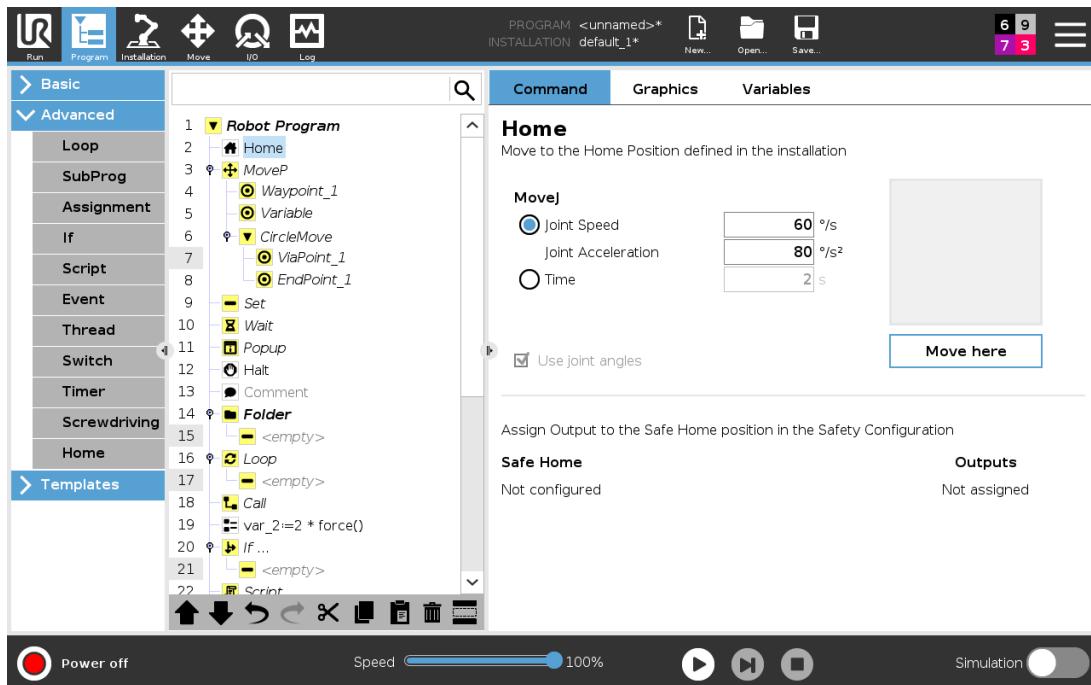
Each **Switch** can have several **Cases** and one **Default Case**. **Switches** can only have one instance of any **Case** values defined. **Cases** can be added using the buttons on the screen. A **Case** command can be removed from the screen for that switch.

### 23.11.9. Timer



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.

## 23.11.10. Home



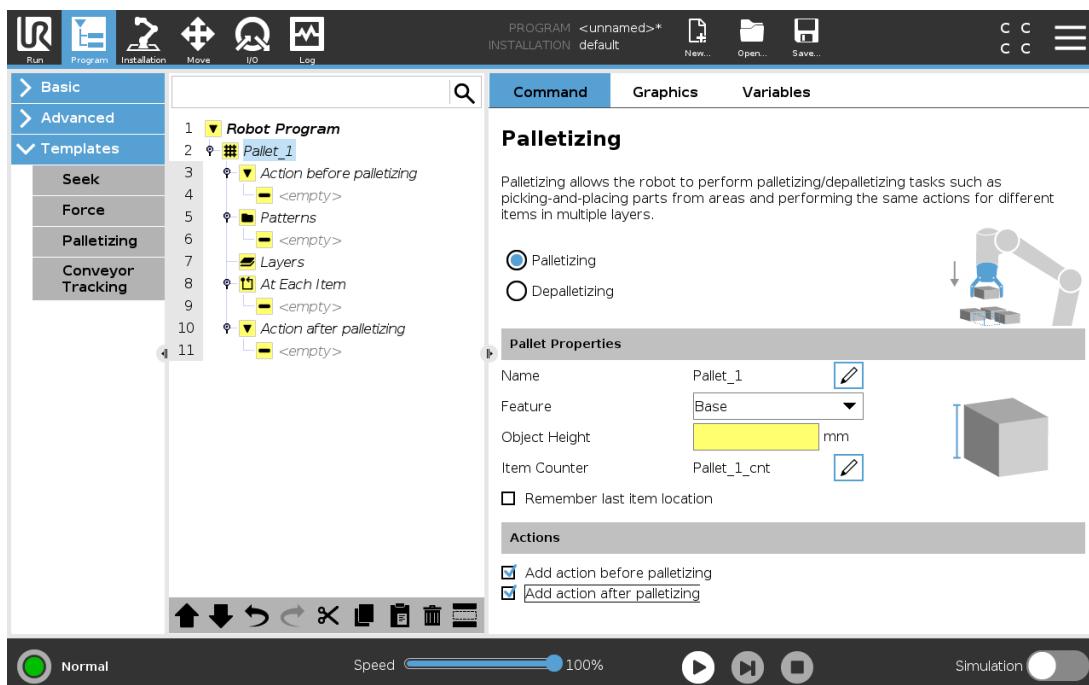
The Home node uses joint angles to move the robot to a predefined Home position. If defined as a Safe Home position, the Home node displays as Home(Safety) in the Program Tree. If the Home position is out of sync with Safety, the node is undefined.

## 23.12. Templates

### 23.12.1. Palletizing

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 [Adding a Separator Between Layers in a Palletizing Sequence on page 175](#)). Furthermore, you can use Features from Pallet Properties to easily adjust the placement of your pallet. To learn about Features, see [24.17. Features on page 215](#). 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 [24.17. Features on page 215](#)) 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 [Adding a Separator Between Layers in a Palletizing Sequence on page 175](#)).
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 [23.10.1. Move on page 146](#) 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.

	<p><b>Line</b> To teach the positions, select each item in the Program Tree:</p> <ul style="list-style-type: none"> <li>StartItem1</li> <li>EndItem1</li> </ul> <p>Insert the number of items in your sequence using the <b>Items</b> text box at the bottom of the screen.</p>
	<p><b>Grid</b> To teach the positions, select each item in the Program Tree:</p> <ul style="list-style-type: none"> <li>CornerItem1</li> <li>CornerItem2</li> <li>CornerItem3</li> <li>CornerItem4</li> </ul> <p>Insert the number of rows and columns in the appropriate text boxes to set the dimensions of the pattern.</p>
	<p><b>Irregular</b> To teach the positions, select each item in the Program Tree:</p> <ul style="list-style-type: none"> <li>Item1</li> <li>Item2</li> <li>Item3</li> </ul> <p>Tap <b>Add Item</b> to add and identify a new item in the sequence.</p>

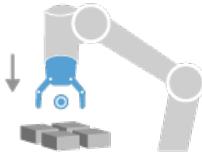
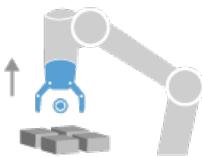
8. In the Program Tree, tap the **Layers** node to configure the layers of your palletizing sequence. Use the **Choose Pattern** drop-down menu to select the pattern for each layer. Tap the **Add layer** button to add additional layers to your program. Layers must be added in the correct order, as they cannot be reordered later.
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.

### (A) 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 [23.10.1. Move on page 146](#)). Tap **Next**.
5. Repeat Step 3.

6. Tap **Set Waypoint** to teach the Exit Waypoint (see [23.10.1. Move on page 146](#)). Tap **Next**.
7. Tap **Finish**.
8. You can now add appropriate gripper action nodes in the Tool Action folder in the Program Tree.

	<b>ToolActionPoint Waypoint:</b> 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.
	<b>Approach Waypoint:</b> The collision-free position and direction you want the robot to take when approaching an item in a layer.
	<b>Tool Action:</b> The action you want the robot attachment to perform for each item.
	<b>Exit Waypoint:</b> The position and direction you want the robot to take when moving away from an item in a layer.

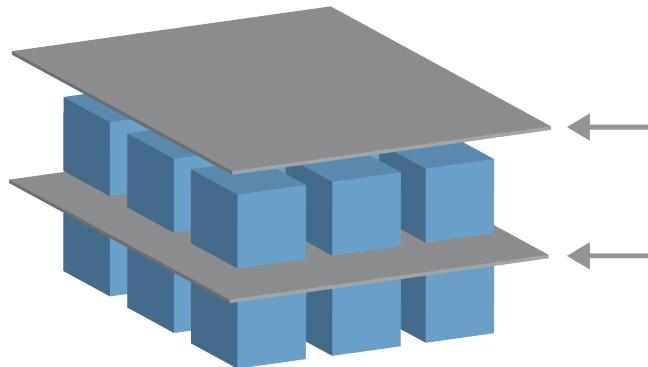
## (B) Manual Configuration

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

[23.10.1. Move on page 146](#)). 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 [23.10.1. Move on page 146](#)).
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

### (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 [23.10.1. Move on page 146](#)). Tap **Next**.
5. Repeat Step 3.
6. Tap **Set Waypoint** to teach the Exit Waypoint (see [23.10.1. Move on page 146](#)). 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 [23.10.1. Move on page 146](#)).
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 [23.10.1. Move on page 146](#)).
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

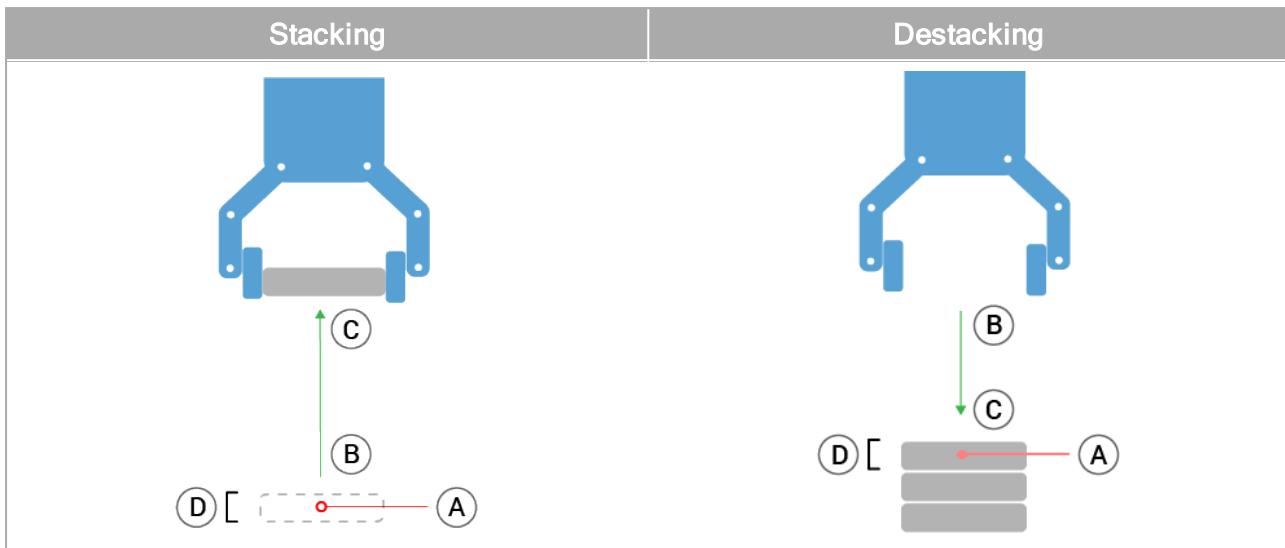
You can customize your palletizing program in the following ways:

- 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 [24.17. Features on page 215](#)) 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 [23.10.1. Move on page 146](#)).
- You can change the speeds and blends radii (see [23.10.1. Move on page 146](#)).
- You can add other program nodes to the At Each Item sequence or the Separator Action sequence.

### 23.12.2. Seek

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.



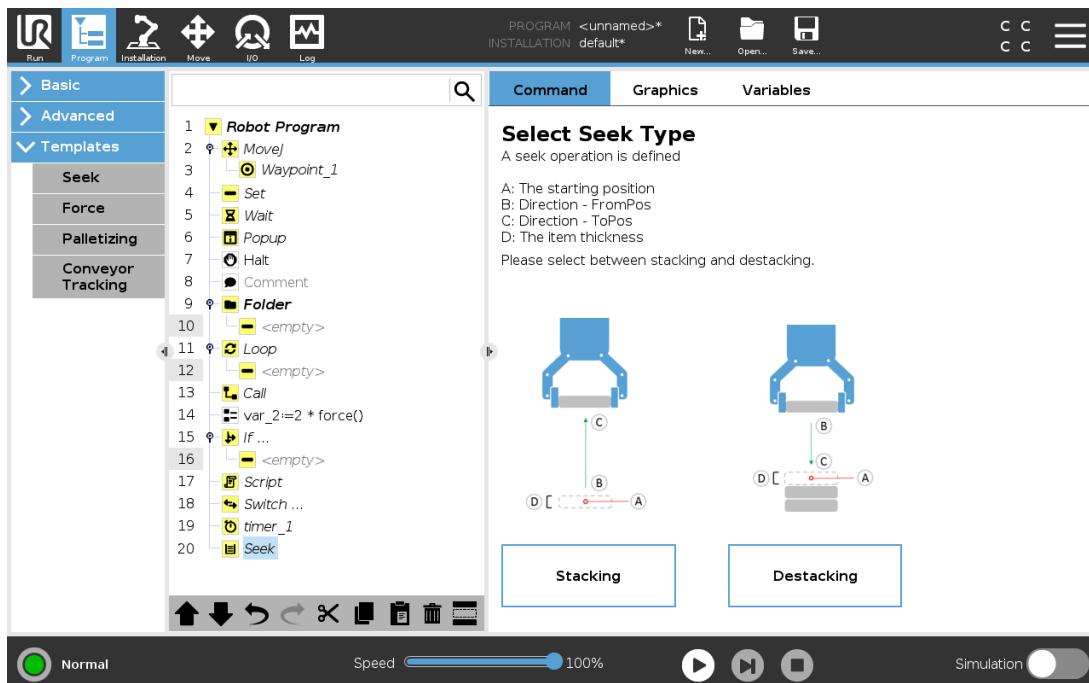
To program a seek operation, define the following:

- *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.

## Stacking

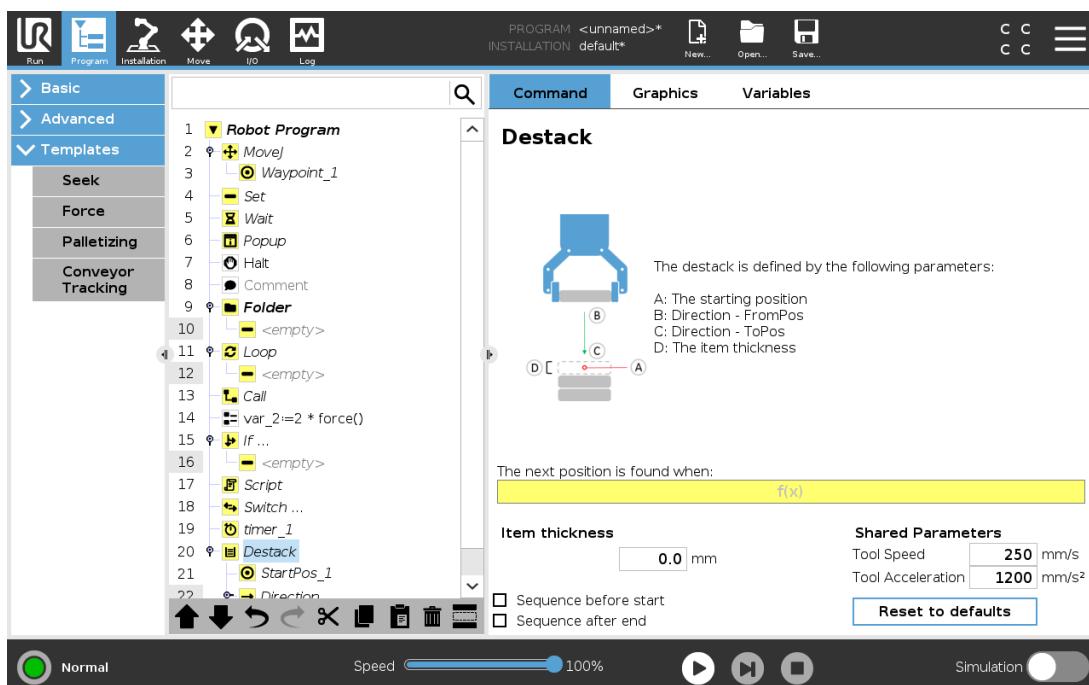


During Stacking, the robot arm moves to point A, then moves *opposite* the direction to search for the next stack position. When the next stack position is found, the robot remembers it and performs the special sequence.

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.

## Destacking



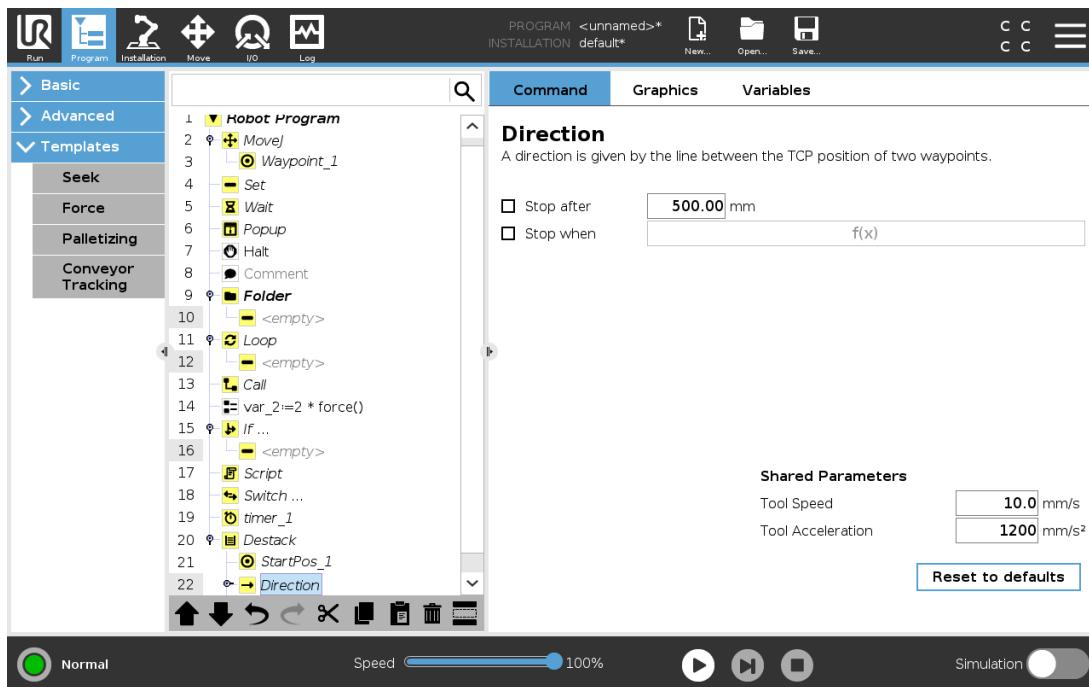
During Destacking, the robot arm moves from point A in the given direction to search for the next item. The condition on the screen determines when the next item is reached. When the condition is satisfied, the robot remembers the position and performs the special sequence.

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.

## 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.

## 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.

### 23.12.3. Force



#### NOTE

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.

**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.



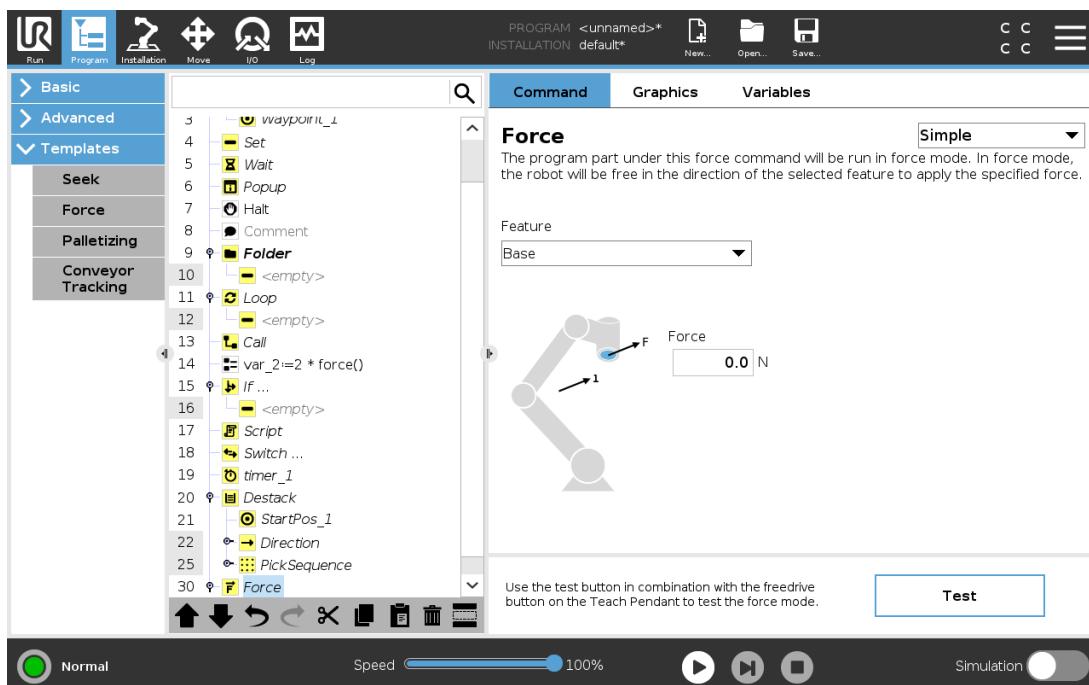
#### NOTE

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.



#### 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 (see [24.17. Features on page 215](#)).

## 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 [23.8. Graphics Tab on page 143](#)), when the robot is operating in force mode.

#### 23.12.4. 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].



##### NOTE

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.

#### 23.12.5. 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.

#### 23.12.6. 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.

## 23.12.7. Conveyor Tracking

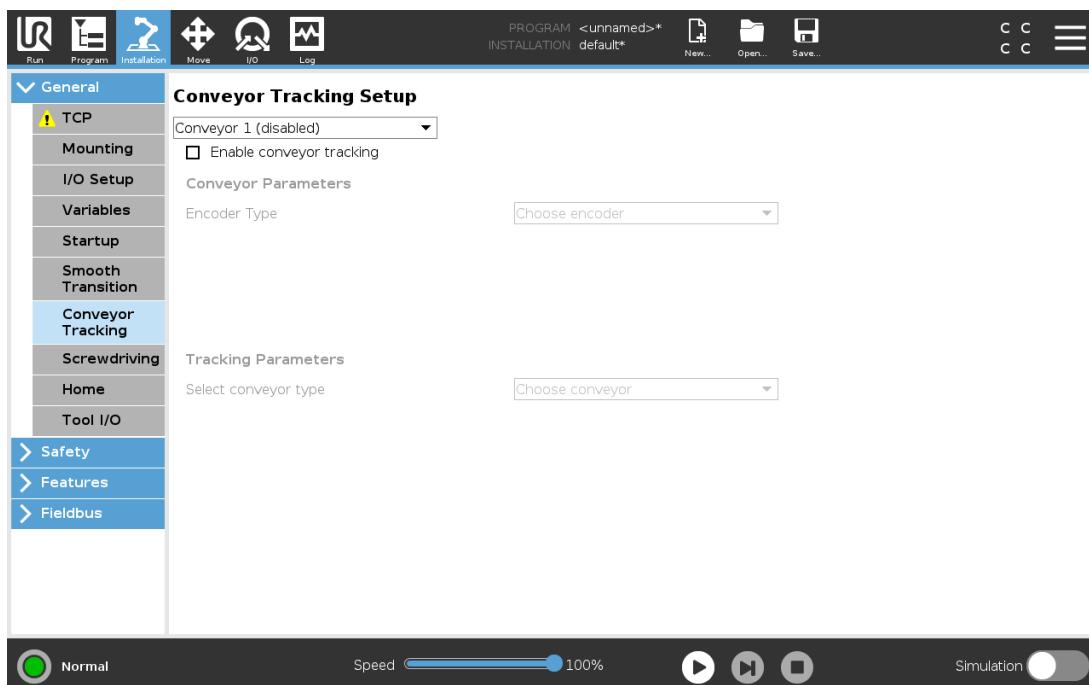


### NOTE

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.

Conveyor Tracking allows the Robot Arm to track the movement of up to two conveyors. Conveyor Tracking is defined in the Installation Tab (see section [24.14. Conveyor Tracking Setup on page 211](#)).



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

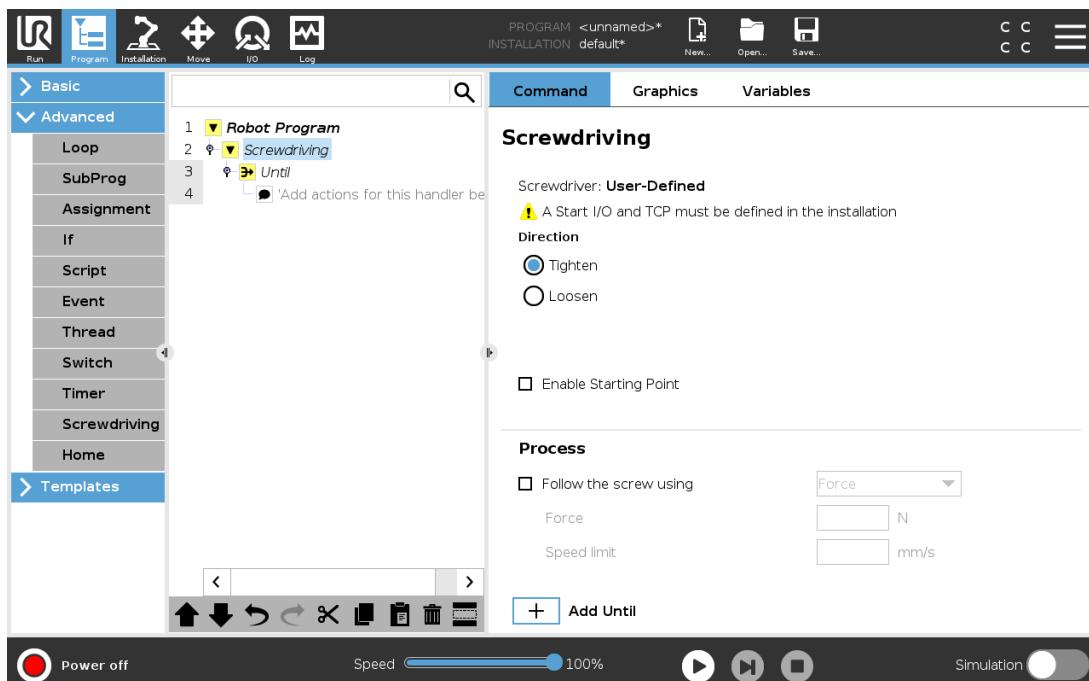
1. In the Header, tap **Program**.
2. Tap **Templates** and select **Conveyor Tracking** to add a Conveyor Tracking node to the Program Tree. Any movements listed under the Conveyor Tracking node tracks the movement of the conveyor.
3. Under Conveyor Tracking, in the Select Conveyor dropdown list, select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

**NOTE**

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.

## 23.12.8. Screwdriving

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 [24.1. General on page 197](#)).



### 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.

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.



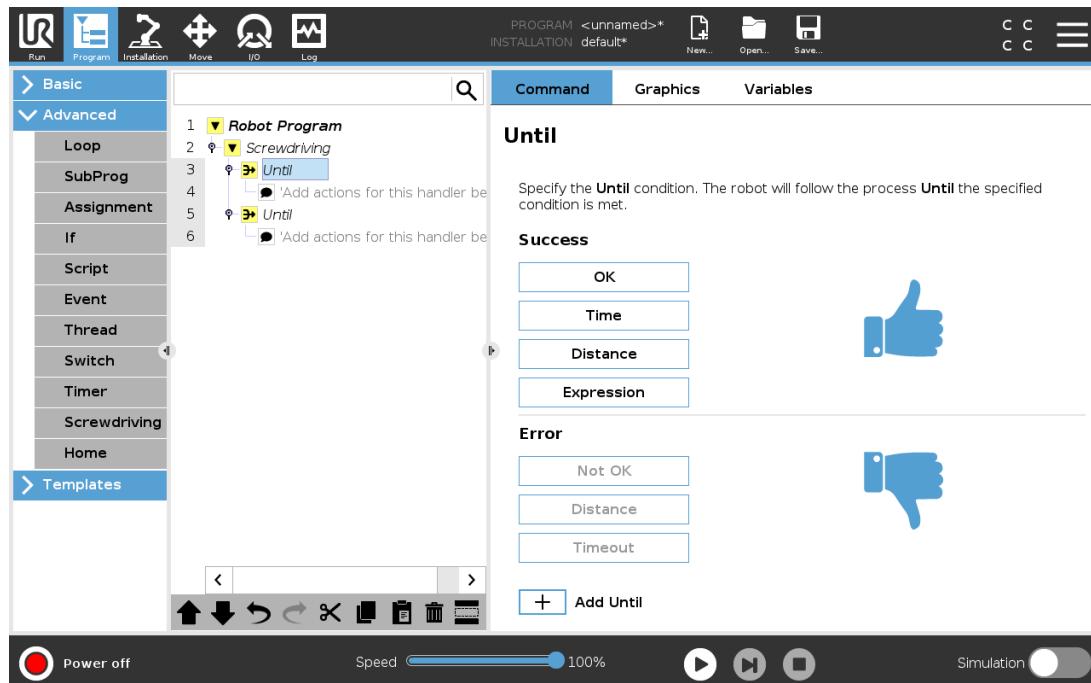
### CAUTION

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 [23.11.2. If on page 164](#)), select **Expression** to describe the condition under which the robot follows the screw.

## 23.12.9. Screwdrive Until

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.

**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**

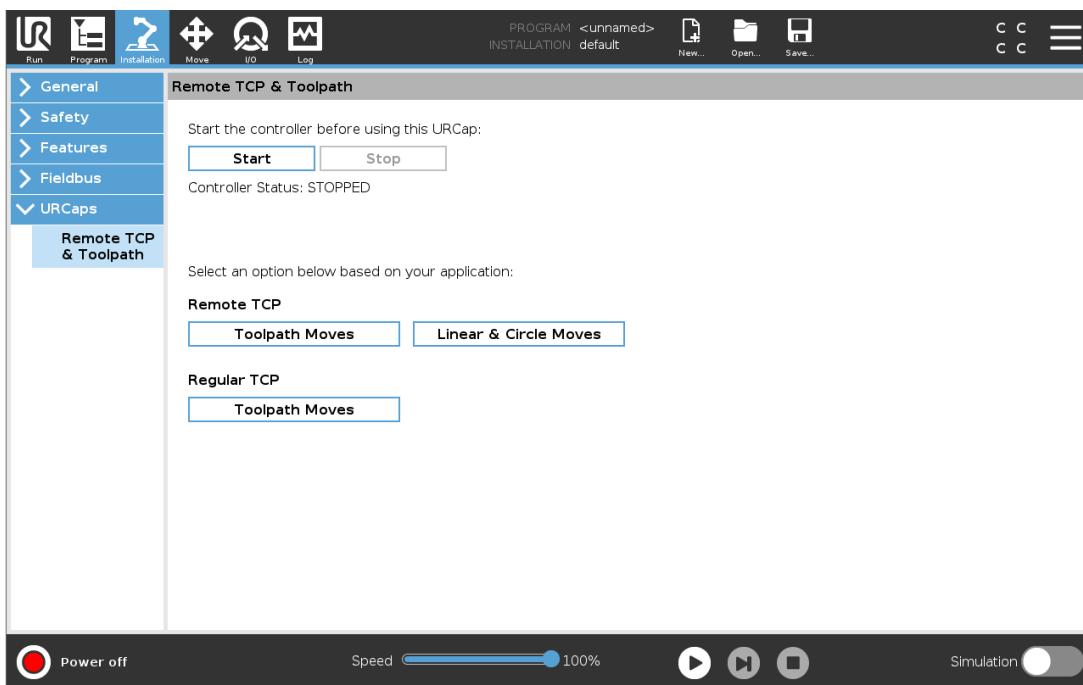
- **Not OK:** Screwdriving stops when a NOT OK signal from the screwdriver is detected.
- **Distance:** Screwdriving stops when the defined distance is exceeded.
- **Timeout:** Screwdriving stops when the defined time is exceeded.

## 23.13. URCaps

### 23.13.1. Remote TCP and Toolpath URCap

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 [17.7. Robot Registration and URCap License files on page 102](#)). 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.



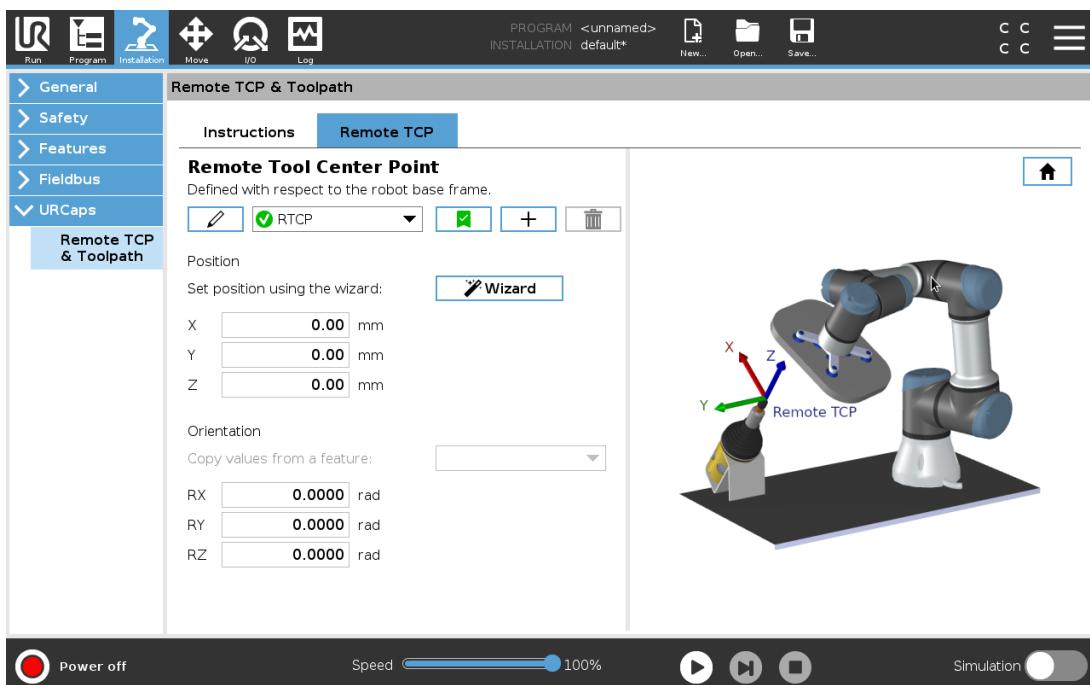
Similar to a regular TCP (see [24.2. TCP Configuration on page 197](#)) you can define and name an RTCP in the Installation Tab's Setup. You can also complete the following actions:

- Add, rename, modify, and remove RTCPs
- Understand the default and active RTCP
- Teach RTCP position
- Copy RTCP orientation

## Setting the RTCP 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.



## 23.13.2. 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 [24.2. TCP Configuration on page 197](#).

### 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 [24.2. TCP Configuration on page 197](#).

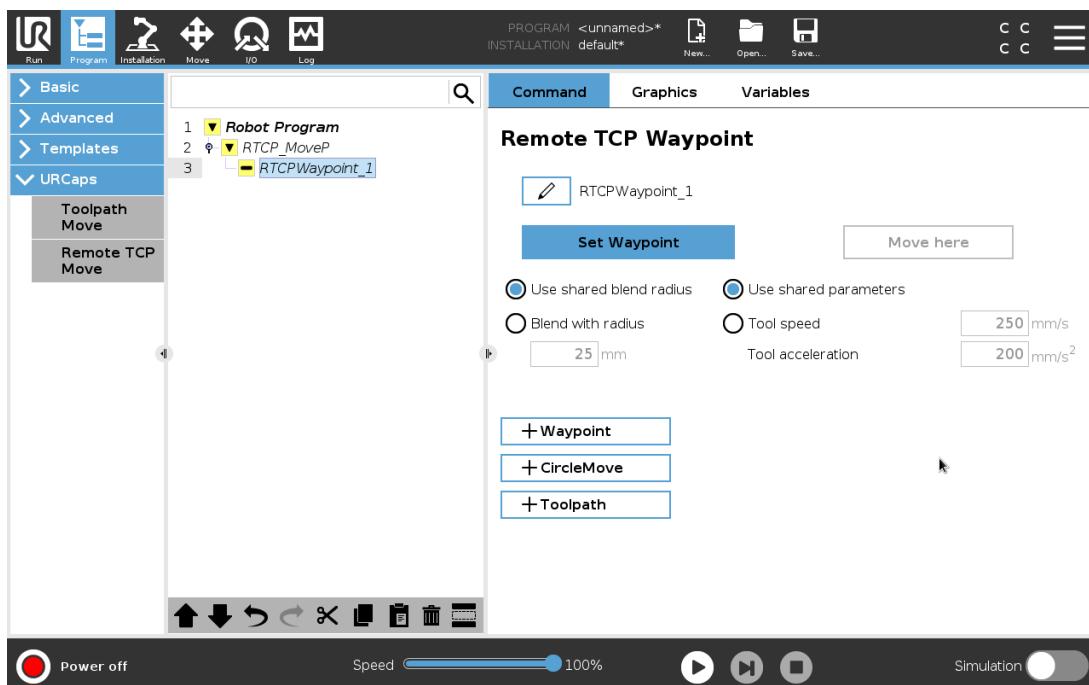


#### NOTE

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  $Ar$  due to centripetal acceleration.

## 23.13.3. 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 RTCP 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**.



### NOTE

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.

### 23.13.4. Remote TCP Toolpath

The Remote TCP and Toolpath URCap generates robots motions automatically, making it easier to follow complex trajectories accurately.

## 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:

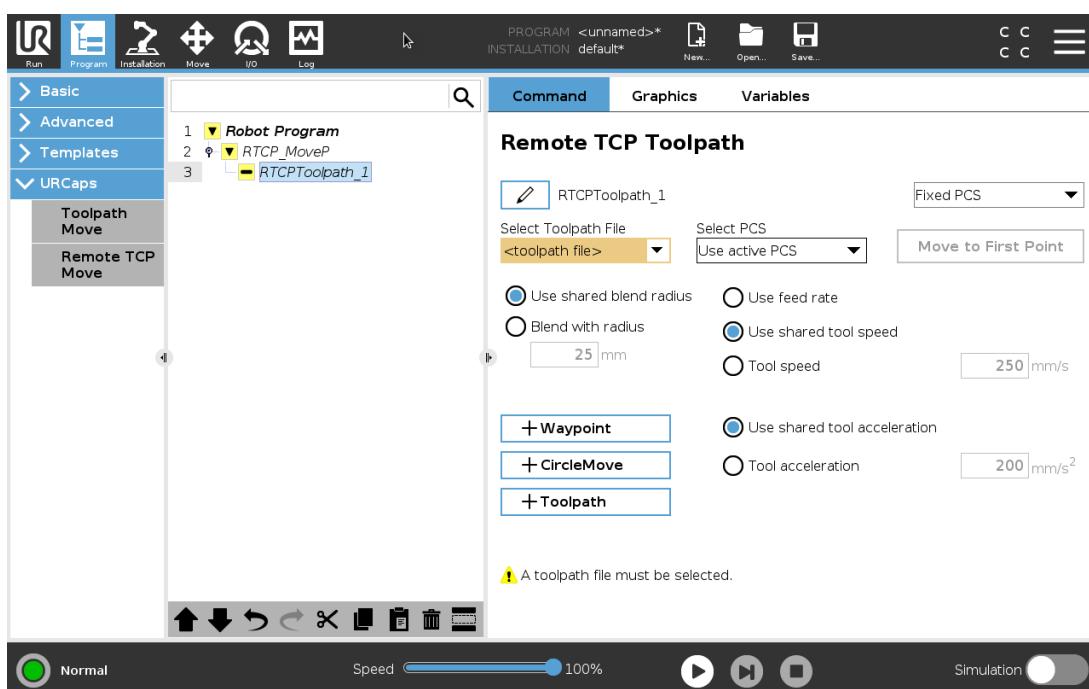
- Toolpath file
- Remote TCP
- Remote TCP PCS

## Configuring a Toolpath using CAD/CAM Software

A toolpath defines the orientation, trajectory, 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.

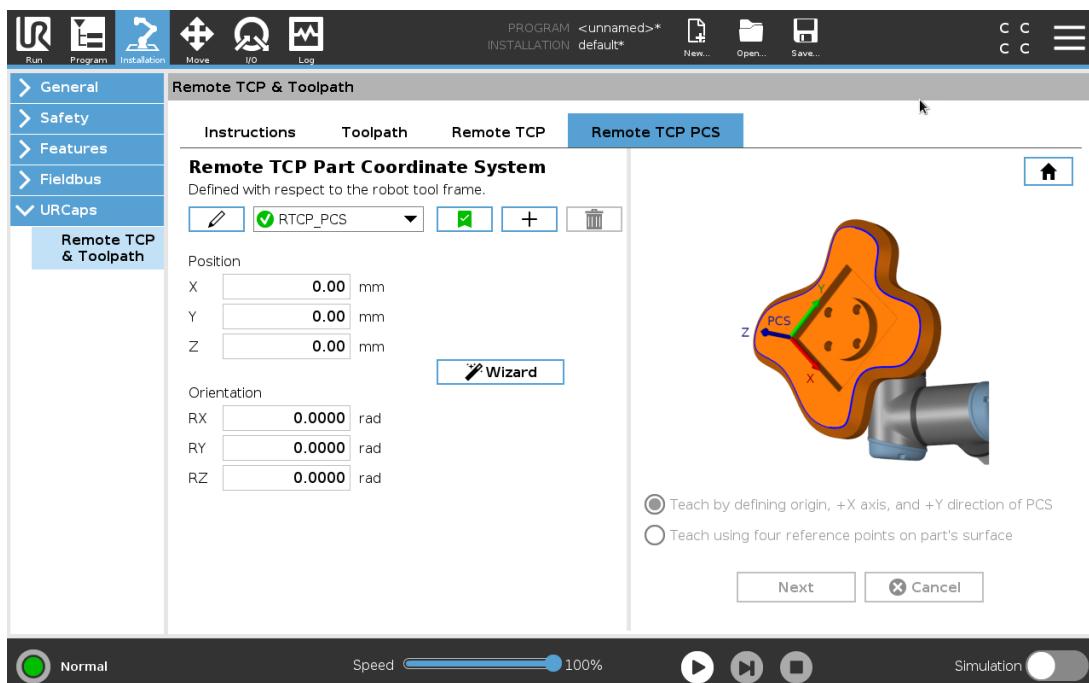
### 23.13.5. Remote TCP

#### 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 achieved.
6. Envision the tool orientation at the first waypoint on the physical part. The positive Z-axis direction should point away from the 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.

### 23.13.6. 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.



## 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 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.

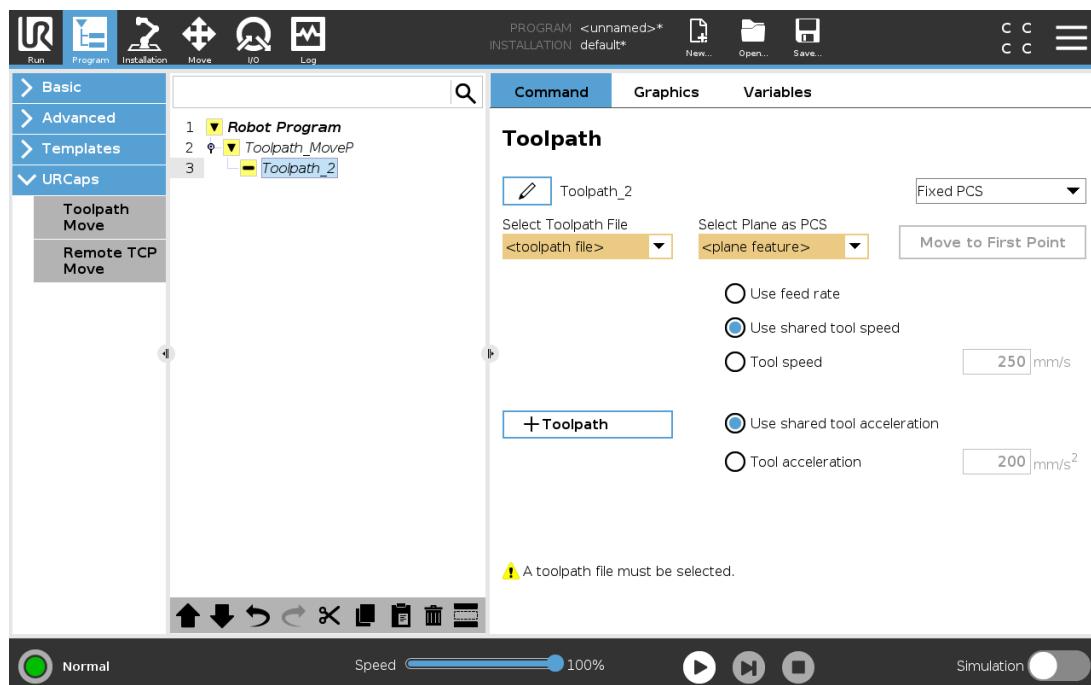
## Setting a Variable PCS

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.

The following section explains how to use a variable PCS in a Remote TCP Toolpath node.

## Configuring 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 from the 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.



#### NOTE

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 [23.10.1. Move](#) on page 146

### 23.13.7. Regular TCP Toolpath Moves

Similar to configuring a Remote TCP Toolpath Move, a regular TCP Toolpath Move requires the following:

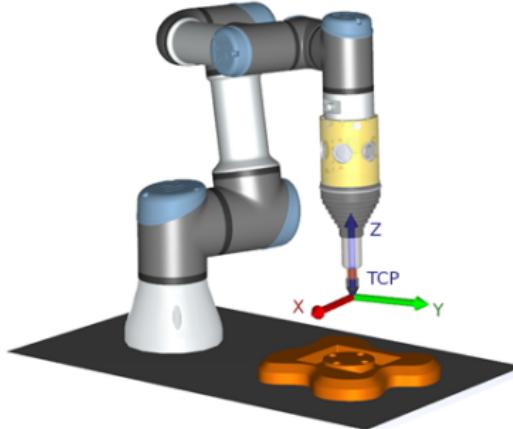
- Toolpath file
- Regular TCP
- Plane Feature as a PCS

### Configuring and Importing a Toolpath File

This is similar to configuring a Toolpath (see [Configuring a Toolpath using CAD/CAM Software on page 190](#)) and importing Toolpath (see [Importing a G-code Toolpath into PolyScope on page 190](#)).

## Configuring a Regular TCP

- Follow the instructions in [24.2. TCP Configuration on page 197](#) to configure a Regular TCP.
- Verify the positive Z-axis of the tool points away from the part surface.



## Configuring a Plane Feature PCS

- Create a plane feature by [Adding a plane](#) or [Teaching a plane](#). See [24.17.5. Plane Feature on page 218](#).
- Fix the part relative to the robot base.
- Verify the correct TCP is used to create the plane feature. For high accuracy, temporarily set up a sharp Remote TCP to complete this teaching process.
- 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.
- Finish the teaching process and confirm the PCS position and orientation.

## Configuring a Toolpath Node

- Access the Program Tab and tap **URCaps**.
- 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.
- Tap **+Toolpath** to insert a Toolpath node.
- In the drop-down menu, select a toolpath file and the corresponding PCS (Plane Feature).
- Adjust the motion parameters if different values are to be applied to the Toolpath node.
- Tap **Move to First Point** to verify the tool can move to the first point of the toolpath.
- Run the program in the simulation mode, at a low speed, to confirm the configurations are correct.

**NOTE**

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 [23.10.1. Move on page 146](#)

# 24. Installation Tab

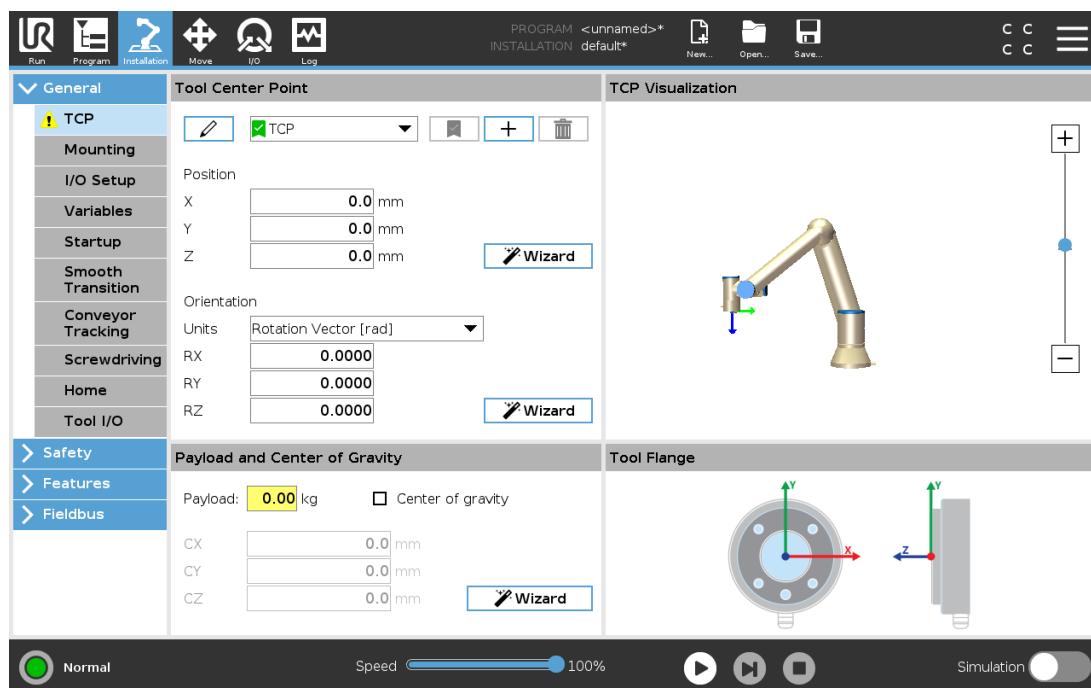
## 24.1. General

The Installation Tab allows you to configure the settings which affect the overall performance of the robot and PolyScope.

## 24.2. TCP Configuration

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.



### 24.2.1. 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.

### 24.2.2. 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 (see ).

## 24.2.3. 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 [23.10.1. Move on page 146](#)) or a Set command. The motion of the active TCP is visualised on the Graphics Tab (see [23.8. Graphics Tab on page 143](#)).

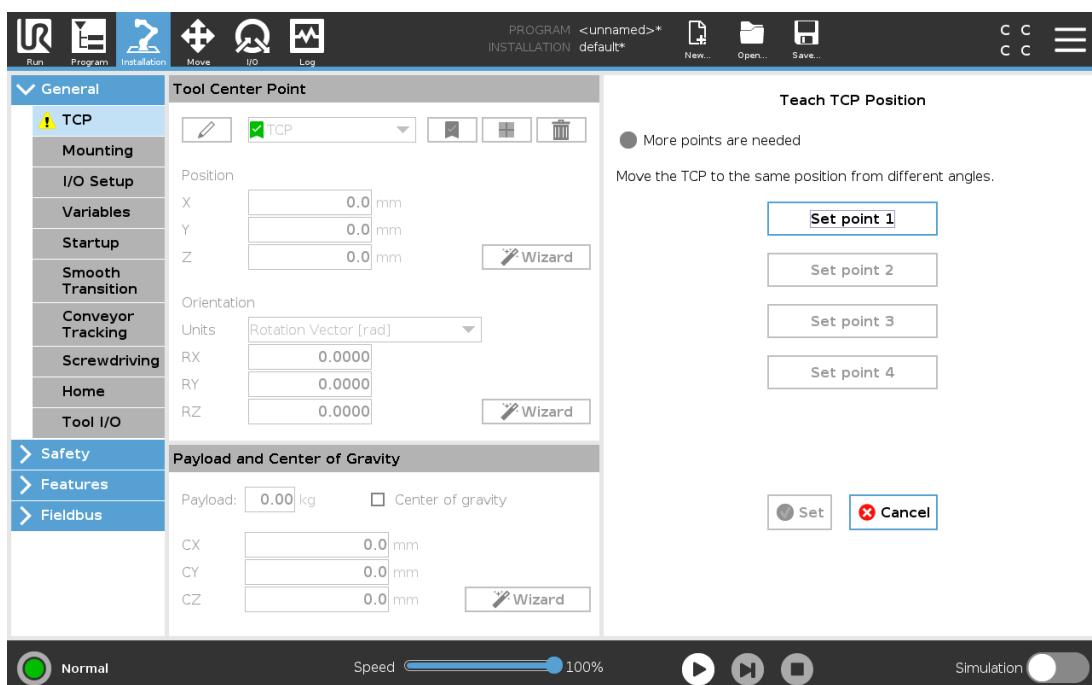
### Default TCP

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.

## 24.2.4. Teaching TCP position

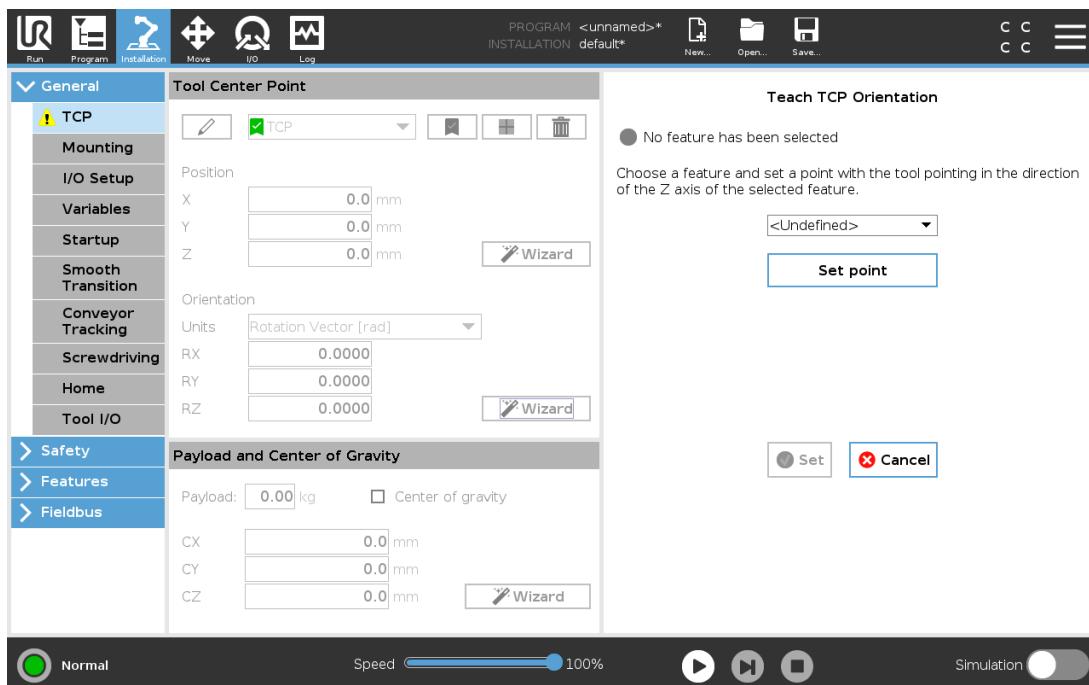


TCP position coordinates can be calculated automatically as follows:

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.

## 24.2.5. Teaching TCP orientation

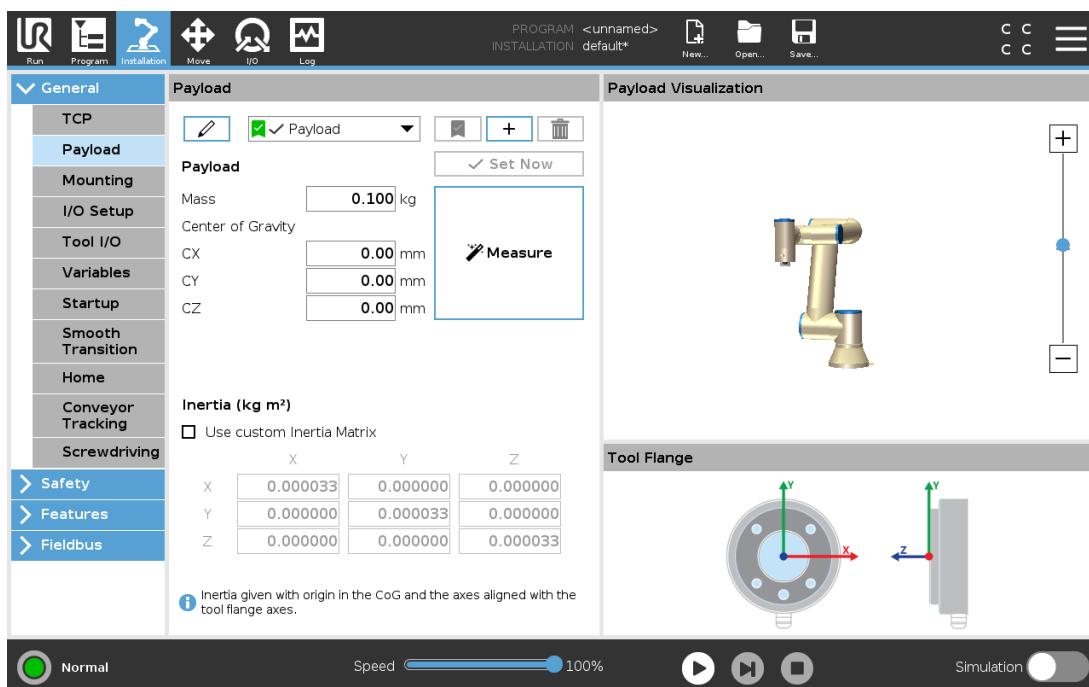


1. Tap **Measure**.
2. Select a feature from the drop-down list. (See [24.17. Features on page 215](#)) for additional information on defining new features
3. Tap **Set point** and use **Move tool arrows** to a position where the tool's orientation and the corresponding TCP coincide with the selected feature's coordinate system.
4. Verify the calculated TCP orientation and apply it to the selected TCP by tapping **Set**.

## 24.3. Payload

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.



### 24.3.1. Adding, Renaming, Modfyng 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 **edit** to rename a Payload.
- Tap the **trash** 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**.

### 24.3.2. 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.

### 24.3.3. 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**.



##### NOTE

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

### 24.3.4. Inertia

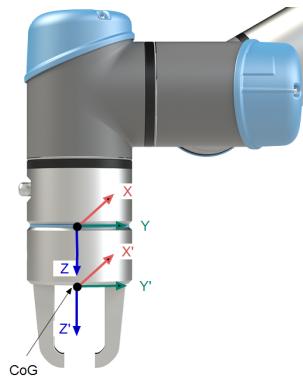
#### Setting Inertia Values

You can select **Use custom Inertia Matrix** to set inertia values.

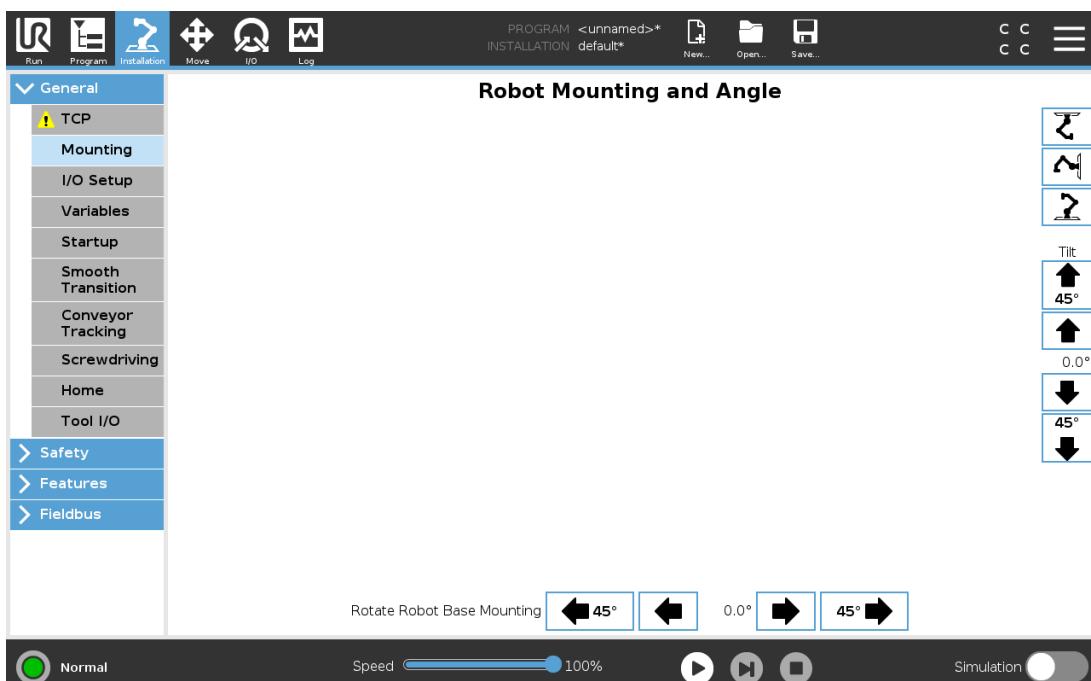
Tap the fields: **I<sub>XX</sub>**, **I<sub>YY</sub>**, **I<sub>ZZ</sub>**, **I<sub>XY</sub>**, **I<sub>XZ</sub>** and **I<sub>YZ</sub>** to set the inertia for the selected Payload.

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 1g/cm<sup>3</sup>



## 24.4. Mounting



Specifying the mounting of the Robot arm serves two purposes:

1. Making the Robot arm appear correctly on screen.
2. Telling the controller about the direction of gravity.

An advanced dynamics model gives the Robot arm smooth and precise motions, as well as allows the Robot arm to hold itself in **Freedrive Mode**. For this reason, it is important to mount the Robot arm correctly.



### WARNING

Failure to mount the Robot's arm correctly may result in frequent Protective Stops, and/or the Robot arm will move when pressing the **Freedrive** button.

If the Robot arm is mounted on a flat table or floor, no change is needed on this screen. However, if the Robot arm is **ceiling mounted**, **wall mounted**, or **mounted at an angle**, this needs to be adjusted using the buttons.

The buttons on the right side of the screen are for setting the angle of the Robot arm's mounting. The top three right side buttons set the angle to **ceiling** ( $180^\circ$ ), **wall** ( $90^\circ$ ), **floor** ( $0^\circ$ ). The **Tilt** buttons set an arbitrary angle.

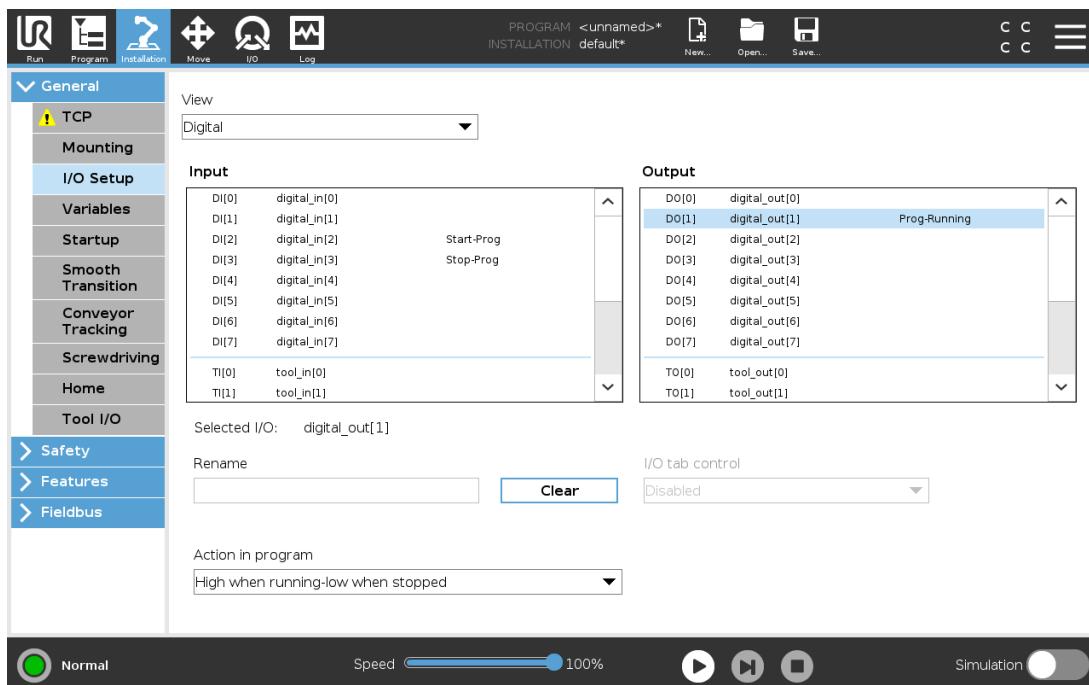
The buttons on the lower part of the screen are used to rotate the mounting of the Robot arm to match the actual mounting.



### WARNING

Use the correct installation settings. Save and load the installation files with the program.

## 24.5. I/O Setup



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.

### 24.5.1. 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.

## 24.5.2. 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 ([23.10.8. Wait on page 159](#)) and ([23.11.2. If on page 164](#)), respectively. You can find named general purpose registers in the **Input** or **Output** selector on the **Expression Editor** screen.

## 24.5.3. I/O Actions and I/O Tab Control

### Input and Output Actions

You can use Physical and Fieldbus digital I/Os to trigger actions or react to the status of a program.

Available Input Actions:

Input	Action
Start	Starts or resumes the current program on a rising edge (only enabled in Remote Control, see <a href="#">29.4.5. Remote Control on page 251</a> )
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

Action	Output state	Program state
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)



### NOTE

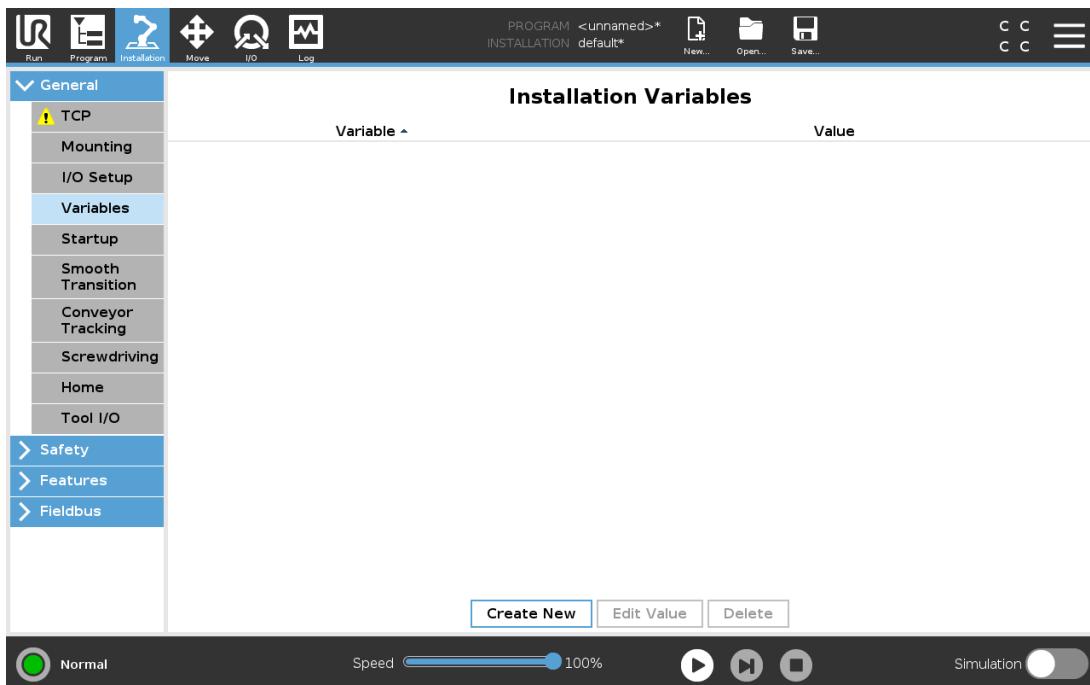
A program is terminated unscheduled if any of the following occur:

- Protective stop
- Fault
- Violation
- Runtime exception

## 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.

## 24.6. Installation Variables



Variables created on the **Installation Variables** screen 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.

Installation variables 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.



### Creating a Variable

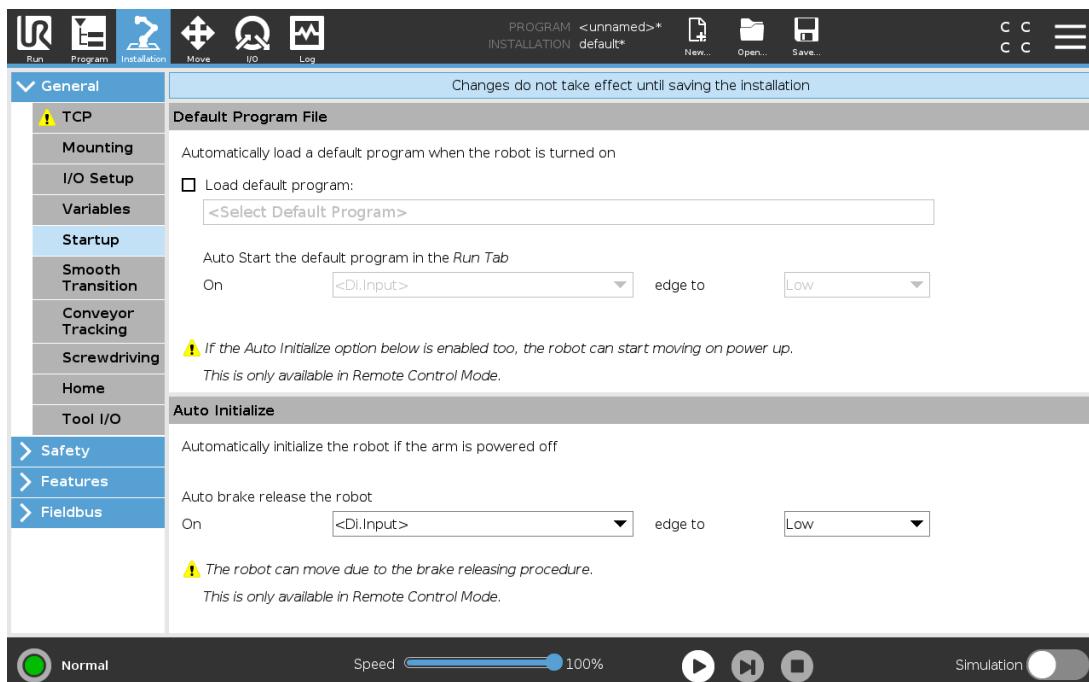
1. Tap **Create New** to access the **Create new Installation variable** field with a suggested name for the new variable.
2. Tap the **Create new Installation variable** field to change the variable name.
3. Tap **OK** if the new variable name is unused in this installation.
4. Select the variable in the list, then tap **Edit Value** to change the value of the installation variable.
5. Once the variable is configured, you must save the installation itself to keep the configuration.

### Deleting a variable

1. Select the variable and tap **Delete**.

If a loaded program has the same name as a program variable, or if a loaded installation has the same name as the installation variable, you are presented the following options: You can either resolve the issue using the installation variables of the same name instead of the program variable or resolve the issue by having the conflicting variables renamed automatically.

## 24.7. Startup



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.

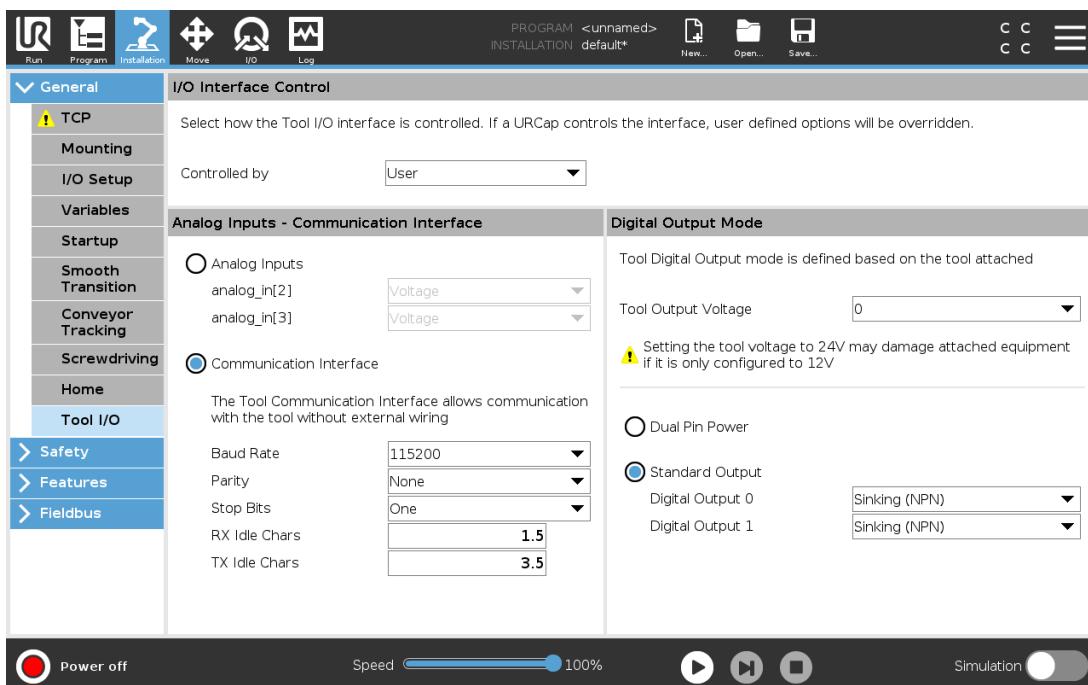
### 24.7.1. Loading a Startup Program

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 [22. Run Tab on page 133](#)) is entered and no program is loaded.

## 24.7.2. 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.

## 24.8. Tool I/O



## 24.9. I/O Interface Control

The **I/O Interface Control** allows you to switch between user control and URcap 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.



### NOTE

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.

## 24.10. Tool Analog Inputs

### 24.10.1. Tool Communication Interface

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.

### 24.10.2. Configuring the Tool Communication Interface (TCI)

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.

## 24.11. Digital Output Mode

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. See chapter five in the [Hardware Installation Manual](#).
- **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.

## 24.11.1. Dual Pin Power

Dual Pin Power is used as a source of power for the tool. Enabling Dual Pin Power disables default tool digital outputs.

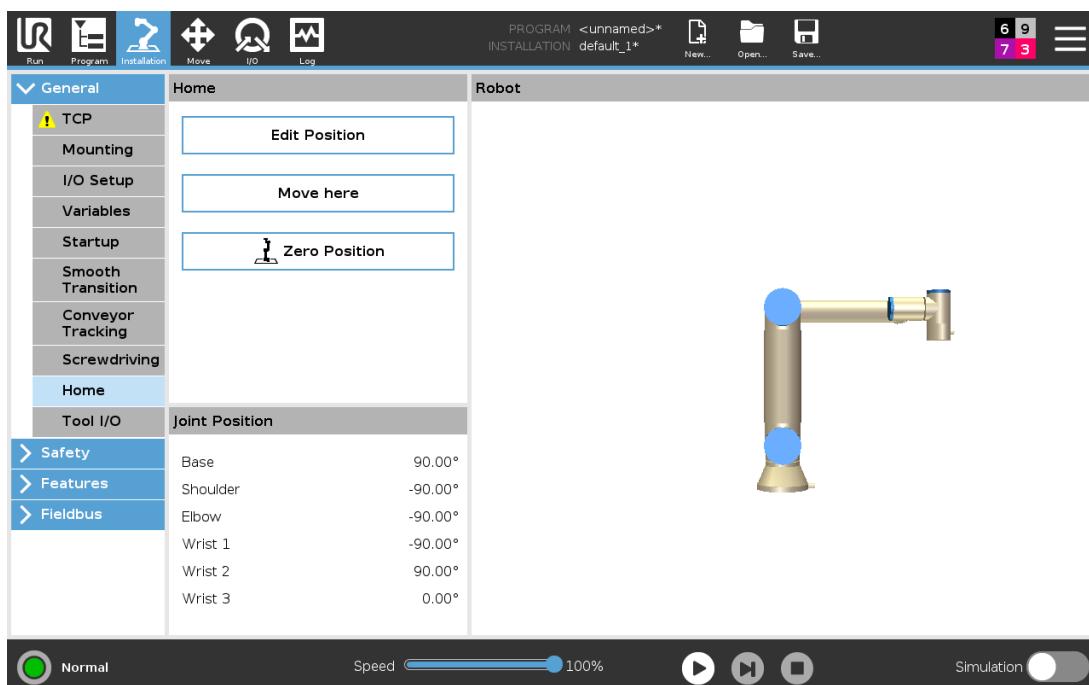
## 24.12. Smooth Transition Between Safety Modes

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.

### 24.12.1. Adjusting Acceleration/Deceleration Settings

1. In the Header, tap **Installation**.
2. In the In the Side Menu on the left, under **General**, select **Smooth Transition**.
3. Select **Hard** to have a higher acceleration/deceleration or select **Soft** for the smoother default transition setting.

## 24.13. 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 21.18. Safe Home Position on page 131) 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.

### 24.13.1. Defining 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.

## 24.14. Conveyor Tracking Setup

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.

### 24.14.1. 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** (section [24.14.2. Conveyor Parameters below](#)) and **Tracking Parameters** (section [24.14.3. Tracking Parameters below](#)).

### 24.14.2. Conveyor Parameters

#### *Incremental*

encoders can be connected to Digital Inputs 8 to 11. Decoding of digital signals runs at 40kHz. 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 can be connected through a MODBUS signal. This requires a Digital MODBUS Output register preconfigured in (section [24.19. MODBUS Client I/O Setup on page 224](#)).

### 24.14.3. 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.

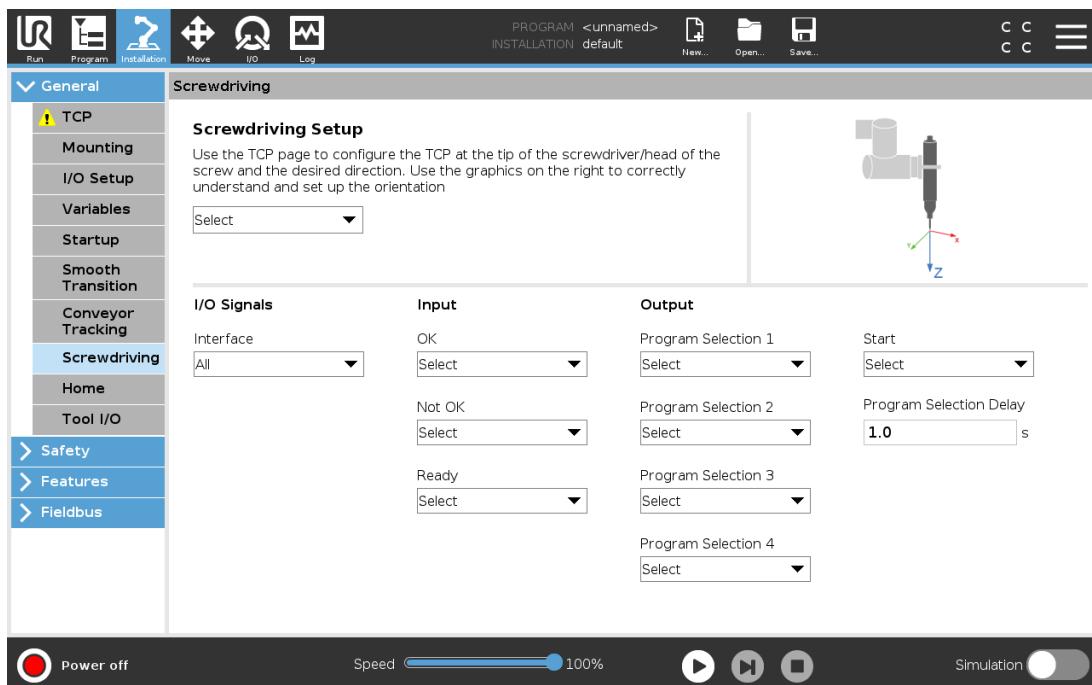
#### *Circular Conveyors*

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.

## 24.15. Screwdriving Setup

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.



## 24.15.1. Configuring a Screwdriver

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.

In any output Program Selection list under Output, you can select an integer output to switch Program Selection (see [23.12.8. Screwdriving on page 184](#)) to a number field.

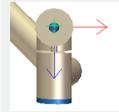
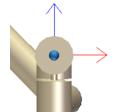
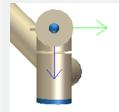
## 24.15.2. Configuring the Screwdriver Position

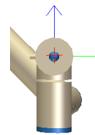
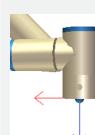
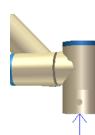
1. Under **Screwdriving Setup**, use the drop-down menu to select a previously defined TCP (see [24.2. TCP Configuration on page 197](#)) where Position and Orientation are set up as follows:
  - Configure the Position to be the tip of the screwdriver tool where it contacts the screw.
  - Configure the 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 (see [23.12.8. Screwdriving on page 184](#)) uses the positive Z direction of the selected TCP to follow the screw and calculate distances.

Typical Orientation values (in Rotation Vector [rad] notation) are illustrated in the following table.

Screwdriving axis parallel to the negative Y direction of the robot's tool flange		Orientation <ul style="list-style-type: none"> <li>• RX: 1.5708 rad</li> <li>• RY: 0.0000 rad</li> <li>• RZ: 0.0000 rad</li> </ul>
Screwdriving axis parallel to the positive Y direction of the robot's tool flange		Orientation <ul style="list-style-type: none"> <li>• RX: -1.5708 rad</li> <li>• RY: 0.0000 rad</li> <li>• RZ: 0.0000 rad</li> </ul>
Screwdriving axis parallel to the positive X direction of the robot's tool flange		Orientation <ul style="list-style-type: none"> <li>• RX: 0.0000 rad</li> <li>• RY: 1.5708 rad</li> <li>• RZ: 0.0000 rad</li> </ul>

Screwdriving axis parallel to the negative X direction of the robot's tool flange		Orientation	<ul style="list-style-type: none"><li>• RX: 0.0000 rad</li><li>• RY: -1.5708 rad</li><li>• RZ: 0.0000 rad</li></ul>
Screwdriving axis parallel to the positive Z direction of the robot's tool flange		Orientation	<ul style="list-style-type: none"><li>• RX: 0.0000 rad</li><li>• RY: 0.0000 rad</li><li>• RZ: 0.0000 rad</li></ul>
Screwdriving axis parallel to the negative Z direction of the robot's tool flange		Orientation	<ul style="list-style-type: none"><li>• RX: 3.1416 rad</li><li>• RY: 0.0000 rad</li><li>• RZ: 0.0000 rad</li></ul>

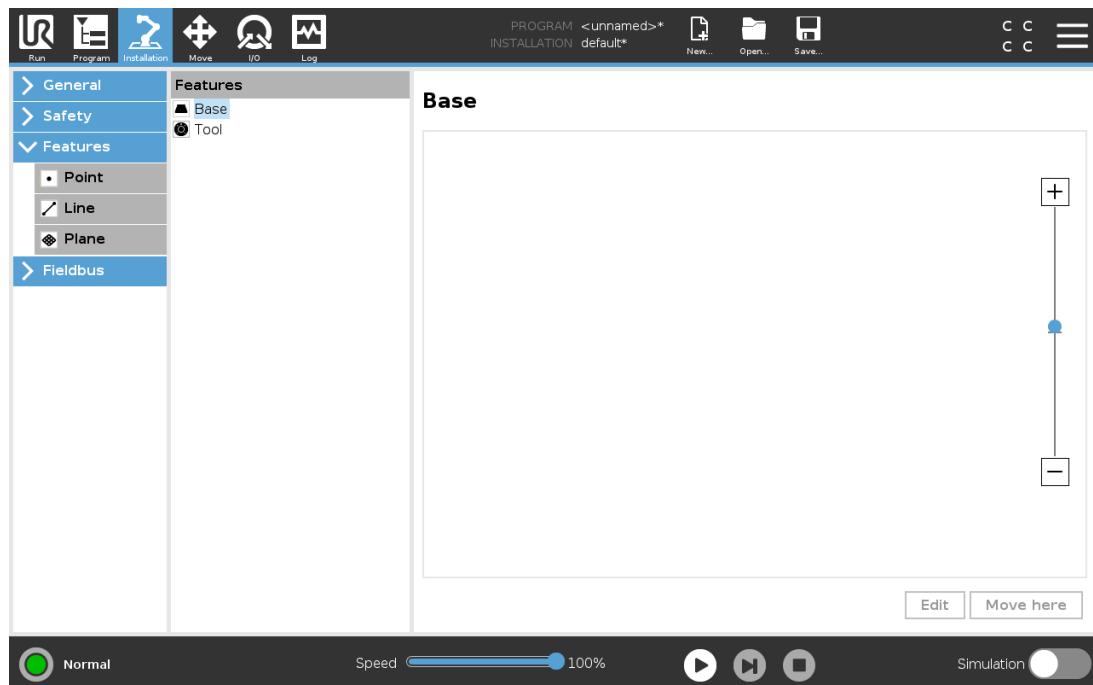
### 24.15.3. Configuring the Screwdriver Interface

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
  - NOK: 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

## 24.16. Safety

See chapter [21. Safety Configuration on page 115](#).

## 24.17. Features

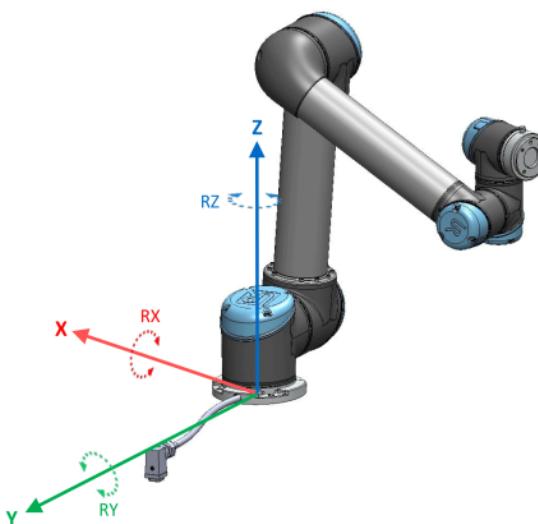


A **Feature** represents an object defined by a six dimensional pose (position and orientation) relative to the robot base. You can name a feature for future reference.

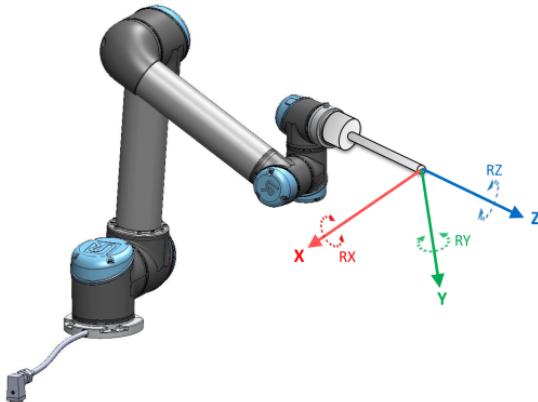
Some subparts of a robot program consist of movements executed relative to specific objects other than the base of the robot arm. These objects can be tables, other machines, workpieces, vision systems, blanks, or boundaries existing around the robot arm.

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 (see figure 13.1).
- The Tool feature is located with origin in the centre of the current TCP (see figure 13.2).



## 13.1: Base feature



## 13.2: Tool (TCP) feature

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: [on the facing page](#)), ([on page 218](#)) and ([24.17.5. Plane Feature on page 218](#)).

## 24.17.1. 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 [23.10.1. Move on page 146](#)).

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 [24.17.6. Example: Manually Updating a Feature to Adjust a Program on page 219](#)) and ([24.17.7. Example: Dynamically Updating a Feature Pose on page 220](#)). When a feature is chosen as a reference, the Move Tool buttons for translation and rotation operate in the selected feature space (see [25.3. Tool Position on page 230](#)) and ([25.1. Move Tool on page 229](#)), 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.

## 24.17.2. 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.

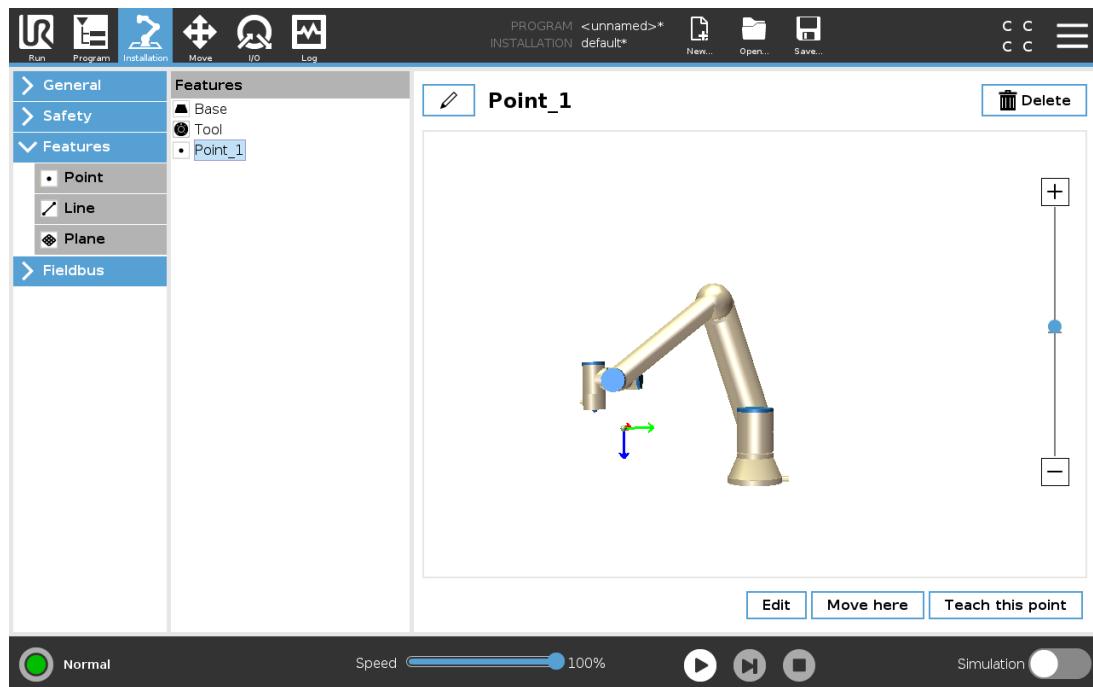
**Move here** is disabled if the robot arm cannot reach the feature.

## 24.17.3. 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.

### Adding a Point

1. In Installation, select **Features**.
2. Under Features select **Point**.

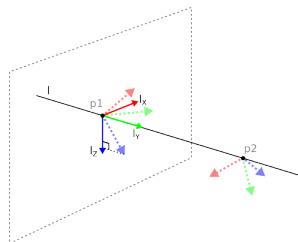


## 24.17.4. Line feature

The line feature defines lines that the robot needs to follow. (e.g., when using conveyor tracking). A line is defined as an axis between two point features  $p_1$  and  $p_2$  as shown in figure 13.3.

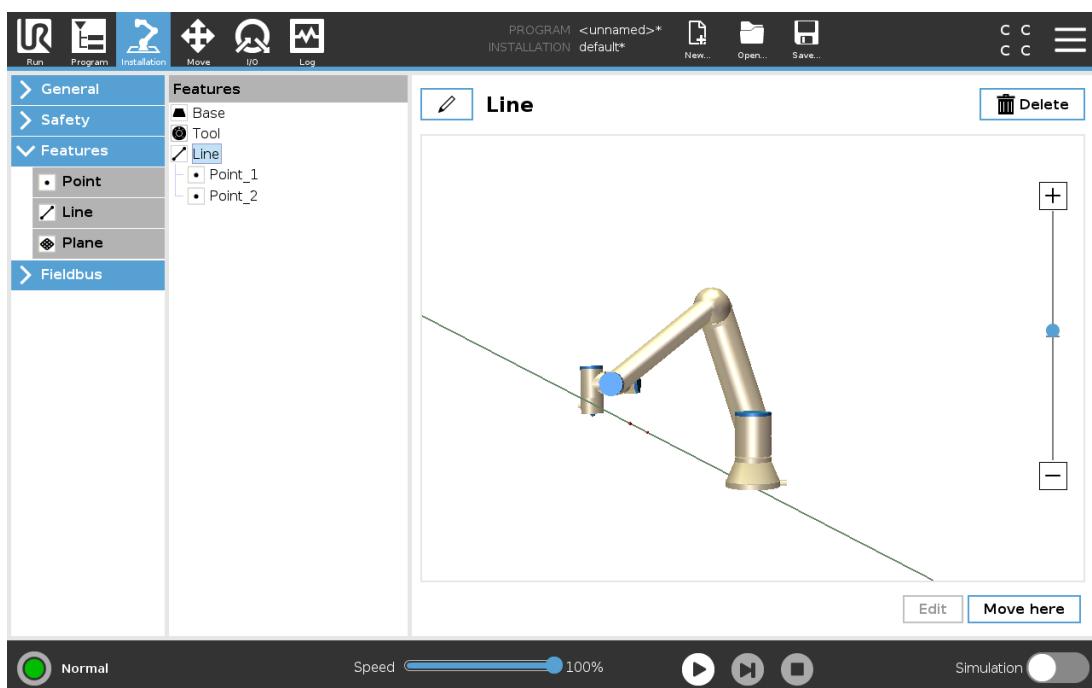
## Adding a Line

1. In Installation, select **Features**.
2. Under Features select **Line**.



13.3: Definition of the line feature

In figure 13.3 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  $p_1$  onto the plane perpendicular to the line. The position of the line coordinate system is the same as the position of  $p_1$ .



## 24.17.5. 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

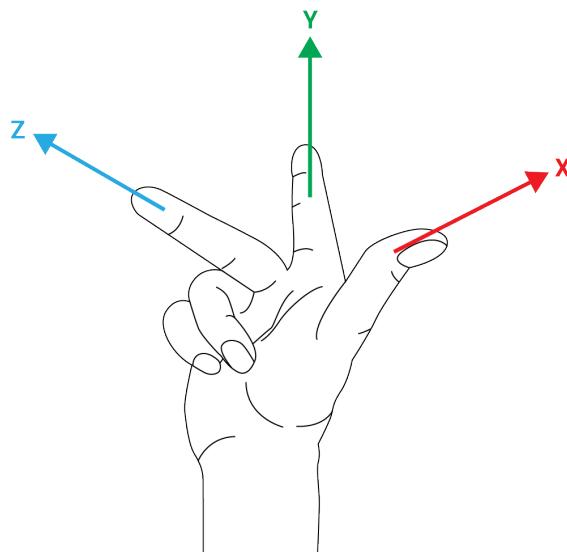
1. In Installation, select **Features**.
2. Under Features select **Plane**.

## 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 Orig0
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.



### NOTE

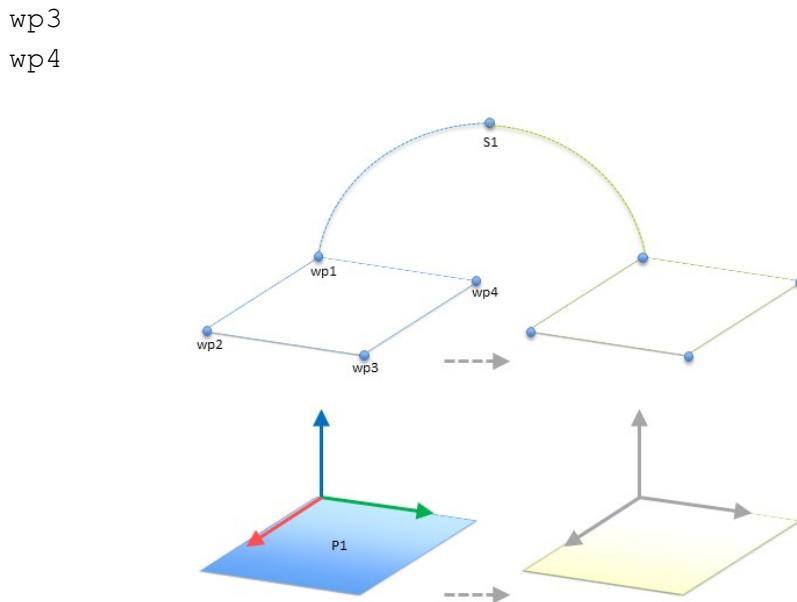
You can re-teach the plane in the opposite direction of the x-axis, if you want that plane to be normal in the opposite direction.

Modify an existing plane by selecting Plane and pressing Modify Plane. You will then use the same guide as for teaching a new plane.

### 24.17.6. Example: Manually Updating a Feature to Adjust a Program

Consider an application where multiple parts of a robot program is relative to a table. Figure 13.4 illustrates the movement from waypoints wp1 through wp4.

```
Robot Program
MoveJ
  S1
MoveL # Feature: P1_var
  wp1
  wp2
```



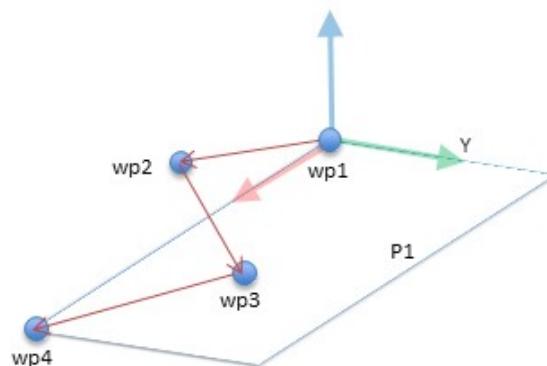
13.4: Simple program with four waypoints relative to a feature plane manually updated by changing the feature

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.

#### 24.17.7. 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 (see 13.5).



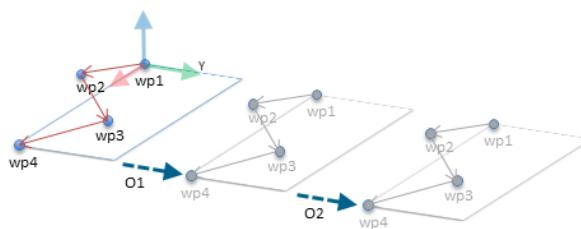
13.5: A *MoveL* command with four waypoints relative to a plane feature

## 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

```



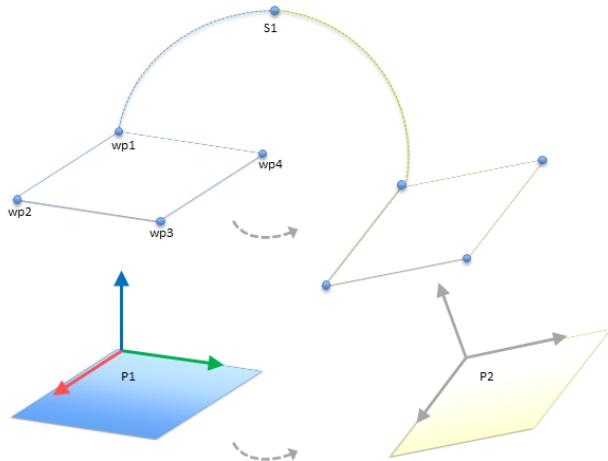
13.6: Applying an offset to the plane feature

## Robot Program

```

MoveJ
    S1
    if (digital_input[0]) then
        P1_var = P1
    else
        P1_var = P2
MoveL # Feature: P1_var
    wp1
    wp2
    wp3
    wp4

```

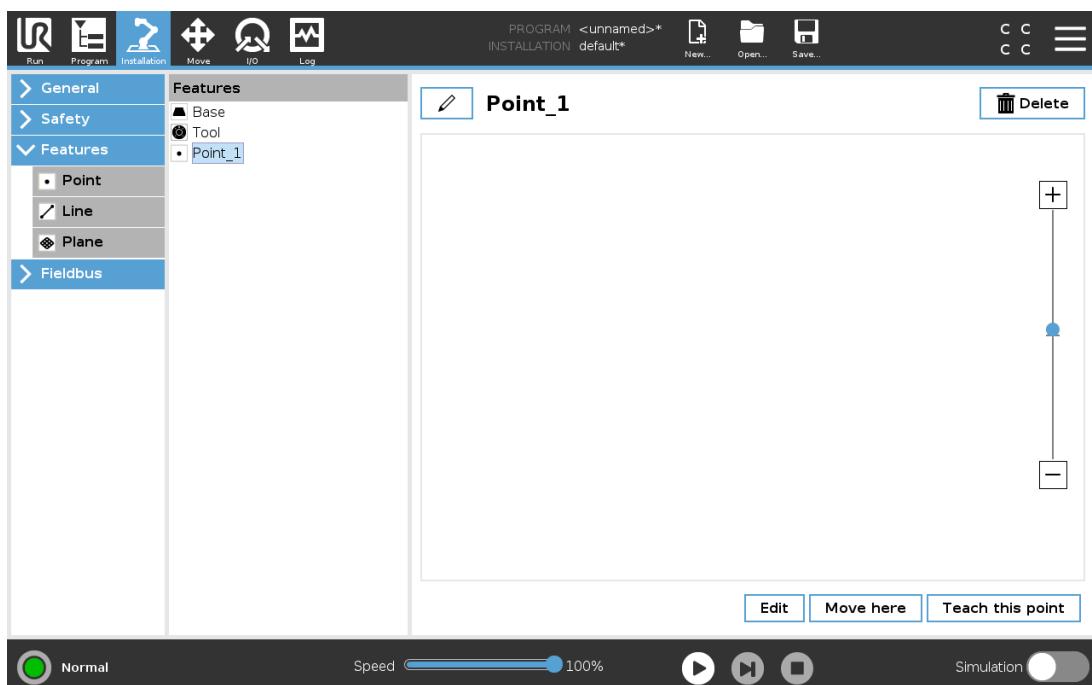


13.7: Switching from one plane feature to another

The movement relative to  $P1$  is repeated a number of times, each time by an offset  $o$ . In this example the offset is set to 10 cm in the Y-direction (see figure 13.6, offsets  $O1$  and  $O2$ ). This is achieved using `pose_add()` or `pose_trans()` script functions to manipulate the variable. It is possible to switch to a different feature while the program is running instead of adding an offset. This is shown in the example below (see figure 13.7) where the reference feature for the *MoveL* command  $P1\_var$  can switch between two planes  $P1$  and  $P2$ .

## 24.17.8. Feature Edit

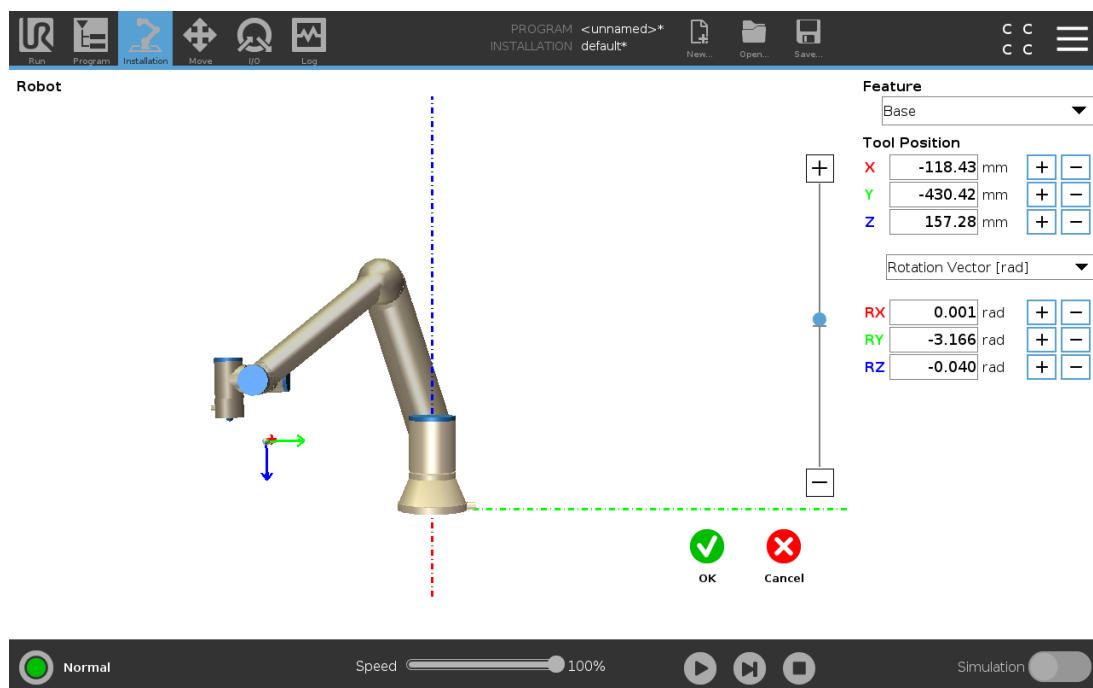
The Feature Edit is an alternative way to add features to your installation and/or edit existing features.



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

## 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.

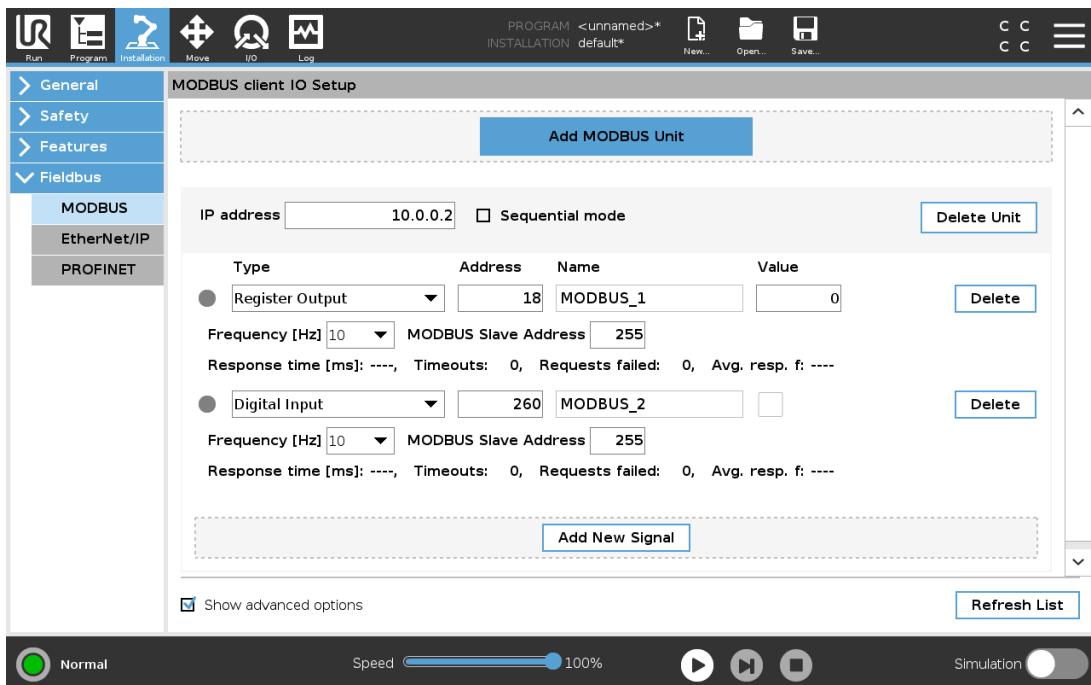
## Editing a Plane

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

## 24.18. Fieldbus

Here you can set the family of industrial computer network protocols used for real-time distributed control accepted by PolyScope: MODBUS, Ethernet/IP and PROFINET.

## 24.19. MODBUS Client I/O Setup



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.

### 24.19.1. Refresh

Push this button to refresh all MODBUS connections. Refreshing disconnects all modbus units, and connects them back again. All statistics are cleared.

### 24.19.2. Add unit

Push this button to add a new MODBUS unit.

### 24.19.3. Delete unit

Push this button to delete the MODBUS unit and all signals on that unit.

### 24.19.4. Set unit IP

Here the IP address of the MODBUS unit is shown. Press the button to change it.

## 24.19.5. Sequential mode

Available only when *Show Advanced Options* (see [24.19.13. Show Advanced Options on page 227](#)) is 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 (see section [24.19.14. Advanced Options on page 227](#)). The signal indicator turns yellow if the actual signal frequency is less than half of the value selected from the **Frequency** drop-down list.

## 24.19.6. Add signal

Push this button to add a signal to the corresponding MODBUS unit.

## 24.19.7. Delete signal

Push this button to delete a MODBUS signal from the corresponding MODBUS unit.

## 24.19.8. 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.

## 24.19.9. 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.

## 24.19.10. 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.

## 24.19.11. 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.

## 24.19.12. 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.

## 24.19.13. Show Advanced Options

This check box shows/hides the advanced options for each signal.

## 24.19.14. 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.

## 24.20. EtherNet/IP

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.

## 24.21. PROFINET

PROFINET is a network protocol that 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*

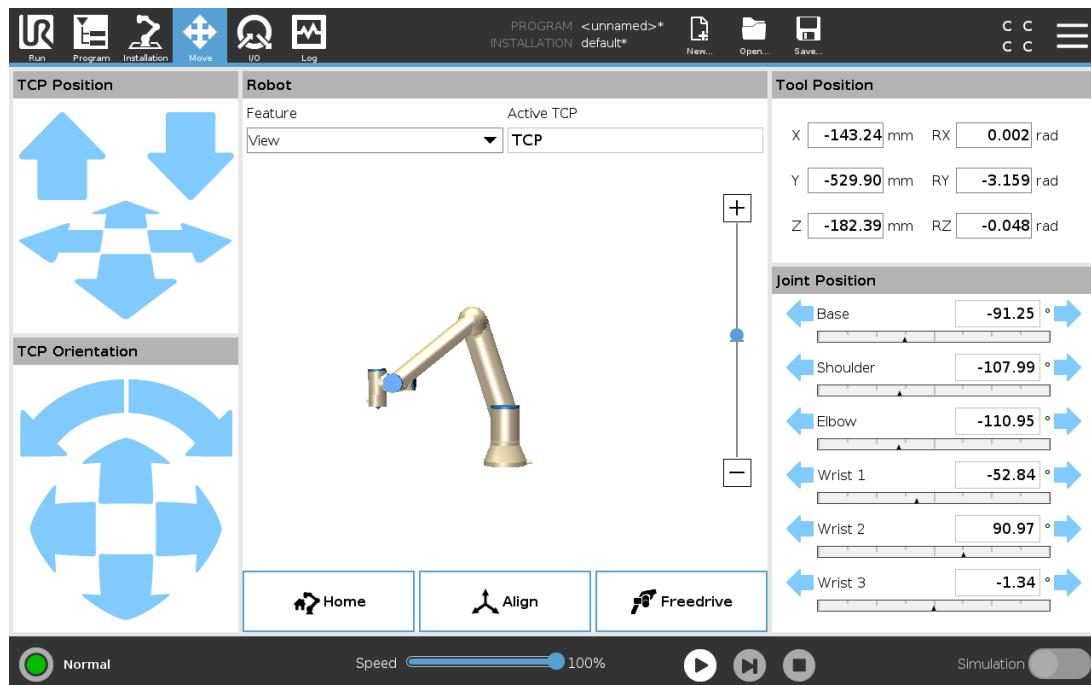
PolyScope pauses the current program. The program resumes from where it stopped.

*Stop*

PolyScope stops the current program.

# 25. Move Tab

On this screen, you can move (jog) the robot arm directly, either by translating/rotating the robot tool, or by moving robot joints individually.



## 25.1. Move Tool

Hold down any of the **Move Tool** arrows to move the robot arm in a particular direction.

- The **Translate arrows** (upper) move the robot tool-tip in the direction indicated.
- The **Rotate arrows** (lower) change the orientation of the robot 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 robot's tool. The TCP is shown as a small blue ball.

## 25.2. Robot

If the current position of the robot TCP comes close to a safety or trigger plane, or the orientation of robot tool is near the tool orientation boundary limit (see [21.11. Planes on page 121](#)), a 3D representation of the proximate boundary limit is shown. When the robot is running a program, the visualization of boundary limits is disabled.

Safety planes are visualized in yellow and black with a small arrow representing the plane normal, which indicates the side of the plane on which the robot TCP is allowed to be positioned. Trigger planes are displayed in blue and green and a small arrow pointing to the side of the plane, where the **Normal** mode limits (see [21.8. Safety Modes on page 119](#)) 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.

## 25.2.1. 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.

## 25.2.2. Active TCP

In the **Robot** field, under **Active TCP**, the name of the current active Tool Center Point (TCP) is displayed.

## 25.2.3. Home

The **Home** button accesses the **Move Robot into Position** screen, where you can hold down the **Auto** button (see [22.4.2. Move robot to: on page 135](#)) to move robot into position previously defined under Installation (see [24.13.1. Defining Home on page 211](#)). The Home button's default setting returns the Robo Arm to an upright position (see [24.13. Home on page 210](#)).

## 25.2.4. Freedrive

The on-screen **Freedrive** button allows the Robot Arm to be pulled into desired positions/poses.

## 25.2.5. Align

The **Align** button allows the Z axis of the active TCP to align to a selected feature.

## 25.3. Tool Position

The text boxes display the full coordinate values of the TCP relative to the selected feature. You can configure several named TCPs (see [24.2. TCP Configuration on page 197](#)). You can also tap **Edit pose** to access the **Pose Editor** screen.

## 25.4. Joint Position

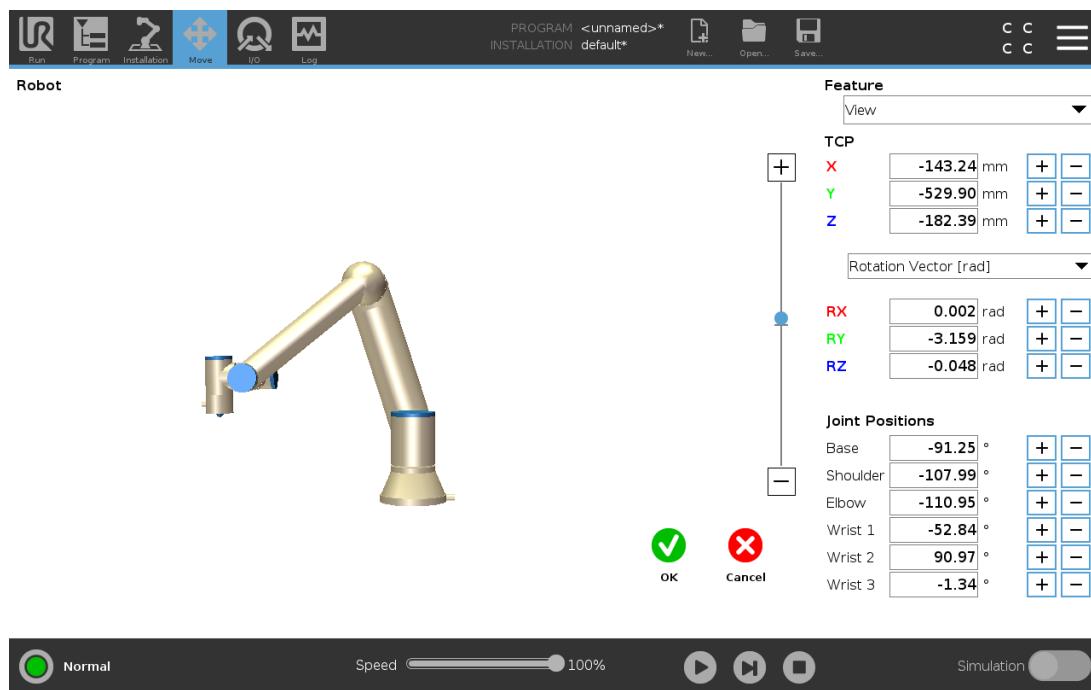
The **Joint Position** field allows you to directly control individual joints. Each joint moves along a default joint limit range from  $-360^\circ$  to  $+360^\circ$ , defined by a horizontal bar. Once the limit is reached you cannot move a joint any further. You can configure joints with a position range different from the default (see [21.10. Joint Limits on page 120](#)), this new range is indicated with red zone inside the horizontal bar.


**WARNING**

1. In the **Setup** tab, if the gravity setting (see [24.4. Mounting on page 202](#)) is wrong, or the robot arm carries a heavy load, the robot arm can start moving (falling) when you press the **Freedrive** tab. In that case, release **Freedrive** again.
2. Use the correct installation settings (e.g. Robot mounting angle, payload mass and payload center of gravity offset). Save and load the installation files along with the program.
3. Payload settings and robot mounting settings must be set correctly before operating the **Freedrive** button. If these settings are not correct, the robot arm will move when you activate **Freedrive**.
4. The **Freedrive** function must only be used in installations where the risk assessment allows it. Tools and obstacles must not have sharp edges or pinch points. Make sure that all personnel remain outside the reach of the robot arm.

## 25.5. Pose Editor Screen

Once you access the **Pose Editor** screen, you can precisely configure a target joint positions, or a target pose (position and orientation) for the TCP. Note: This screen is **offline** and does not control the Robot Arm directly.



### Robot

The 3D image shows the current Robot Arm position. The **shadow** shows the Robot Arm target position controlled by the specified values on the screen. Press the magnifying glass icons to zoom in/out or drag a finger across it to change the view.

If the specified target position of the robot TCP is close to a safety or trigger plane, or the orientation of

robot tool is near the tool orientation boundary limit (see [21.11. Planes on page 121](#)), a 3D representation of the proximate boundary limit is shown. Safety planes are visualized in yellow and black with a small arrow representing the plane normal, which indicates the side of the plane on which the robot TCP is allowed to be positioned. Trigger planes are displayed in blue and green and a small arrow pointing to the side of the plane, where the **Normal** mode limits (see [21.8. Safety Modes on page 119](#)) 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 target robot TCP is no longer in proximity of the limit, the 3D representation disappears. If the target TCP is in violation or very close to violating a boundary limit, the visualization of the limit turns red.

## Feature and Tool Position

The active TCP and coordinate values of the selected feature are displayed. The X, Y, Z coordinates specify tool position. The RX, RY, RZ coordinates specify orientation. For further information about configuring several named TCPs, (see [24.2. TCP Configuration on page 197](#)).

Use the drop down menu above the RX, RY and RZ boxes to choose the orientation representation type:

- **Rotation Vector [rad]** The orientation is given as a *rotation vector*. The length of the axis is the angle to be rotated in radians, and the vector itself gives the axis about which to rotate. This is the default setting.
- **Rotation Vector [°]** The orientation is given as a *rotation vector*, where the length of the vector is the angle to be rotated in degrees.
- **RPY [rad]** Roll, pitch and yaw (RPY) angles, where the angles are in radians. The RPY-rotation matrix (X, Y', Z" rotation) is given by:  
$$R_{rpy}(\gamma, \beta, \alpha) = RZ(\alpha) \cdot RY(\beta) \cdot RX(\gamma)$$
- **RPY [°]** Roll, pitch and yaw (RPY) angles, where angles are in degrees.

You can tap the values to edit the coordinates. You can also tap the + or - buttons to the right of a box to add/subtract an amount to/from the current value. Or you can hold down a button to directly increase/decrease the value.

## Joint Positions

Individual joint positions are specified directly. Each joint position can have Joint Limit range from -360° to +360°. You can configure Joint Positions as follows:

- Tap the joint position to edit the values.
- Tap the + or - buttons to the right of a box to add or subtract an amount to/from the current value.
- Hold down a button to directly increase/decrease the value.

## OK Button

If you activate this screen from the **Move** screen (see [25. Move Tab on page 229](#)), tap the **OK** button to return to the **Move** screen. The Robot Arm moves to the specified target. If the last specified value was a tool coordinate, the Robot Arm moves to the target position using movement type **MoveL**; or it uses

movement type **MoveJ** if a joint position was specified last (see [Movement Types on page 146](#)).

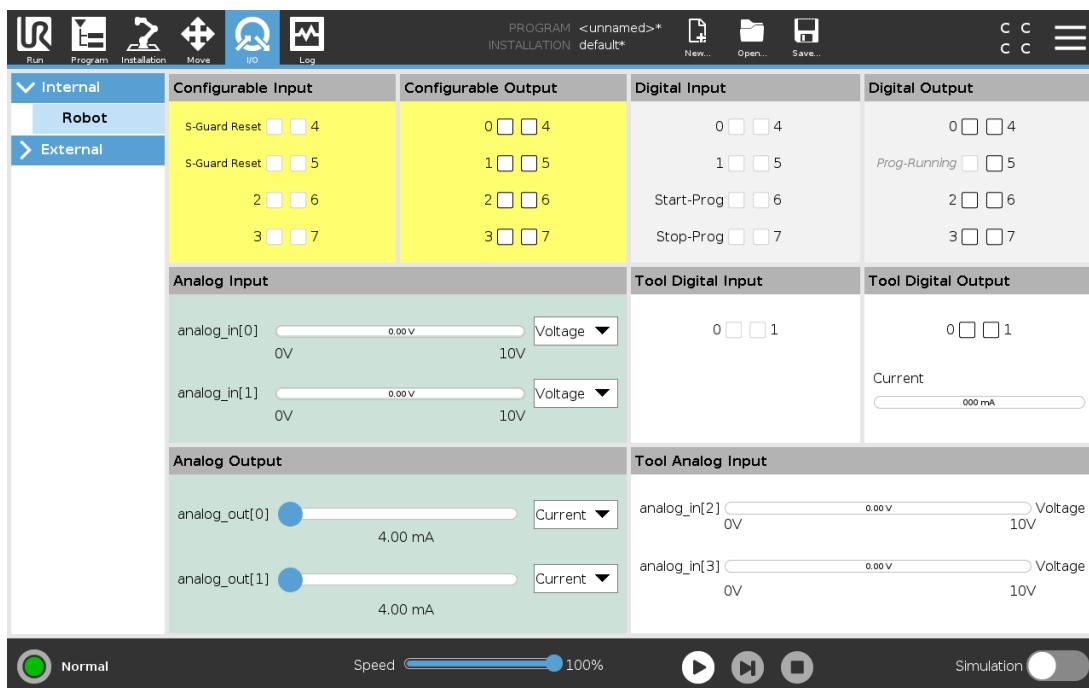
## Cancel Button

The **Cancel** button exits the screen discarding all changes.



# 26. I/O Tab

## 26.1. Robot



On this screen you can always monitor and set the live I/O signals from/to the robot control box. The screen displays the current state of the I/O, including during program execution. If anything is changed during program execution, the program will stop. At program stop, all output signals will retain their states. The screen is updated at only 10Hz, so a very fast signal might not display properly.

Configurable I/O's can be reserved for special safety settings defined in the safety I/O configuration section of the installation (see [21.16. I/O on page 128](#)); 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 toggable and will be displayed as LED's only.

The electrical details of the signals are described in chapter [21.16. I/O](#).

### Voltage

In Tool Output, Voltage can be configured only when Tool Output is controlled by the User. Selecting a URCap removes access to the Voltage.

### Analog Domain Settings

The analog I/O's can be set to either current [4-20mA] or voltage [0-10V] output. The settings will be remembered for eventual later restarts of the robot controller when a program is saved. Selecting a URCap, in Tool Output, removes access to the Domain Settings for the analog Tool Inputs.

## 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 changes as illustrated below.

Tool Analog Input	
Baud Rate	115200
Parity	None
Stop Bits	One
RX Idle Chars	1.50
TX Idle Chars	3.50



### NOTE

When the **Dual Pin Power** is enabled, the tool digital outputs must be named as follows:

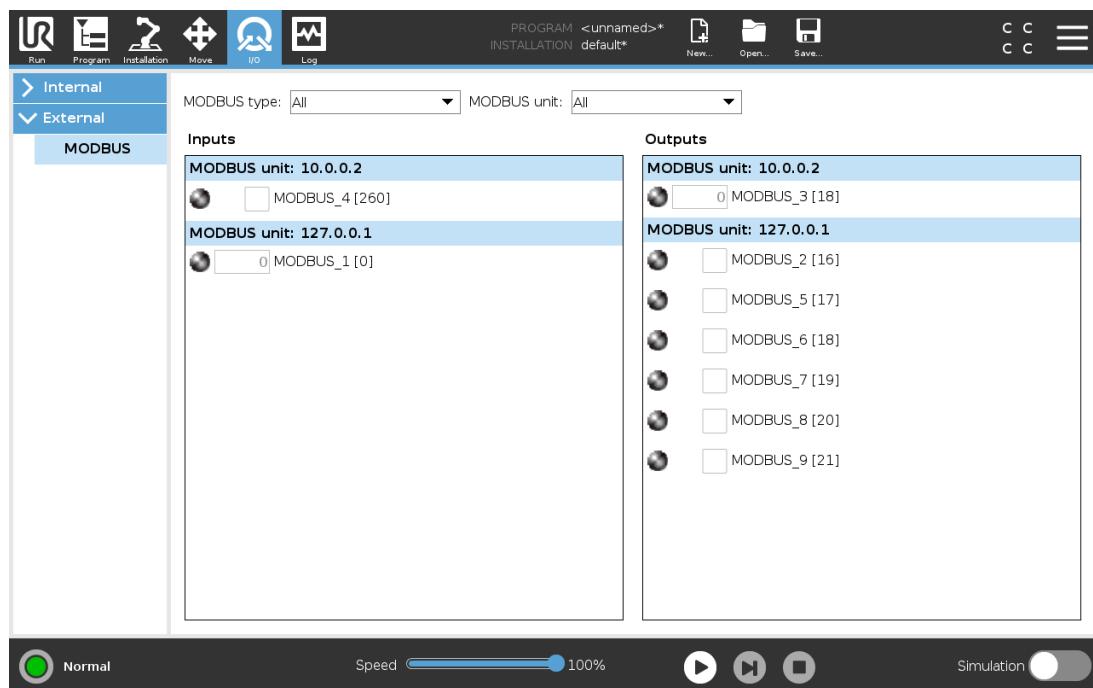
- tool\_out[0] (Power)
- tool\_out[1] (GND)

The **Tool Output** field is illustrated below.

Tool Digital Output	
Power	<input type="checkbox"/> <input checked="" type="checkbox"/> GND
Current	
	<input type="range" value="0"/> 000 mA

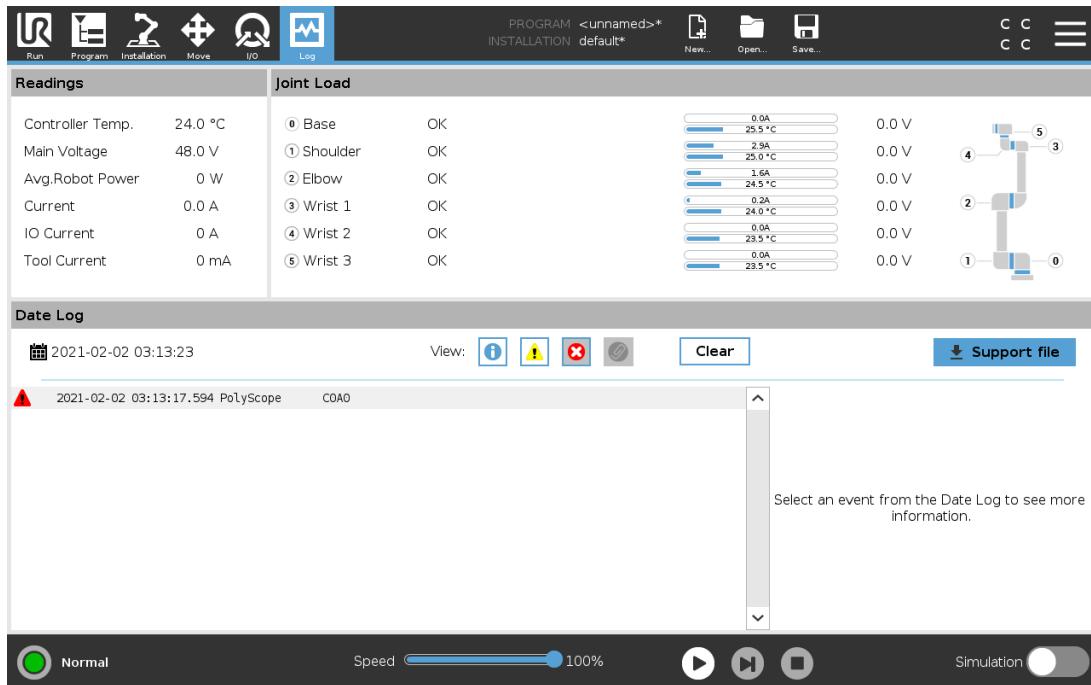
## 26.2. MODBUS

The screenshot below displays 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 (see ).





# 27. Log Tab



## 27.1. 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

## 27.2. 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.

## 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.

## 27.3. Saving Error Reports

A detailed status report is available when a paper clip icon appears on the log line.



### NOTE

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
- Protective Stop
- Unhandled exception in URCap
- Violation

The exported report contains: a user program, a history log, an installation and a list of running services.

## 27.4. 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.



### NOTE

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.



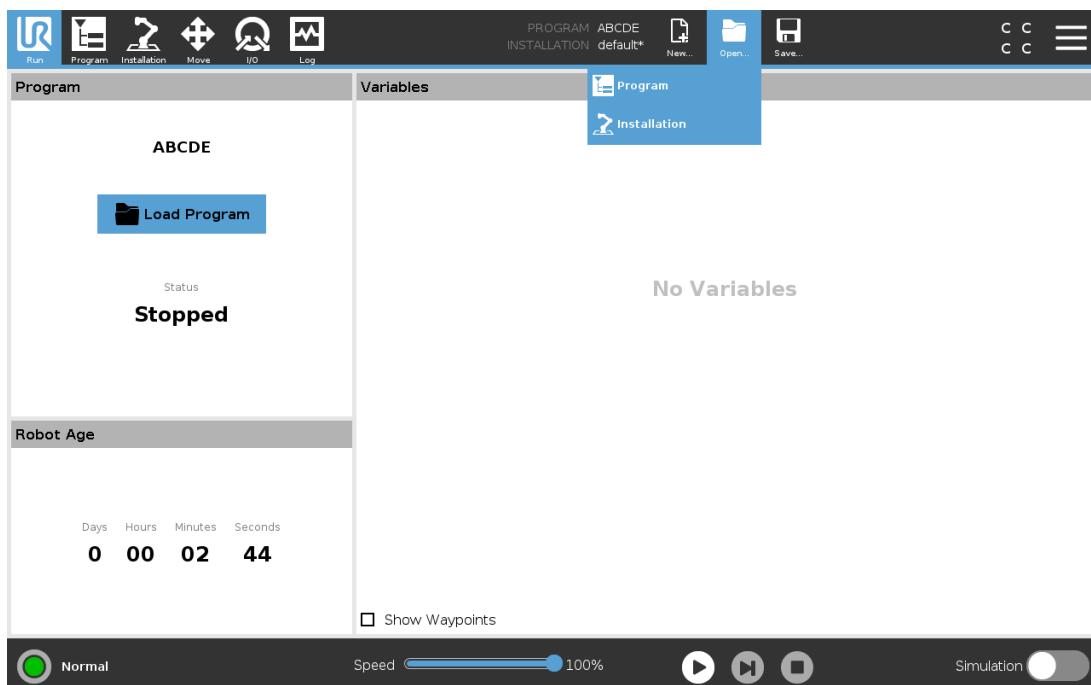
# 28. Program and Installation Manager



The Program and Installation Manager refers to three icons that allow you to create, load and configure Programs and Installations: **New...**, **Open...** and **Save....** 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.

## 28.1. Open...

Allows you to load a Program and/or Installation.



### Opening a Program

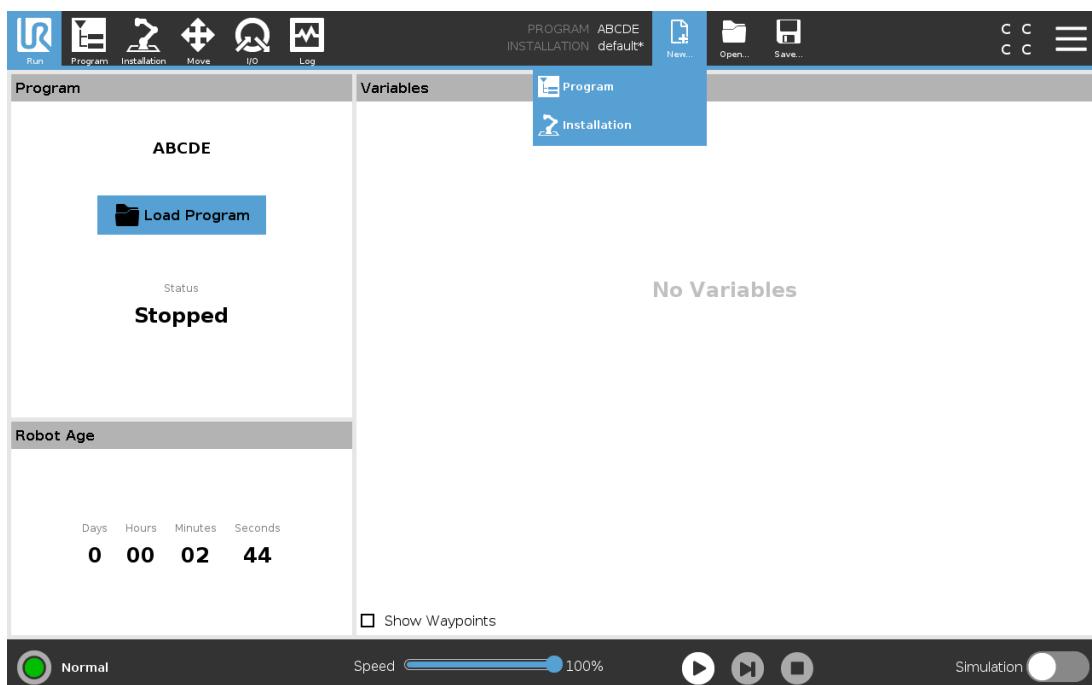
1. In the Program and Installation Manager, tap **Open...** and select Program.
2. On the Load Program screen, select an existing program and tap Open.
3. In the File Path, verify that the desired program name is displayed.

### Opening 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.

## 28.2. New...

Allows you to create a new Program and/or Installation.



### Creating 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.

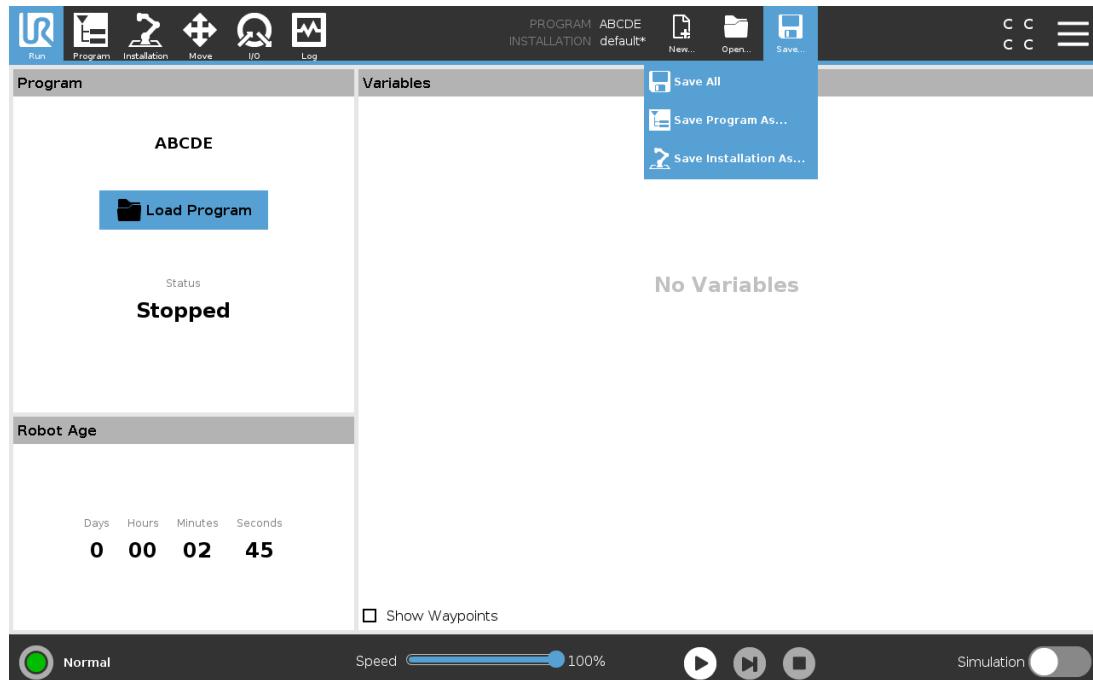
### Creating a new Installation

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.

## 28.3. Save...



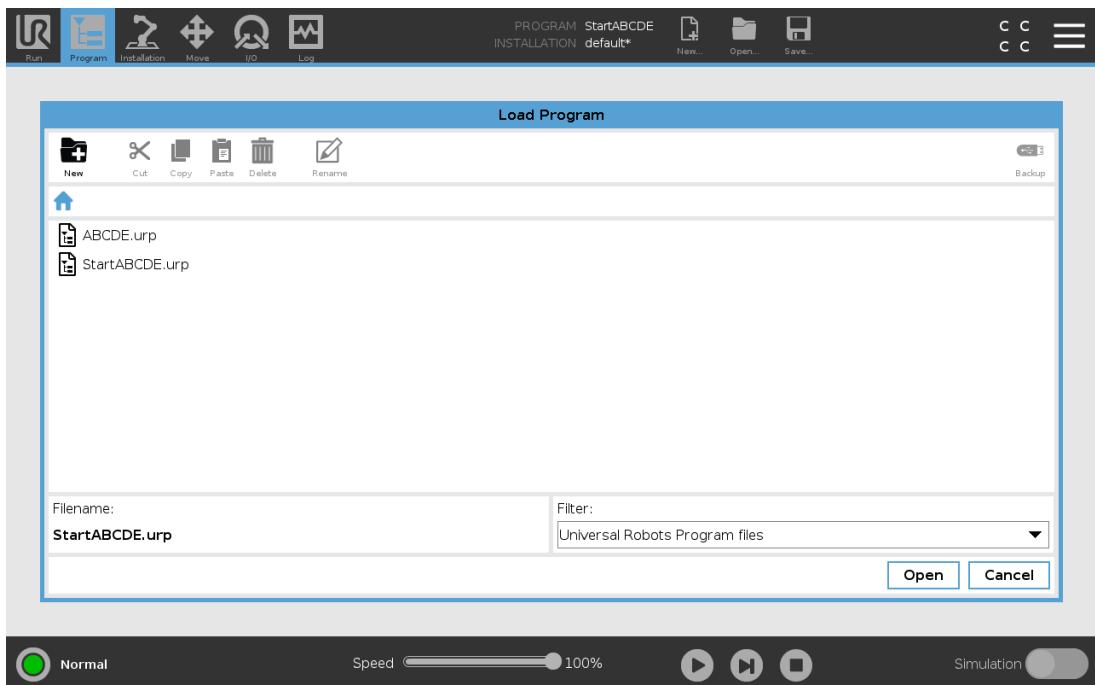
**Save...** proposes three options. Depending on the program/installation you load-create, you can:

**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.

## 28.4. File manager



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.

# 29. Hamburger menu

## 29.1. Help

You can find definitions for all elements that make up PolyScope capabilities.

1. In the right corner of the **Header**, tap the **Hamburger** menu and select **Help**.
2. Tap one of the red question marks that appears, to define desired element.
3. In the top right corner of element definition screen, tap the red X to exit Help.

## 29.2. About

You can display Version and Legal data.

1. Tap the **Hamburger** menu and select **About**.
2. Tap either **Version** or **Legal** to display data.
3. Tap **Close** to return to your screen.

## 29.3. Settings

### Personalizing 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.

#### 29.3.1. Preferences

##### Language

You can change PolyScope language and measurement unit (Metric or Imperial).

##### Time

You can access and/or adjust the current time and date displayed on the PolyScope.

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 [27.2. Date Log on page 239](#)) under **Date Log**.

## Hiding Speed Slider

Located at the base of the Run tab screen, the Speed Slider allows the operator to change the speed of a running Program.

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**.

### 29.3.2. Password

#### Admin

Use the Admin password to change the security configuration of the system, including network access.

The Admin password is equal to the password used for the root user account on the Linux system running on the robot, which may be needed in some network use cases such as SSH or SFTP.



#### WARNING

You cannot recover a lost Admin password.

- Take the appropriate steps to ensure your admin password is not lost.

#### Setting the Admin password

1. In the Header, tap the Hamburger menu icon and select **Settings**.
2. Under **Password**, tap Admin.
3. Under **Current password**, put in the default password: **easybot**.
4. Under **New password**, create a new password.  
Creating a strong, secret password obtains the best security for your system.
5. Under **Confirm new password**, repeat your new password.
6. Tap **Apply** to confirm your password change.

#### Safety

The Safety password prevents unauthorized modification of the Safety settings.

## 29.4. System

### 29.4.1. 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.



#### NOTE

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.

#### Backing up 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.

#### Restoring the System



#### NOTE

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 g restore error appears for not finding the matching serial number

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.

### 29.4.2. Update

Install updates from a USB to ensure the robot software is up-to-date.

## Updating 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.

### 29.4.3. Network

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

Tap **Apply** to apply changes.

### 29.4.4. Managing URCaps

You can manage your existing **URCaps** or install a new one in your robot.

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.

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.

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.

## 29.4.5. Remote Control

A robot can either be in Local Control (controlled from the Teach Pendant) or Remote Control (controlled externally).

 Local Control does not allow	 Remote Control does not allow
Power on and brake release sent to the robot over network	Moving the robot from Move Tab
Receiving and executing robot programs and installation sent to the robot over network	Starting from Teach Pendant
Autostart of programs at boot, controlled from digital inputs	Load programs and installations from the Teach Pendant
Auto brake release at boot, controlled from digital inputs	Freedrive
Start of programs, controlled from digital inputs	

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.

## Enabling 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.

**NOTE**

- 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.

## 29.4.6. Security

### General

General allows you to configure Magic Files and to configure network access to the robot.

#### Magic files

A Magic File is a script on a USB drive that executes once it is inserted into the system.

Magic files have unrestricted privileges to make system changes, thus they must be considered as a security liability.

##### Running magic files

1. In the Header, tap the Hamburger menu and select **Settings**.
2. Under Security, select **General**.
3. Enter your Admin password.
4. Enable **Run Magic Files**.

#### Configure 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 one host.

Leave the field blank to disable subnet restriction.

Use **Disable inbound access to additional interfaces (by port)** to make sure any inbound connection to the designated ports will be refused.

For example:

- Use 0-65535 to block all ports. Use 564 to block a specific port.
- Use 2318-3412,22,56-67 to block specific ports and specific ranges of ports.

Leave the field blank to avoid blocking ports. Any enabled service (refer to **Settings->Security->Services**) will take precedence over port blocking.



### 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.

#### Configure Inbound Connections

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**.
5. Enter the interfaces to be closed in **Disable inbound access to additional interfaces (by port)**.

## Secure Shell (SSH)

Secure Shell (SSH) provides a private (encrypted) and authenticated connection to the robot allowing:

- operating system access
- file copying
- tunneling of network interfaces



### NOTE

SSH is a powerful tool if used as intended. Make sure you understand how to use SSH technology before you rely on SSH technology to secure a robot application.

#### Enabling SSH access

1. In the Header, tap the Hamburger menu and select **Settings**.
2. Under Security, select **Secure Shell**.
3. Enter your Admin password.
4. 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.
5. Select the Authentication type.

## 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.

### Managing Authorized Keys

Key based authentication relies on pre-shared keys.

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.

## Services

Services lists the standard services running on the robot. You can enable or disable each service.

An enabled service remains available when 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.

The listed services must be disabled by selecting **General**, then **Inbound Connections** to prevent access.

### Enabling a Service

1. In the Header, tap the Hamburger menu and select **Settings**.
2. Under Security, select **Services**.
3. Enter your Admin password.
4. In the list, select an option and tap **Enable**, or tap **Disable**.

## 29.5. Shutdown Robot

The **Shutdown Robot** button allows the robot to be powered off or restarted.

### Shutting 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**.

# 30. Glossary

## *Stop Category 0*

Robot motion is stopped by immediate removal of power to the robot. It is an uncontrolled stop, where the robot can deviate from the programmed path as each joint brake as fast as possible. This protective stop is used if a safety-related limit is exceeded or in case of a fault in the safety-related parts of the control system. For more information, [see ISO 13850 or IEC 60204-1](#).

## *Stop Category 1*

Robot motion is stopped with power available to the robot to achieve the stop and then removal of power when the stop is achieved. It is a controlled stop, where the robot will continue along the programmed path. Power is removed as soon as the robot stands still. For more information, [see ISO 13850 or IEC 60204-1](#).

## *Stop Category 2*

A controlled stop with power left available to the robot. The safety-related control system monitors that the robot stays at the stop position. For more information, [see IEC 60204-1](#).

## *Category 3*

The term *Category* should not be confused with the term *Stop Category*. *Category* refers to the type of architecture used as basis for a certain *Performance Level*. A significant property of a *Category 3* architecture is that a single fault cannot lead to loss of the safety function. For more information, [see ISO 13849-1](#).

## *Performance Level*

A Performance Level (PL) is a discrete level used to specify the ability of safety-related parts of control systems to perform a safety functions under foreseeable conditions. PLd is the second highest reliability classification, meaning that the safety function is extremely reliable. For more information, [see ISO 13849-1](#).

## *Diagnostic coverage (DC)*

is a measure of the effectiveness of the diagnostics implemented to achieve the rated performance level. For more information, [see ISO 13849-1](#).

## *MTTFd*

The Mean time to dangerous failure (MTTFd) is a value based on calculations and tests used to achieve the rated performance level. For more information, [see ISO 13849-1](#).

## *Integrator*

The integrator is the entity that designs the final robot installation. The integrator is responsible for making the final risk assessment and must ensure that the final installation complies with local laws and regulations.

## *Risk assessment*

A risk assessment is the overall process of identifying all risks and reducing them to an appropriate level. A risk assessment should be documented. Consult [ISO 12100](#) for further information.

## *Collaborative robot application*

The term *collaborative* refers to collaboration between operator and robot in a robot application. See precise definitions and descriptions in [ISO 10218-1](#) and [ISO 10218-2](#).

*Safety configuration*

Safety-related functions and interfaces are configurable through safety configuration parameters. These are defined through the software interface, see part [Part II PolyScope Manual on page 93](#).

## 30.1. Index

### A

- About** 247
- Align** 230
- Auto** 230
- Automatic mode** 111
- Automatic Mode Safeguard Reset** 129
- Automatic Mode Safeguard Stop** 129

### B

- Base** 62, 95, 148
- Base feature** 215
- Blend parameters** 151
- Blending** 150
- Bracket** 32

### C

- Cone angle** 127
- Cone center** 127
- Configurable I/O** 35
- control box** 74, 99, 235
- Control Box** 2, 30, 32, 34, 43-44, 206
- Conveyor Tracking** 35, 183
- Conveyor Tracking Setup** 211
- Custom** 118

**D**

- 
- Delete** 122
  - Direction Vector** 156
  - Disabled** 121, 123
  - Disabled Tool direction limit** 127

**E**

- 
- Edit Position** 125
  - Elbow** 62, 95
  - Elbow Force** 119
  - Elbow Speed** 119
  - Error** 185
  - Ethernet** 32, 224
  - EtherNet/IP** 32, 203, 227
  - Expression Editor** 170

**F**

- 
- Factory Presets** 117
  - Feature** 212, 215, 230
  - Feature menu** 181
  - File Path** 243
  - Folder** 162
  - Footer** 95, 134
  - Frame** 181
  - Freedrive** 22, 111, 182, 202, 216, 230-231

**G**

- 
- General purpose I/O** 35

## H

**Hamburger Menu** 98

**Header** 95

**Home** 230

## I

**I/O** 32, 36, 97, 128, 203, 205, 235

**Initialize** 98-99

**input signals** 128

**Installation** 97, 243-244

**Installation variables** 206

**integrator** 13

## J

**Joint Limits** 120

**joint space** 146

## L

**Log** 97, 239

## M

**Manual High Speed** 98, 113

**Manual mode** 111

**Mini Displayport** 32

**MODBUS** 32, 211, 224, 226, 236

**mode**

Automatic 97, 112

Local 97  
Manual 97, 112  
Remote 97  
**Modes** 22, 121  
**Momentum** 118  
**Motion** 181  
**Mounting bracket** 2  
**Move** 97, 111, 146-147, 160, 232  
**Move robot to** 135  
**Move Tool** 229  
**MoveJ** 146, 216, 233  
**MoveL** 146, 216, 232  
**MoveP** 146, 216

## N

**New...** 97, 243  
**Normal** 122  
**Normal & Reduced** 122  
**Normal & Reduced Tool direction limit** 127  
**Normal mode** 119, 127, 144, 232  
**Normal Plane** 124  
**Normal Tool direction limit** 127  
**Not Reduced mode** 130

## O

**Open...** 97, 243  
**output signals** 129

## P

**Pan angle** 128

Play 98, 134  
Point 181  
PolyScope 1, 22, 95, 98, 101, 131, 133, 164, 197, 224, 228, 247, 252  
popup 161  
Pose Editor 230-231  
Position 125  
Position range 120  
Power 118  
Profinet 228  
Program 97, 133, 183, 243-244  
Program and Installation Manager 97, 243  
program node 137  
Program Node 142  
Program Tree 137

## R

Radius 125  
Recovery mode 22, 119  
Reduced 122  
Reduced mode 22, 119, 125, 127-129  
Reduced Tool direction limit 127  
Relative waypoint 148  
Remote Control 207, 251  
Rename 122  
Restrict Elbow 123  
risk assessment 2, 7, 13, 15, 19  
Robot 125, 229  
robot arm 95, 99, 178, 182, 206, 229  
Robot arm 74  
Robot Arm 32  
robot cable 46-47  
Robot Limits 117  
Robot Moving 129

**Robot Not Stopping** 129  
**Robot Program Node** 142  
**Run** 96, 133

## S

**safe Home** 130

**Safeguard Reset** 129

**Safety Checksum** 97, 117

**Safety Configuration** 13, 115-117, 120

**Safety functions** 17-18

**Safety I/O** 17, 21, 35-37

**Safety instructions** 53

**Safety planes** 121, 229, 232

**Safety Settings** 7, 115, 248

**Save...** 97, 243, 245

**Screen** 95

**Script manual** 3

**Service manual** 3

**Settings** 247

**Setup** 231

**Shoulder** 62, 95

**Show** 122

**Shut Down** 254

**Simple** 181

**Simulation** 98

**Speed Slider** 98, 111

**standard** 74

**Standard** 76

**Step** 98

**Stop** 98

**stopped state** 99

**Stopping Distance** 118

**Stopping Time** 118

Success 185  
Switch Case construction 170  
System Emergency stop 128  
System Emergency Stop 129

## T

TCI 159  
Teach Pendant 2, 30, 43, 95, 99, 130, 182, 251  
Templates 183  
Test button 182  
Tilt angle 128  
Tool 125  
Tool Center Point 119, 148, 197, 230  
Tool Center Position 125  
Tool Communication Interface 209  
Tool Direction 126-127  
Tool feature 215  
Tool Flange 95  
Tool Force 118  
Tool I/O 47  
Tool Position 125-126  
Tool Speed 118  
Trigger Plane 124  
Trigger Reduced Mode 122

## U

Until 156  
Until Distance 157  
Until Expression 157  
Until Tool Contact 157  
UR+ 3  
URCaps 250

**V**

**Variable feature** 148

**Variable waypoint** 148

**Variables** 133, 145

**Voltage** 235

**W**

**Wait** 159

**Warning signs** 8

**Warranty** 61

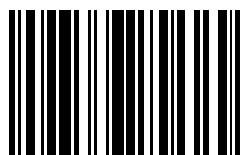
**Waypoint** 146, 148-150, 155

**Waypoints** 101

**Wrist** 95

Software Version: 5.11

Document Version: 9.4.97



99404