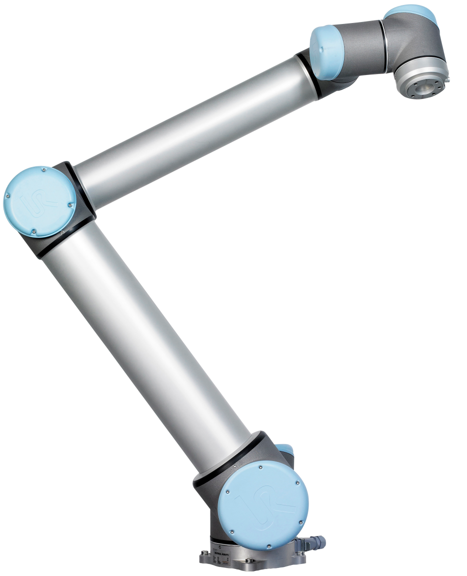
UNIVERSAL ROBOTS

User Manual



UR10/CB3

Original instructions (en)

UNIVERSAL ROBOTS

User Manual

UR10/CB3

Version 3.1 (rev. 17782)

Original instructions (en)

Serial number UR10: ______________________

Serial number CB3: ______________________

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 manual is periodically reviewed and revised.

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

Copyright © 2009-2015 by Universal Robots A/S

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

Contents

Preface ix
What do the Boxes Contain . . . ix
Important Safety Notice . . . x
How to Read This Manual . . . x
Where to Find More Information . . . x

I Hardware Installation Manual I-1

1 Safety I-3
1.1 Introduction . . . I-3
1.2 Validity and Responsibility . . . I-3
1.3 Limitation of Liability . . . I-4
1.4 Warning Symbols in this Manual . . . I-4
1.5 General Warnings and Cautions . . . I-5
1.6 Intended Use . . . I-7
1.7 Risk Assessment . . . I-7
1.8 Emergency Stop . . . I-8
1.9 Movement Without Drive Power . . . I-8

2 Transportation I-11

3 Mechanical Interface I-13
3.1 Workspace of the Robot . . . I-13
3.2 Mounting . . . I-13

4 Electrical Interface I-17
4.1 Introduction . . . I-17
4.2 Electrical warnings and cautions . . . I-17
4.3 Controller I/O . . . I-19
4.3.1 Common specifications for all digital I/O . . . I-19
4.3.2 Safety I/O . . . I-20
4.3.3 General purpose digital I/O . . . I-24
4.3.4 Digital input from a button . . . I-24
4.3.5 Communication with other machines or PLCs . . . I-25
4.3.6 General purpose analog I/O . . . I-25
4.3.7 Remote ON/OFF control . . . I-27
4.4 Tool I/O . . . I-28
4.4.1 Tool Digital Outputs . . . I-29
4.4.2 Tool Digital Inputs . . . I-30
4.4.3 Tool Analog Inputs . . . I-30

4.5 Ethernet I-31
4.6 Mains connection I-32
4.7 Robot connection I-33

5 Safety-related Functions and Interfaces **I-35**
5.1 Limiting Safety-related Functions I-35
5.2 Safety Modes I-37
5.3 Safety-related Electrical Interfaces I-37
5.3.1 Safety-related Electrical Inputs I-37
5.3.2 Safety-related Electrical Outputs I-39

6 Maintenance and Repair **I-41**
6.1 Safety Instructions I-41

7 Disposal and Environment **I-43**

8 Certifications **I-45**
8.1 Third Party Certifications I-45
8.2 Declarations According to EU directives I-45

9 Warranties **I-47**
9.1 Product Warranty I-47
9.2 Disclaimer I-47

A Stopping Time and Stopping Distance **I-49**
A.1 CATEGORY 0 stopping distances and times I-49

B Declarations and Certificates **I-51**
B.1 CE Declaration of Incorporation (original) I-51
B.2 Safety System Certificate I-52
B.3 Environmental Test Certificate I-53
B.4 EMC Test Certificate I-54

C Applied Standards **I-55**

D Technical Specifications **I-61**

II PolyScope Manual **II-1**

10 Introduction **II-3**
10.1 Getting Started II-3
10.1.1 Installing the Robot Arm and Control Box II-3
10.1.2 Turning the Control Box On and Off II-4
10.1.3 Turning the Robot Arm On and Off II-4
10.1.4 Quick Start II-4
10.1.5 The First Program II-5
10.2 PolyScope Programming Interface II-6
10.3 Welcome Screen II-8
10.4 Initialization Screen II-9

11 On-screen Editors **II-11**
11.1 On-screen Keypad . . . II-11
11.2 On-screen Keyboard . . . II-12
11.3 On-screen Expression Editor . . . II-12
11.4 Pose Editor Screen . . . II-13

12 Robot Control **II-17**
12.1 Move Tab . . . II-17
  12.1.1 Robot . . . II-17
  12.1.2 Feature and Tool Position . . . II-18
  12.1.3 Move Tool . . . II-18
  12.1.4 Move Joints . . . II-18
  12.1.5 Freedrive . . . II-18
12.2 I/O Tab . . . II-19
12.3 MODBUS client I/O . . . II-20
12.4 AutoMove Tab . . . II-20
12.5 Installation → Load/Save . . . II-22
12.6 Installation → TCP Configuration . . . II-23
  12.6.1 Adding, modifying and removing TCPs . . . II-23
  12.6.2 The default and the active TCP . . . II-23
  12.6.3 Teaching TCP position . . . II-24
  12.6.4 Teaching TCP orientation . . . II-25
  12.6.5 Payload . . . II-25
  12.6.6 Center of gravity . . . II-25
12.7 Installation → Mounting . . . II-26
12.8 Installation → I/O Setup . . . II-27
12.9 Installation → Safety . . . II-28
12.10 Installation → Variables . . . II-28
12.11 Installation → MODBUS client I/O Setup . . . II-29
12.12 Installation → Features . . . II-32
12.13 Conveyor Tracking Setup . . . II-35
12.14 Installation → Default Program . . . II-36
  12.14.1 Loading a Default Program . . . II-37
  12.14.2 Starting a Default Program . . . II-37
  12.14.3 Auto Initialization . . . II-37
12.15 Log Tab . . . II-38
12.16 Load Screen . . . II-38
12.17 Run Tab . . . II-41

13 Programming **II-43**
13.1 New Program . . . II-43
13.2 Program Tab . . . II-44
  13.2.1 Program Tree . . . II-44
  13.2.2 Program Execution Indication . . . II-45
  13.2.3 Undo/Redo Buttons . . . II-45
  13.2.4 Program Dashboard . . . II-45

13.3 Variables . . . II-46
13.4 Command: Empty . . . II-47
13.5 Command: Move . . . II-48
13.6 Command: Fixed Waypoint . . . II-51
13.7 Command: Relative Waypoint. . . . II-53
13.8 Command: Variable Waypoint . . . II-54
13.9 Command: Wait . . . II-55
13.10 Command: Set . . . II-55
13.11 Command: Popup . . . II-56
13.12 Command: Halt . . . II-57
13.13 Command: Comment . . . II-57
13.14 Command: Folder . . . II-58
13.15 Command: Loop . . . II-58
13.16 Command: SubProgram . . . II-59
13.17 Command: Assignment . . . II-60
13.18 Command: If . . . II-61
13.19 Command: Script . . . II-62
13.20 Command: Event . . . II-63
13.21 Command: Thread . . . II-64
13.22 Command: Pattern . . . II-64
13.23 Command: Force . . . II-66
13.24 Command: Pallet . . . II-69
13.25 Command: Seek . . . II-70
13.26 Command: Start/Stop Conveyor Tracking . . . II-73
13.27 Command: Suppress . . . II-74
13.28 Graphics Tab . . . II-74
13.29 Structure Tab . . . II-75
13.30 Variables Tab . . . II-76
13.31 Command: Variables Initialization . . . II-77

14 Setup Screen **II-79**
14.1 Language and Units . . . II-80
14.2 Update Robot. . . . II-81
14.3 Set Password . . . II-82
14.4 Calibrate Screen . . . II-83
14.5 Setup Network . . . II-83
14.6 Set Time. . . . II-84

15 Safety Configuration **II-85**
15.1 Changing the Safety Configuration . . . II-86
15.2 Safety Synchronization and Errors . . . II-86
15.3 Tolerances . . . II-87
15.4 Safety Checksum . . . II-88
15.5 Safety Modes . . . II-88
15.6 Freedrive Mode . . . II-89
15.7 Password Lock . . . II-89
15.8 Apply . . . II-89

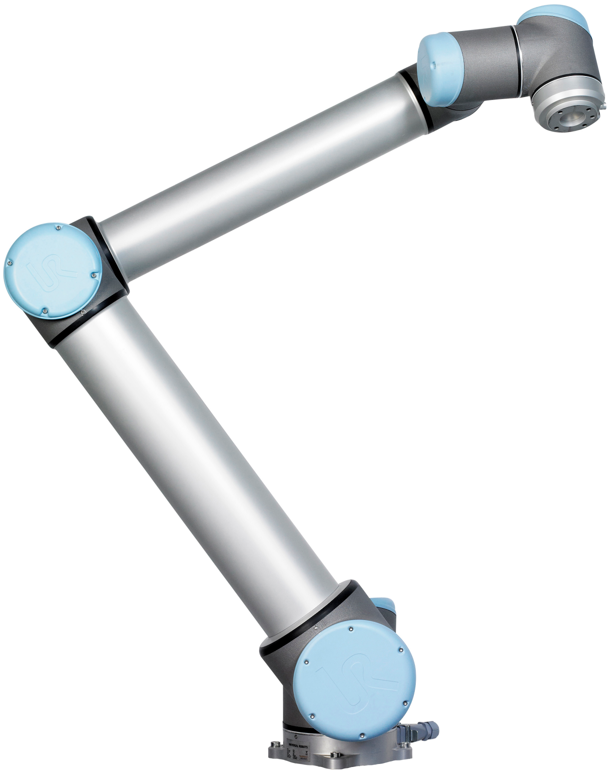
15.9 General Limits II-90
15.10 Joint Limits II-93
15.11 Boundaries. II-94
    15.11.1 Selecting a boundary to configure II-95
    15.11.2 3D visualization II-95
    15.11.3 Safety plane configuration II-96
    15.11.4 Tool Boundary configuration II-99
15.12 Safety I/O II-101

Glossary **II-103**

Index **II-105**

Preface

Congratulations on the purchase of your new Universal Robot, UR10.



The robot can be programmed to move a tool, and communicate with other machines using electrical signals. It is an arm composed of extruded aluminum tubes and joints. Using our patented programming interface, PolyScope, it is easy to program the robot to move the tool along a desired trajectory.

What do the Boxes Contain

When you order a complete robot, you receive two boxes. One contains the the robot arm and the following items are included in the other one:

- 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;
- Mains cable compatible with your region;
- Tool cable;
- Stylus pen with laser;

- UR production test certificate;
- This manual.

Important Safety Notice

The robot is *partly completed machinery* (see 8.2) and as such a risk assessment is required for each installation of the robot. It is particularly important that all of the safety instructions in chapter 1 are followed.

How to Read This Manual

This manual contains instructions for installing and using the robot. It consists of the following parts:

Hardware Installation Manual: The mechanical and electrical installation of the robot.

PolyScope Manual: Programming of the robot.

This manual is intended for the integrator who is expected to have a basic level of mechanical and electrical training. It is also helpful, though not necessary, to be familiar with elementary concepts of programming. No special knowledge about robots in general or Universal Robots in particular is required.

Where to Find More Information

The support website (`http://support.universal-robots.com/`), available to all UR distributors, contains additional information, such as:

- Other language versions of this manual;
- PolyScope Manual updates after the PolyScope is upgraded to a new version.
- The *Service Manual* with instructions for troubleshooting, maintenance and repair of the robot.
- The *Script Manual* for advanced users.

Part I

Hardware Installation Manual

1 Safety

1.1 Introduction

This chapter contains important safety information, which must be read and understood by the integrator of UR robots.

The first subsections in this chapter are more general and the later subsections contain more specific engineering data relevant for setting up and programming the robot.

It is essential that all assembly instructions and guidance provided in other chapters and parts of this manual are observed and followed.

Special attention shall be paid to text associated with warning symbols. See Chapter 5 for detailed descriptions of the safety-related functions and interfaces.

1.2 Validity and Responsibility

The information does not cover how to design, install and operate 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 UR 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:

- Making a risk assessment for the complete 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 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.

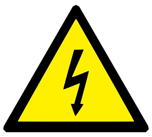
Guidance on how to find and read applicable standards and laws is provided on `http://support.universal-robots.com/`

1.3 Limitation of Liability

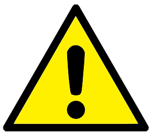
Any information given in this manual regarding safety must not be construed as a warranty by UR that the industrial manipulator will not cause injury or damage even if all safety instructions are complied with.

1.4 Warning Symbols in this Manual

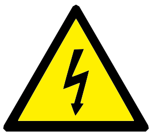
The table below defines the captions specifying the danger levels used throughout this manual. The same warning signs are used on the product.



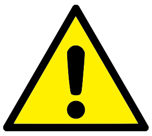
DANGER:
This indicates an imminently hazardous electrical situation which, if not avoided, could result in death or serious injury.



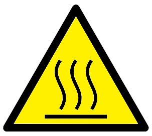
DANGER:
This indicates an imminently hazardous situation which, if not avoided, could result in death or serious injury.



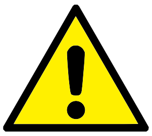
WARNING:
This indicates a potentially hazardous electrical situation which, if not avoided, could result in injury or major damage to the equipment.



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.

1.5 General Warnings and Cautions

This section contains some general warnings and cautions. Some of which are repeated or explained in different parts of the manual. Other warnings and cautions are present throughout the manual.

DANGER:
Make sure to install the robot and all electrical equipment according to the specifications and warnings found in the Chapters 3 and 4.

WARNING:

1. Make sure the robot arm and tool 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.
6. If the software prompts a fatal error, immediately activate 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 normal I/O. Use safety-related interfaces only.
8. Make sure to use the correct installation settings (e.g. Robot mounting angle, weight in TCP, TCP offset, safety configuration). Save and load the installations file along with the program.
9. The freedrive function (Impedance/back-drive) shall only be used in installations where the risk assessment allows it. Tools and obstacles shall not have sharp edges or pinch points. Make sure that all people keep their heads and faces outside the reach of the robot.
10. Be aware of robot movement when using the teach pendant.
11. Do not enter the safety range of the robot or touch the robot when the system is in operation.

11. Collisions can release high portions of kinetic energy, which are significantly higher at high speeds and with high payloads. (Kinetic Energy = $\frac{1}{2}\text{Mass} \cdot \text{Speed}^2$)
12. Combining different machines might increase hazards or create new hazards. Always make an overall risk assessment for the complete installation. 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.
13. 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. UNIVERSAL ROBOTS DISCLAIMS ANY LIABILITY IF THE PRODUCT IS CHANGED OR MODIFIED IN ANY WAY.
14. If the robot is purchased with an extra module (e.g. euromap67 interface) then look up that module in the respective manual.

WARNING:

1. The robot and controller box generate heat during operation. Do not handle or touch the robot while in operation or immediately after operation. To cool the robot down, power off the robot and wait one hour.
2. Never stick fingers behind the internal cover of the controller box.

CAUTION:

1. When the robot is combined with or working with machines capable of damaging the robot, then it is highly recommended to test all functions and the robot program separately. It is recommended to test the robot program using temporary waypoints outside the workspace of other machines. Universal Robots cannot be held responsible for any damages caused to the robot or to other equipment due to programming errors or malfunctioning of the robot.
2. Do not expose the robot to permanent magnetic fields. Very strong magnetic fields can damage the robot.

1.6 Intended Use

UR robots are industrial and intended for handling tools and fixtures, or for processing or transferring components or products. For details about the environmental conditions under which the robot should operate, see appendices B and D.

UR robots are equipped with special safety-related features, which are purposely designed for collaborative operation, where the robot operates without fences and/or together with a human.

Collaborative operation is only intended for non-hazardous applications, where the complete application, including tool, 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 the 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 before performing a risk assessment;
- Use where the rated performance levels are insufficient;
- Use where the reaction times of the safety functions are insufficient;
- Use as a climbing aid;
- Operation outside the permissible operating parameters.

1.7 Risk Assessment

One of the most important things that an integrator needs to do is to make a risk assessment. The robot itself is partly completed machinery, as the safety of the robot installation depends on how the robot is integrated (E.g. tool, obstacles and other machines).

It is recommended that the integrator uses guidelines in ISO 12100 and ISO 10218-2 to conduct the risk assessment.

The risk assessment shall consider two scenarios:

- Teaching the robot while developing the robot installation;
- Normal operation of the robot installation.

If the robot is installed in a non-collaborative installation (E.g. when using a hazardous tool) the risk assessment might conclude that the integrator needs to connect additional safety devices (E.g. an enable device) to protect him while programming.

Universal Robots has identified the potential significant hazards listed below as hazards which must be considered by the integrator. Note that other significant hazards might be present in a specific robot installation.

1. Entrapment of fingers between robot foot and base (joint 0).
2. Entrapment of fingers between wrist 1 and wrist 2 (joint 3 and joint 4).
3. Penetration of skin by sharp edges and sharp points on tool or tool connector.
4. Penetration of skin by sharp edges and sharp points on obstacles near the robot track.
5. Bruising due to stroke from the robot.
6. Sprain or bone fracture due to strokes between a heavy payload and a hard surface.
7. Consequences due to loose bolts that hold the robot arm or tool.
8. Items falling out of tool, e.g. due to a poor grip or power interruption.
9. Mistakes due to different emergency stop buttons for different machines.

Information on stopping times and stopping distances are found in appendix A.

1.8 Emergency Stop

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

Emergency stop shall not be used as a risk reduction measure, but as a secondary protective device.

The risk assessment of the robot application shall conclude if more emergency stop buttons must be connected. Emergency stop buttons should comply with IEC 60947-5-5, see more in section 4.3.2.

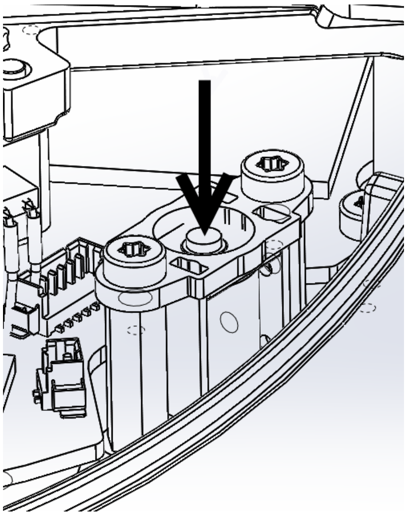
1.9 Movement Without Drive Power

In the unlikely event of an emergency situation where one or more robot joints need to be moved and robot power is either not possible or unwanted, there are two different ways to force movements of the robot joints:

1. Forced back-driving: Force a joint to move by pushing or pulling the robot arm hard (500 N). Each joint brake has a friction clutch which enables movement during high forced torque.
2. Manual brake release: Remove the joint cover by removing the few M3 screws that fix it. Release the brake by pushing the plunger on the small electromagnet as shown in the picture below.

WARNING:

1. Moving the robot arm manually is intended for urgent emergencies only and might damage the joints.
2. If the brake is released manually, gravitational pull can cause the robot arm to fall. Always support the robot arm, tool and work item when releasing the brake.



2 Transportation

Transport the robot in the original packaging. Save the packaging material in a dry place; you may need to pack down and move the robot later on.

Lift both tubes of the robot arm at the same time when moving it from the packaging to the installation place. Hold the robot in place until all mounting bolts are securely tightened at the base of the robot.

The controller box shall be lifted by the handle.

WARNING:

1. Make sure not to overload your back or other bodyparts when the equipment is lifted. Use proper lifting equipment. All regional and national guidelines for lifting 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 mounting instructions in chapter 3.

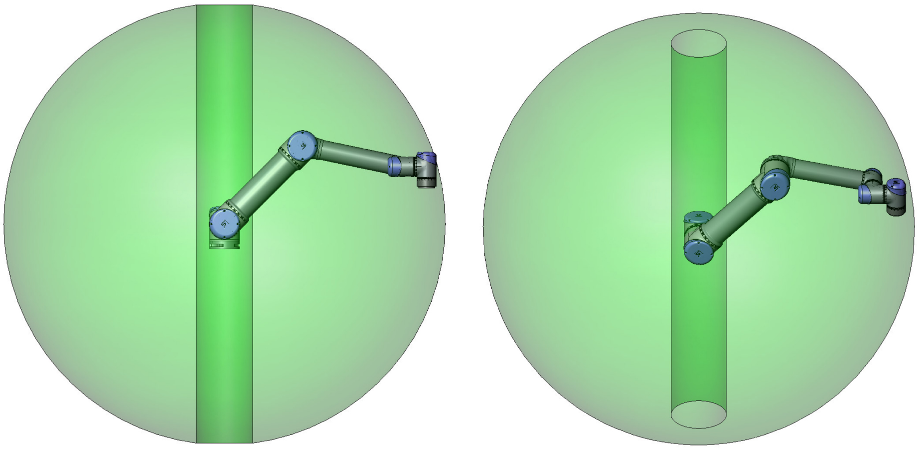
3 Mechanical Interface

The robot consists essentially of six robot joints and two aluminum tubes, connecting the *base* with the tool of the robot. The robot permits the tool to be translated and rotated within the workspace. The next section describes the basics of mounting the various parts of the robot system.

Electrical installation instructions in chapter 4 must be observed.

3.1 Workspace of the Robot

The workspace of the UR10 robot extends 1300 mm from the base joint. It is important to consider the cylindrical volume directly above and directly below the robot base when a mounting place for the robot is chosen. Moving the tool close to the cylindrical volume should be avoided if possible, because it causes the joints to move fast even though the tool is moving slowly, causing the robot to work inefficiently and the conduction of the risk assessment to be difficult.



Front Tilted

3.2 Mounting

Robot Arm The robot arm is mounted using four M8 bolts, using the four 8.5 mm holes on the base. It is recommended to tighten these bolts with 20 N m torque. If very accurate repositioning of the robot arm is desired, two Ø8 holes are provided for use with a pin. Also, an accurate base counterpart can be purchased as an accessory. Figure 3.1 shows where to drill holes and mount the screws.

Mount the robot on a sturdy surface strong enough to withstand at least ten times the full torque of the base joint and at least five times the weight of the robot arm. Furthermore the surface shall be vibration free.

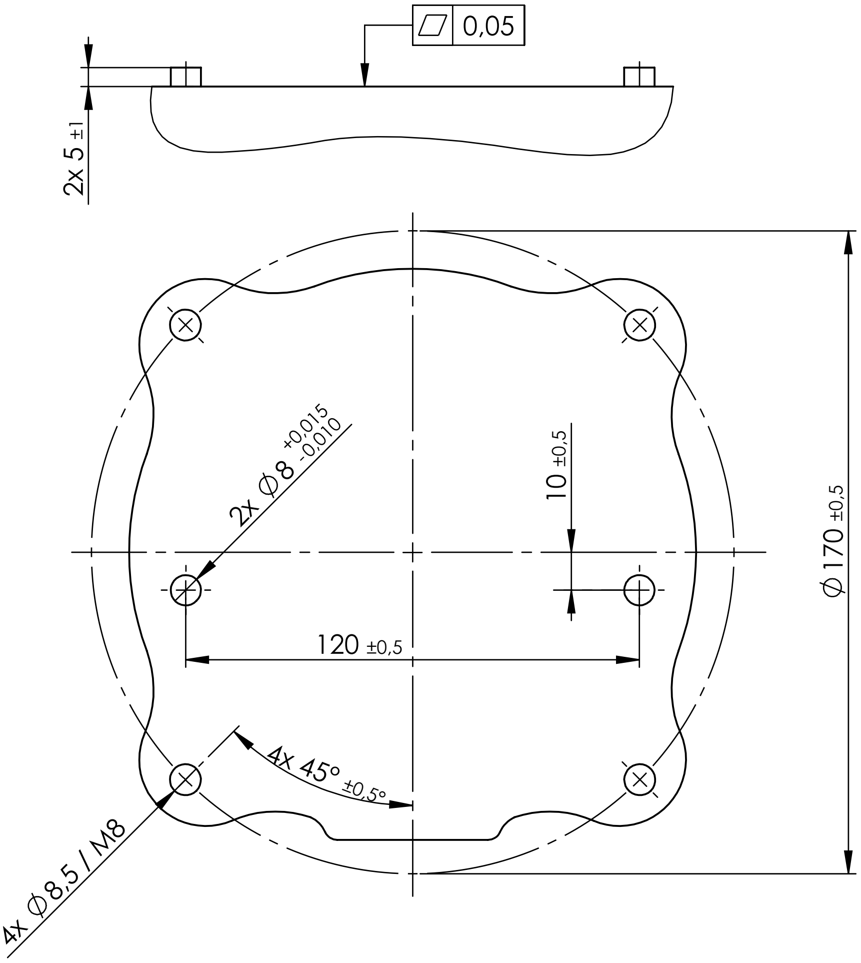


Figure 3.1: Holes for mounting the robot. Use four M8 bolts. All measurements are in mm.

If the robot is mounted on a linear axis or a moving platform then the acceleration of the moving mounting base shall be very low. A high acceleration might cause the robot to stop, thinking it bumped into something.

DANGER:
Make sure the robot arm is properly and securely bolted in place. The mounting surface shall be sturdy.

CAUTION:
If the robot is bathed in water over an extended time period it might be damaged. The robot should not be mounted in water or in a wet environment.

Tool The robot tool flange has four M6 thread holes for attaching a tool to the robot. The holes need to be tightened with 9 N m. If very accurate repositioning of the tool is desired, the Ø6 hole is provided for use with a pin. Figure 3.2 shows where to drill holes and mount the screws.

DANGER:

1. Make sure the tool is properly and securely bolted in place.
2. Make sure that the tool is constructed such that it cannot create a hazardous situation by dropping a part unexpectedly.

Control Box The control box can be hung on a wall, or it can be placed on the ground. A clearance of 50 mm on each side is needed for sufficient airflow. Extra brackets for mounting can be bought.

Teach Pendant The teach pendant can be hung on a wall or on the control box. Extra brackets for mounting the teach pendant can be bought. Make sure that no one can trip over the cable.

DANGER:

1. Make sure that the control box, teach pendant, and cables do not come into contact with liquids. A wet control box could cause death.
2. The control box and teach pendant must not be exposed to dusty or wet environments that exceed IP20 rating. Pay special attention to environments with conductive dust.

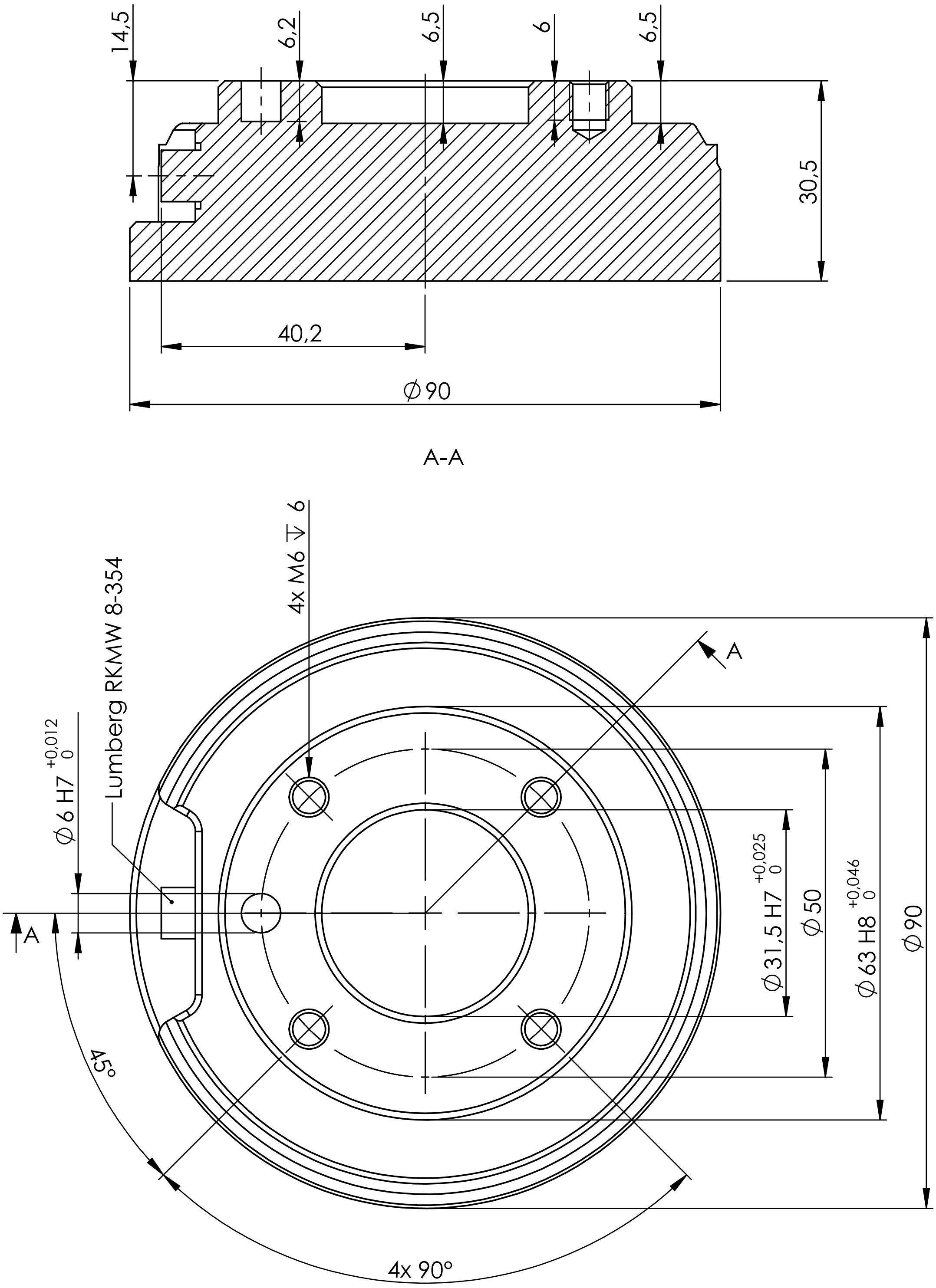

14,5
6,2
6,5
6
6,5
30,5
40,2
Ø90
A-A
4x M6 ↧ 6
Lumberg RKMW 8-354
Ø6 H7 +0,012 0
A
A
45°
4x 90°
Ø31,5 H7 +0,025 0
Ø50
Ø63 H8 +0,046 0
Ø90

Figure 3.2: The tool output flange, ISO 9409-1-50-4-M6. This is where the tool is mounted at the tip of the robot. All measures are in mm.

4 Electrical Interface

4.1 Introduction

This chapter describes all the electrical interfaces of the robot arm and control box.

The different interfaces are divided into five groups with different purposes and properties:

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

The term "I/O" refers both digital and analog control signals going from or to an interface.

These five groups are described in the following sections. Examples are given for most types of I/O.

The warnings and cautions in the following section are relevant for all five groups and must be observed.

4.2 Electrical warnings and cautions

The following warnings and cautions must be observed when a robot application is designed and installed. The warnings and cautions also apply for service work.

DANGER:

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 one of safety stop 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/O inside the control box can be configured for either normal or safety-related I/O. Read and understand the complete section 4.3.

DANGER:

1. Make sure that all equipment not rated for water exposure remains dry. If water comes inside the product, lockout and tagout all power and then contact your supplier.
2. Use original cables supplied with the robot only. Do not use the robot for applications where the cables will be subjected to flexing. Contact your supplier if longer or flexible cables are needed.
3. Minus connections are referred to as “GND” and are connected to the shield of the robot and the controller box. All mentioned GND connections are only for powering and signalling. For PE (Protective Earth) use the M6 sized 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. Care must be taken 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 the holes. Make sure that all shavings are removed before reinstalling the plate. Remember to use correct gland sizes.

CAUTION:

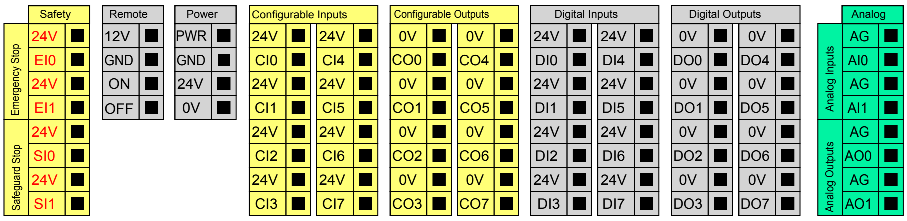
1. The robot has been tested according to international IEC standards for EMC (ElectroMagnetic Compatibility). Disturbing signals with levels higher than those defined in the specific IEC standards can cause unexpected behavior of the robot. Very high signal levels or excessive exposure can damage the 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 extended tests are performed.

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

4.3 Controller I/O

This chapter explains how to connect equipment to I/O inside the control box. This I/O is extremely flexible and can be used for wide range of different equipment; including pneumatic relays, PLCs and emergency stop buttons.

The illustration below shows the layout of electrical interface inside the control box.


Safety
Emergency Stop
Safeguard Stop
24V
EI0
24V
EI1
24V
SI0
24V
SI1
Remote
12V
GND
ON
OFF
Power
PWR
GND
24V
0V
Configurable Inputs
24V
CI0
24V
CI1
24V
CI2
24V
CI3
24V
CI4
24V
CI5
24V
CI6
24V
CI7
Configurable Outputs
0V
CO0
0V
CO1
0V
CO2
0V
CO3
0V
CO4
0V
CO5
0V
CO6
0V
CO7
Digital Inputs
24V
DI0
24V
DI1
24V
DI2
24V
DI3
24V
DI4
24V
DI5
24V
DI6
24V
DI7
Digital Outputs
0V
DO0
0V
DO1
0V
DO2
0V
DO3
0V
DO4
0V
DO5
0V
DO6
0V
DO7
Analog
Analog Inputs
Analog Outputs
AG
AI0
AG
AI1
AG
AO0
AG
AO1

The meaning of the different colors must be observed, see below.

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

The "configurable" I/O can be configured as either safety-related I/O or general purpose I/O in the GUI. See more in part II.

How to use the digital I/O is described in the following subsections. The section describing the common specifications must be observed.

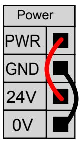
4.3.1 Common specifications for all digital I/O

This section define electrical specifications for the following 24V digital I/O of the control box.

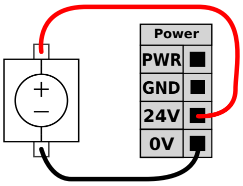
- Safety I/O.
- Configurable I/O.
- General purpose I/O.

It is very important that UR robots are installed according the electrical specifications, which are the same for all three different kinds of 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 is to use the internal power supply, see below.


Power
PWR
GND
24V
0V

If more current is needed, an external power supply can be connected as shown below.



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

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

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

Terminals	Parameter	Min	Typ	Max	Unit
Digital outputs					
`[COx / DOx]`	Current	0	-	1	A
`[COx / DOx]`	Voltage drop	0	-	0.5	V
`[COx / DOx]`	Leakage current	0	-	0.1	mA
`[COx / DOx]`	Function	-	PNP	-	Type
`[COx / DOx]`	IEC 61131-2	-	1A	-	Type
Digital Inputs					
`[EIx/SIx/CIx/DIx]`	Voltage	-3	-	30	V
`[EIx/SIx/CIx/DIx]`	OFF region	-3	-	5	V
`[EIx/SIx/CIx/DIx]`	ON region	11	-	30	V
`[EIx/SIx/CIx/DIx]`	Current (11-30V)	2	-	15	mA
`[EIx/SIx/CIx/DIx]`	Function	-	PNP	-	Type
`[EIx/SIx/CIx/DIx]`	IEC 61131-2	-	3	-	Type

NOTE:
The word “configurable” is used for I/O that can be configured as either safety-related I/O or normal I/O. These are the yellow terminals with black text.

4.3.2 Safety I/O

This section describes the dedicated safety inputs (Yellow terminal with red text) and the configurable I/O (Yellow terminals with black text) when configured as safety I/O. The common specifications in section 4.3.1 must be observed.

Safety devices and equipment must be installed according to the safety instructions and the risk assessment, see chapter 1.

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

The two permanent safety inputs are the emergency stop and the safeguard stop. The emergency stop input is for emergency stop equipment only. The safeguard stop input is for all kinds of safety-related protective equipment. The functional difference is shown below.

	Emergency Stop	Safeguard Stop
Robot stops moving	Yes	Yes
Program execution	Stops	Pauses
Robot 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	2
Performance level of monitoring function (ISO 13849-1)	PLd	PLd

It is possible to 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 II.

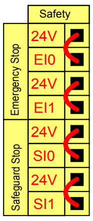
Some examples of how to use safety I/O are shown in the following subsections.

DANGER:

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 one of safety stop 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 stop function could be overridden.

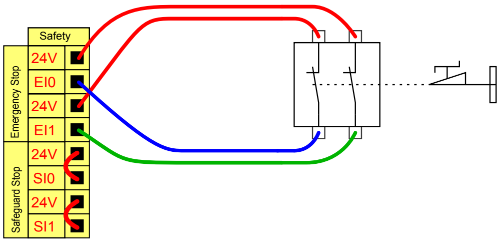
4.3.2.1 Default safety configuration

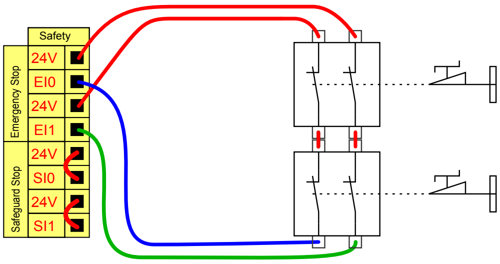
The robot is shipped with a default configuration which enables operation without any additional safety equipment, see illustration below.


Safety
Emergency Stop
Safeguard Stop
24V
EI0
24V
EI1
24V
SI0
24V
SI1

4.3.2.2 Connecting emergency stop buttons

In most applications it is required to use one or more extra emergency stop buttons. The illustration below show how one or more emergency stop buttons.


Safety
Emergency Stop
Safeguard Stop
24V
EI0
24V
EI1
24V
SI0
24V
SI1


Safety
Emergency Stop
Safeguard Stop
24V
EI0
24V
EI1
24V
SI0
24V
SI1

4.3.2.3 Sharing emergency stop with other machines

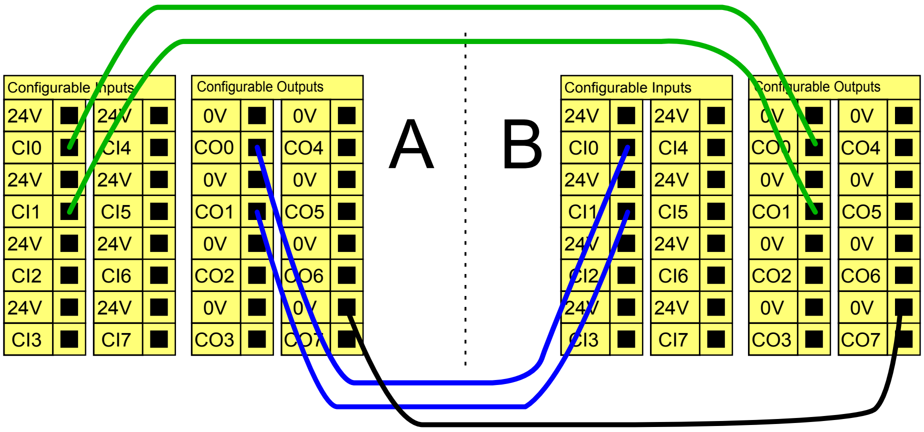
It is often desired to set up a common emergency stop circuit when the robot is used together with other machines. By doing so, the operator does not need to think about which emergency stop buttons to use.

The normal emergency stop input cannot be used for sharing purposes, since both machines will wait for the each other to go out of the emergency stopped condition.

In order to share the emergency stop function with other machinery, the following configurable I/O functions must be configured through the GUI.

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

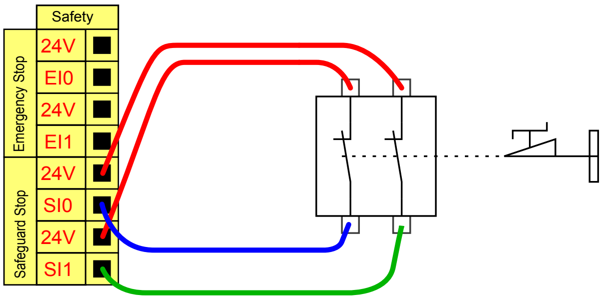
The illustration below shows how two UR robots share their emergency stop functions. In this example the configured I/Os used are "CI0-CI1" and "CO0-CO1".


A
B
Configurable Inputs
24V
CI0
24V
CI1
24V
CI2
24V
CI3
24V
CI4
24V
CI5
24V
CI6
24V
CI7
Configurable Outputs
0V
CO0
0V
CO1
0V
CO2
0V
CO3
0V
CO4
0V
CO5
0V
CO6
0V
CO7

If more than two UR robot or other machines needs to be connected, a safety PLC is needed to control the emergency stop signals.

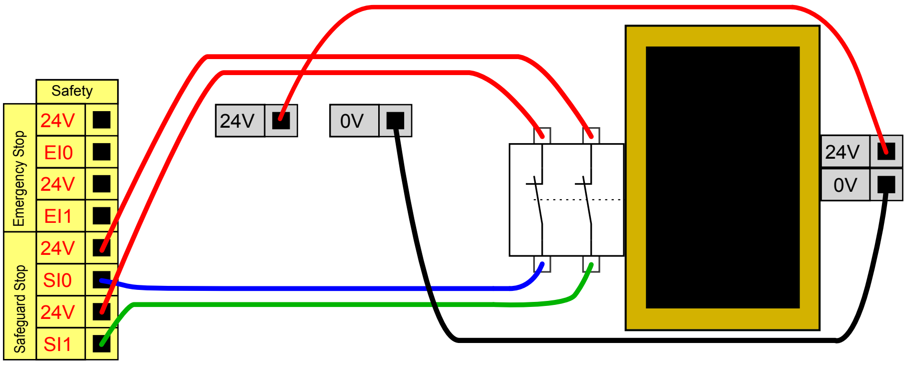
4.3.2.4 Safeguard stop with automatic resume

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.


Safety
Emergency Stop
Safeguard Stop
24V
EI0
24V
EI1
24V
SI0
24V
SI1

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

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

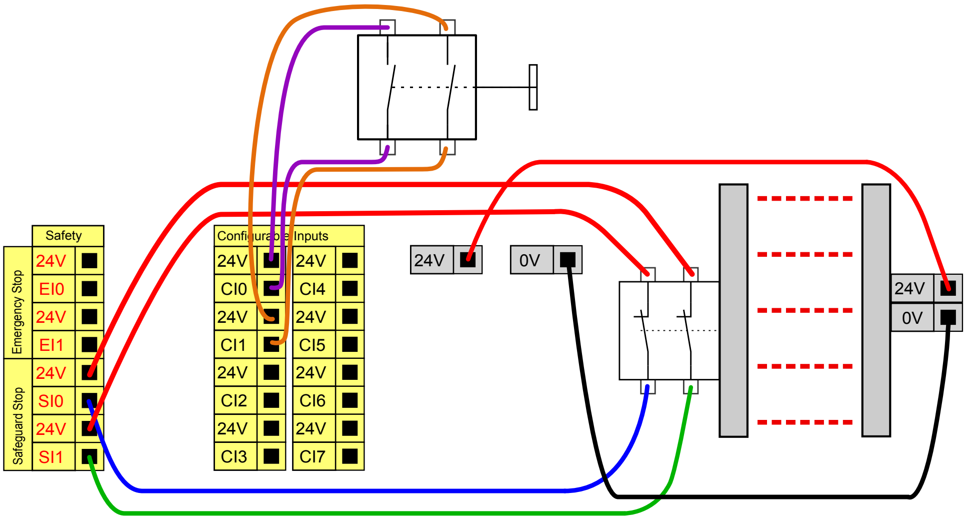

Safety
Emergency Stop
Safeguard Stop
24V
EI0
24V
EI1
24V
SI0
24V
SI1
24V
0V
24V
0V

DANGER:

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

4.3.2.5 Safeguard stop with reset button

If the safeguard interface is used to interface 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 “CI0-CI1”, see below.


Safety
Emergency Stop
Safeguard Stop
24V
EI0
24V
EI1
24V
SI0
24V
SI1
Configurable Inputs
24V
CI0
24V
CI1
24V
CI2
24V
CI3
24V
CI4
24V
CI5
24V
CI6
24V
CI7
24V
0V
24V
0V

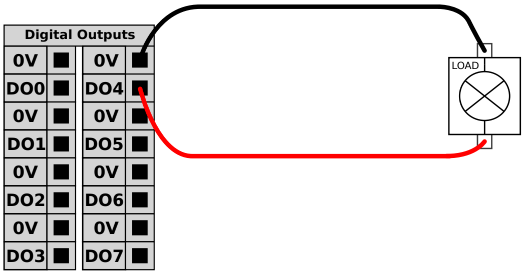
4.3.3 General purpose digital I/O

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 4.3.1 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 more in part II. 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 be used if they are not configured to perform a safety function.

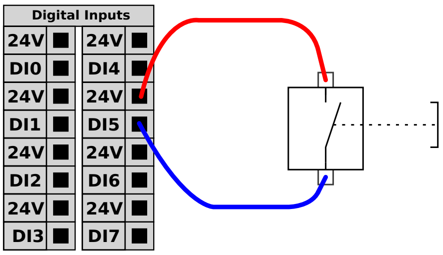
4.3.3.1 Load controlled by a digital output

This example shows how to connect a load to be controlled from a digital output, see below.


Digital Outputs
0V
DO0
0V
DO1
0V
DO2
0V
DO3
0V
DO4
0V
DO5
0V
DO6
0V
DO7
LOAD

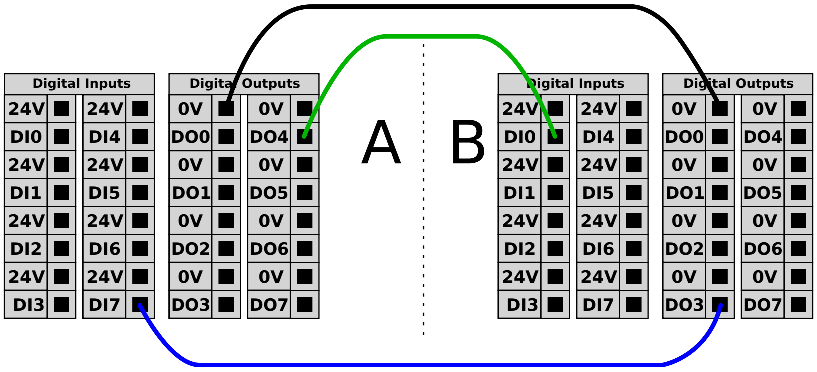
4.3.4 Digital input from a button

The example below shows how to connect a simple button to a digital input.


Digital Inputs
24V
DI0
24V
DI1
24V
DI2
24V
DI3
24V
DI4
24V
DI5
24V
DI6
24V
DI7

4.3.5 Communication with other machines or PLCs

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



4.3.6 General purpose analog I/O

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

The following is recommended to achieve a high 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 of equipment that works in current mode. Current signals are less sensitive to interferences.

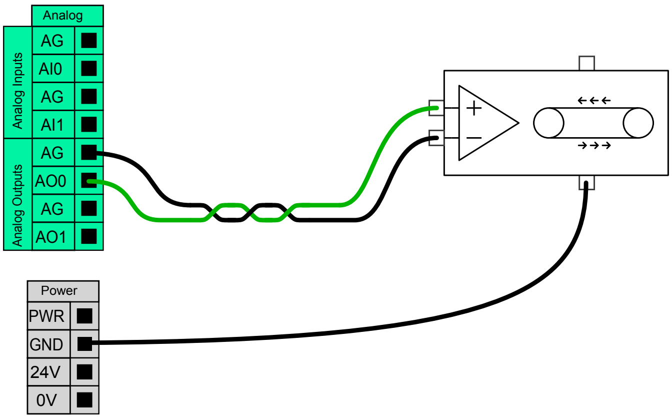
Input modes can be selected in the GUI, see part II. The electrical specifications are shown below.

Terminals	Parameter	Min	Typ	Max	Unit
Analog input in current mode					
`[AIx - AG]`	Current	4	-	20	mA
`[AIx - AG]`	Resistance	-	20	-	ohm
`[AIx - AG]`	Resolution	-	12	-	bit
Analog input in voltage mode					
`[AIx - AG]`	Voltage	0	-	10	V
`[AIx - AG]`	Resistance	-	10	-	Kohm
`[AIx - AG]`	Resolution	-	12	-	bit
Analog output in current mode					
`[AOx - AG]`	Current	4	-	20	mA
`[AOx - AG]`	Voltage	0	-	10	V
`[AOx - AG]`	Resolution	-	12	-	bit
Analog output in voltage mode					
`[AOx - AG]`	Voltage	0	-	10	V
`[AOx - AG]`	Current	-20	-	20	mA
`[AOx - AG]`	Resistance	-	1	-	ohm
`[AOx - AG]`	Resolution	-	12	-	bit

The following examples show how to use the analog I/O.

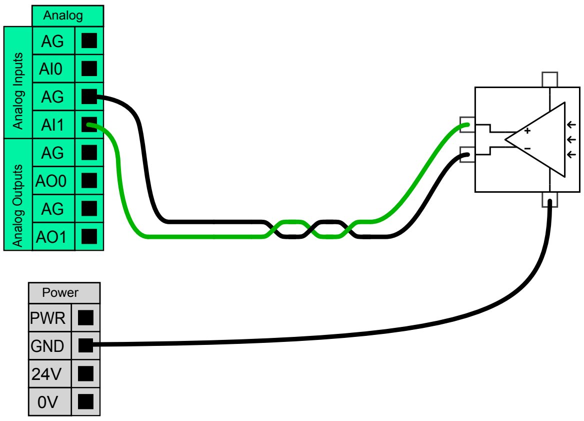
4.3.6.1 Using an analog output

Below is an example of how to control a conveyor belt with an analog speed control input.



4.3.6.2 Using an Analog Input

Below is an example of how to connect an analog sensor.



4.3.7 Remote ON/OFF control

Remote ON/OFF control can be used to turn the control box on and off without using the teach pendant. It is typically used in the following applications:

- 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 small auxiliary 12V supply, which is kept active when the controller box is turned off. The “on” and “off” inputs are intended for short time activation only. The on input works in the same way as the power button. Always use the “off” input for remote off control as this signal allows the control box safe files and shut down nicely.

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

The following examples show how to use remote ON/OFF.

NOTE:

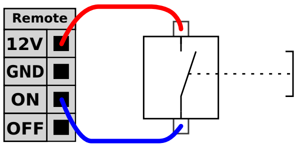
A special feature in the software can be used to load and start programs automatically, see part II.

CAUTION:

1. Never use the “on” input or the power button to turn off the control box.

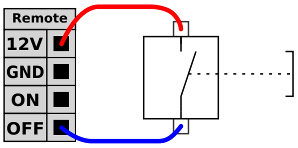
4.3.7.1 Remote ON button

The illustration below shows how to connect a remote on button.



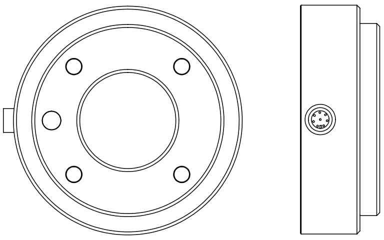
4.3.7.2 Remote OFF button

The illustration below shows how to connect a remote off button.



4.4 Tool I/O

At the tool end of the robot there is a small connector with eight pins, see illustration below.



This connector provides power and control signals for grippers and sensors used on a specific robot tool. The following industrial cables are suitable:

- Lumberg RKMV 8-354.

The eight wires inside the cable have different colors. The different colors designate different functions, see table below:

Color	Signal
Red	0V (GND)
Gray	0V/+12V/+24V (POWER)
Blue	Digital output 8 (DO8)
Pink	Digital output 9 (DO9)
Yellow	Digital input 8 (DI8)
Green	Digital input 9 (DI9)
White	Analog input 2 (AI2)
Brown	Analog input 3 (AI3)

The internal power supply can be set to either 0V, 12V or 24V at the I/O tab the GUI, see part II. The electrical specifications are shown below:

Parameter	Min	Typ	Max	Unit
Supply voltage in 24V mode	-	24	-	V
Supply voltage in 12V mode	-	12	-	V
Supply current in both modes	-	-	600	mA

The following sections describe the different I/O's of the tool.

DANGER:

1. Construct tools and gripper so that an interruption of power does not create any hazards. E.g. a work-piece falling out of the tool.
2. Take care when using 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.

NOTE:

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

4.4.1 Tool Digital Outputs

The digital outputs are implemented as NPN. When a digital output is activated the corresponding connection is driven to GND, and when it is deactivated the corresponding connection is open (open-collector/open-drain). The electrical specifications are shown below:

Parameter	Min	Typ	Max	Unit
Voltage when open	-0.5	-	26	V
Voltage when sinking 1A	-	0.05	0.20	V
Current when sinking	0	-	1	A
Current through GND	0	-	1	A

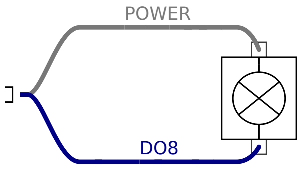
An example of how to use a digital output is shown in the following subsection.

CAUTION:

1. The digital outputs in the tool are not current limited and overriding the specified data can cause permanent damage.

4.4.1.1 Using the Tool Digital Outputs

The example below illustrates how to turn on a load, when using the internal 12V or 24V power supply. Remember that you have to define the output voltage at the I/O tab. Keep in mind that there is voltage between the POWER connection and the shield/ground, even when the load is turned off.



4.4.2 Tool Digital Inputs

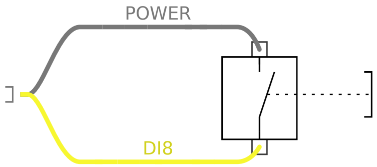
The digital inputs are implemented as PNP with weak pull-down resistors. This means that a floating input will always read low. The electrical specifications are shown below.

Parameter	Min	Typ	Max	Unit
Input voltage	-0.5	-	26	V
Logical low voltage	-	-	2.0	V
Logical high voltage	5.5	-	-	V
Input resistance	-	47k	-	Ω

An example of how to use a digital input is shown in the following subsection.

4.4.2.1 Using the Tool Digital Inputs

The example below shows how to connect a simple button.



4.4.3 Tool Analog Inputs

The tool analog inputs are non-differential and can be set to either voltage and current on the I/O tab, see part II. The electrical specifications are shown below.

Parameter	Min	Typ	Max	Unit
Input voltage in voltage mode	-0.5	-	26	V
Input voltage in current mode	-0.5	-	5.0	V
Input current in current mode	-2.5	-	25	mA
Input resistance @ range 0V to 5V	-	29	-	kΩ
Input resistance @ range 0V to 10V	-	15	-	kΩ
Input resistance @ range 4mA to 20mA	-	200	-	Ω

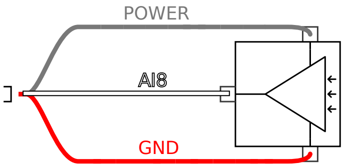
Two examples of how to use an analog inputs are shown in the following subsections.

CAUTION:

1. Analog inputs are not protected against over voltage in current mode. Overrating the limit in the electrical specification can cause permanent damage to the input.

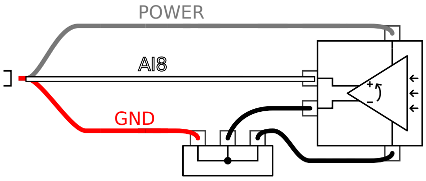
4.4.3.1 Using the Tool Analog Inputs, Non-differential

The example below shows how to connect an analog sensor with a non-differential output. The output of the sensor 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. Remember to check that a sensor with voltage output can drive the internal resistance of the tool, or the measurement might be invalid.



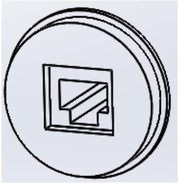
4.4.3.2 Using the Tool Analog Inputs, Differential

The example below shows how to connect an analog sensor with a differential output. Connect the negative output part to GND (0V) and it works in the same way as a non-differential sensor.



4.5 Ethernet

An Ethernet connection is provided at the bottom of the control box, see illustration below.



The Ethernet interface can be used for the following:

- MODBUS I/O expansion modules. See more in part II.
- Remote access and control.

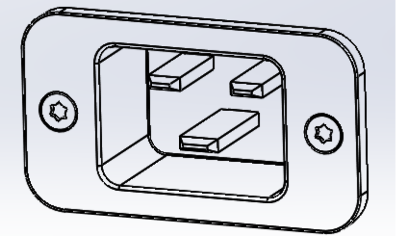
The electrical specifications are shown below.

Parameter	Min	Typ	Max	Unit
Communication speed	10	-	100	Mb/s

4.6 Mains connection

The mains cable from the controller box has a standard IEC plug in 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. This must be done through the standard IEC C20 plug at the bottom of the control box through a corresponding IEC C19 cord, see illustration below.



The mains supply shall be equipped with the following as a minimum:

- Connection to earth.
- Main fuse.
- Residual current device.

It is recommended to install a main switch to power of all equipment in the robot application as an easy means for lockout and tagout under service.

The electrical specifications are shown in the table below.

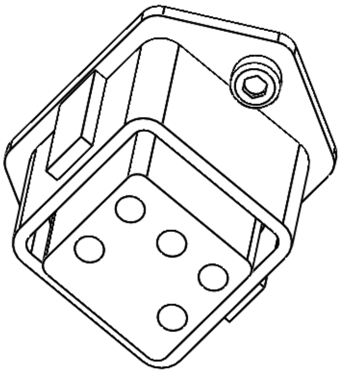
Parameter	Min	Typ	Max	Unit
Input voltage	100	-	240	VAC
External mains fuse (@ 100-200V)	15	-	16	A
External mains fuse (@ 200-240V)	8	-	16	A
Input frequency	47	-	63	Hz
Stand-by power	-	-	0.5	W
Nominal operating power	90	250	500	W

DANGER:

1. Make sure that the robot is grounded correctly (Electrical connection to earth). Use the unused bolts associated with grounding symbols inside the controller 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. Make sure that the input power to the controller box is protected with a RCD (Residual Current Device) and a correct fuse.
3. Lockout and 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. Make sure that all cables are connected correctly before the controller box is powered. Always use an original and correct power cord.

4.7 Robot connection

The cable from the robot must be plugged into the connector at bottom of the control box, see illustration below. Ensure that the connector is properly locked before turning on the robot arm. Disconnecting the robot cable may only be done when the robot power is turned off.



CAUTION:

1. Do not disconnect the robot cable when the robot arm is turned on.
2. Do not extend or modify the original cable.

5 Safety-related Functions and Interfaces

UR robots are equipped with a range of built-in safety-related functions as well as safety-related electrical interfaces to connect to other machines and additional protective devices. Each safety function and interface is monitored according to ISO 13849-1 (see Chapter 8 for certifications) with Performance Level d (PLd).

NOTE:
If the robot discovers a fault in the safety system, e.g. one of the wires in the emergency stop circuit is cut, or a position sensor is broken, a category 0 stop is initiated. The worst case reaction time, from the time an error occurs to the point in time that it is detected, and the robot is stopped and powered off, is 1250 ms.

Part II of the PolyScope Manual describes configuration of the safety-related features, inputs, and outputs. See Chapter 4 for descriptions on how to connect safety devices to the electrical interface.

5.1 Limiting Safety-related Functions

The robot has a number of safety-related functions that can be used to limit the movement of its joints and of the robot *Tool Center Point* (TCP). The TCP is the center point of the output flange with the addition of the TCP offset (see Part II, the PolyScope Manual).

The limiting safety-related functions are:

Limiting Safety Function	Description
Joint position	Min. and max. angular joint position
Joint speed	Max. angular joint speed
TCP position	Planes in Cartesian space limiting robot TCP position
TCP speed	Max. speed of the robot TCP
TCP force	Max. pushing force of the robot TCP
Momentum	Max. momentum of the robot arm
Power	Max. applied robot arm power

Advanced path control software decreases speed or issues a program execution stop if the robot arm approaches a safety-related limit. Violations of limits will hence only occur in exceptional cases. Nevertheless, if a limit is violated, the safety system issues a category 0 stop with the performance listed in the table:

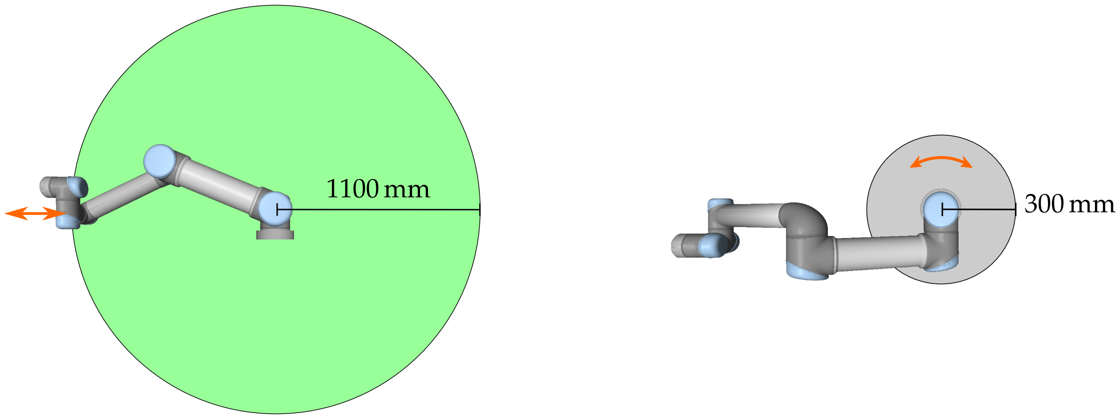


Figure 5.1: Certain areas of the workspace should receive attention regarding pinching hazards, due to the physical properties of the robot arm. One area is defined for radial motions, when the wrist 1 joint is at a distance of at least 1100 mm from the base of the robot. The other area is within 300 mm of the base of the robot, when moving in the tangential direction.

	Worst Case			
Limiting Safety Function	Trueness	Detection Time	De-energizing Time	Reaction Time
Joint position	1.15 °	100 ms	1000 ms	1100 ms
Joint speed	1.15 °/s	250 ms	1000 ms	1250 ms
TCP position	20 mm	100 ms	1000 ms	1100 ms
TCP orientation	1.15 °	100 ms	1000 ms	1100 ms
TCP speed	50 mm/s	250 ms	1000 ms	1250 ms
TCP force	25 N	250 ms	1000 ms	1250 ms
Momentum	3 kg m/s	250 ms	1000 ms	1250 ms
Power	10 W	250 ms	1000 ms	1250 ms

The system is considered *de-energized* when the 48 V bus voltage reaches an electrical potential below 7.3 V. The de-energizing time is the time from a detection of an event until the system has been de-energized.

WARNING:

There are two exceptions to the force limiting function that are important to notice when designing the work cell for the robot. These are illustrated in Figure 5.1. As the robot stretches out, the knee-joint effect can give high forces in the radial direction (away from the base), but at the same time, low speeds. Similarly, the short leverage arm, when the tool is close to the base and moving tangential (around) the base, can cause high forces, but also at low speeds. Pinching hazards can be avoided for instance by, removing obstacles in these areas, placing the robot differently, or by using a combination of safety planes and joint limits to remove the hazard by preventing the robot moving into this region of its workspace.

5.2 Safety Modes

Normal and Reduced mode The safety system has two configurable safety modes: *Normal* and *Reduced*. Safety limits can be configured for each of these two modes. Reduced mode is active when the robot TCP is positioned beyond a *Trigger Reduced mode* plane or when triggered by a safety input.

On the side of the *Trigger Reduced mode* planes where the normal mode limit set is defined, there is an area of 20 mm where the reduced mode limit set is accepted. When Reduced mode is triggered by a safety input, both limit sets are accepted for 500 ms.

Recovery Mode When a safety limit is violated, 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 II of the PolyScope Manual). The safety limits of *Recovery* mode are:

Limiting Safety Function	Limit
Joint speed	30 °/s
TCP speed	250 mm/s
TCP force	100 N
Momentum	10 kg m/s
Power	80 W

The safety system issues a category 0 stop if a violation of these limits appears.

WARNING:
Notice that limits for the *joint position*, the *TCP position*, and the *TCP orientation* are disabled in Recovery Mode. Take caution when moving the robot arm back within the limits.

5.3 Safety-related Electrical Interfaces

The robot is equipped with several safety-related electrical inputs and outputs. All safety-related electrical inputs and outputs are dual channel. They are safe when low, e.g. the emergency stop is not active when the signal is high (+24V).

5.3.1 Safety-related Electrical Inputs

The table below gives an overview of the safety-related electrical inputs.

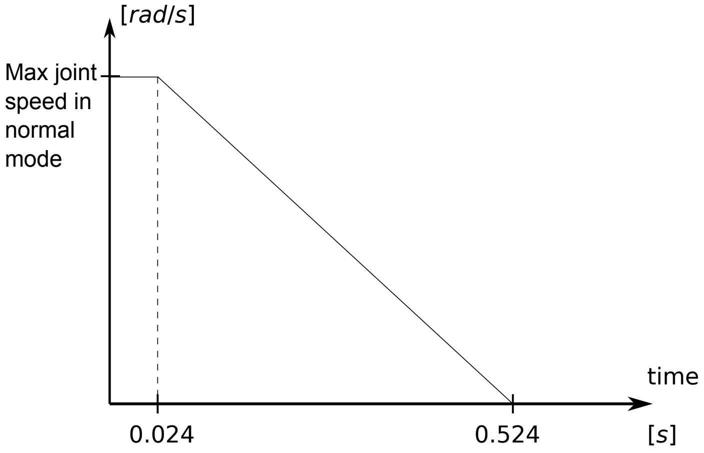


Figure 5.2: The green area below the ramp is the allowed speeds for a joint during braking. At time 0 an event (emergency stop or safeguard stop) is detected at the safety processor. Deceleration begins after 24 ms.

Safety Input	Description
Robot emergency stop	Performs a category 1 stop, informing other machines using the *System emergency stop* output.
Emergency stop button	Performs a category 1 stop, informing other machines using the *System emergency stop* output.
System emergency stop	Performs a category 1 stop.
Safeguard stop	Performs a category 2 stop.
Safeguard reset input	Resumes the robot from a *Safeguard stopped* state, when an edge on the Safeguard reset input occurs.
Reduced mode	The safety system transitions to *Reduced* mode limits.

A category 1 and 2 stop decelerates the robot with drive power on, which enables the robot to stop without deviating from its current path.

Monitoring of safety inputs Category 1 and 2 stops are monitored by the safety system in the following way:

1. The safety system monitors that the braking initiates within 24 ms, see Figure 5.2.
2. If a joint is moving, its speed is monitored to never be higher than the speed obtained by constantly decelerating from the maximum joint speed limit for *Normal* mode to 0 rad/s in 500 ms.
3. If a joint is at rest (joint speed is less than 0.2 rad/s), it is monitored that it does not move more than 0.05 rad from the position it had when the speed was measured below 0.2 rad/s.

Additionally, for a category 1 stop, the safety system monitors that after the robot arm is at rest, the powering off is finalized within 600 ms. Furthermore, after a safeguard stop input, the robot arm is only allowed to start moving again after a positive edge on the safeguard reset input occurs. If any of the above properties are not satisfied, the safety system issues a category 0 stop.

A transition to *Reduced* mode triggered by the reduced mode input is monitored as follows:

1. The safety system accepts both *Normal* and *Reduced* mode limit sets for 500 ms after the reduced mode input is triggered.
2. After 500 ms, only the *Reduced* mode limits are in effect.

If any of the above properties are not satisfied, the safety system issues a category 0 stop.

A category 0 stop is performed by the safety system with the performance described in the following table. The worst-case reaction time is the time to stop and to *de-energize* (discharge to an electrical potential below 7.3 V) a robot running at full speed and payload.

	Worst Case		
Safety Input Function	Detection Time	De-energizing Time	Reaction Time
Robot emergency stop	250 ms	1000 ms	1250 ms
Emergency stop button	250 ms	1000 ms	1250 ms
System emergency stop	250 ms	1000 ms	1250 ms
Safeguard stop	250 ms	1000 ms	1250 ms

5.3.2 Safety-related Electrical Outputs

The table below gives an overview of the safety-related electrical outputs:

Safety Output	Description
System emergency stop	Activated by an active *Robot emergency stop* input or by the Emergency stop button.
Robot moving	While this signal is inactive, no single joint of the robot arm moves more than 0.1 rad.
Robot not stopping	Inactive when the robot arm has been requested to stop and has not stopped yet.
Reduced mode	Active when the safety system is in *Reduced* mode.
Not reduced mode	The *Reduced mode* output negated.

If a safety output is not set properly, the safety system issues a category 0 stop, with the following worst-case reaction times:

Safety Output	Worst Case Reaction Time
System emergency stop	1100 ms
Robot moving	1100 ms
Robot not stopping	1100 ms
Reduced mode	1100 ms
Not reduced mode	1100 ms

6 Maintenance and Repair

It is essential for both maintenance and repair work that it be performed in compliance with all safety instructions in this manual.

Maintenance, calibration and repair work must be performed according to the newest versions of Service Manuals found on the support website `http://support.universal-robots.com`. All UR distributors have access to this support site.

Repairs shall only be performed by authorized system integrators or by Universal Robots.

All parts returned to Universal Robots shall be returned according to the service manual.

6.1 Safety Instructions

After maintenance and repair work, checks must be carried out to ensure the required safety level. The valid national or regional work safety regulations must be observed for this check. 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.

The following safety procedures and warnings must be observed when working on the robot arm or control box.

DANGER:

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.

DANGER:

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.

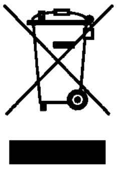
7 Disposal and Environment

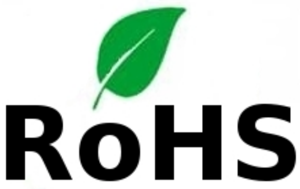
UR robots must be disposed of in accordance with the applicable national laws, regulations and standards.

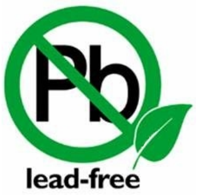
UR robots are produced with restricted use of hazardous substances to protect the environment; as defined by the European RoHS directive 2011/65/EU. These substances include mercury, cadmium, lead, chromium VI, polybrominated biphenyls and polybrominated diphenyl ethers.

Fee for disposal and handling of electronic waste of UR 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:







8 Certifications

This chapter presents a range of different certifications and declarations that have been prepared for the product.

8.1 Third Party Certifications

Third party certifications are voluntary. However, to provide the best service to robot integrators, UR has chosen to certify their robots at the following recognized test institutes:

TÜV NORD	UR robots are safety approved by TÜV NORD, a notified body under the machinery directive 2006/42/EC in EU. A copy of the TÜV NORD safety approval certificate can be found in appendix B.
DELTA	UR robots are safety and performance tested by DELTA. An electromagnetic compatibility (EMC) certificate can be found in appendix B. An environmental test certificate can be found in appendix B.

8.2 Declarations According to EU directives

EU declarations are primarily relevant for European countries. However, some countries outside Europe recognize or even require them, too. European directives are available from the official homepage: `http://eur-lex.europa.eu`.

UR robots are certified according to the directives listed below.

2006/42/EC — Machinery Directive (MD)

UR Robots are partly completed machinery according to the Machinery Directive 2006/42/EC. Note that a CE mark is not affixed according to this directive for partly completed machinery. If the UR robot is used in a pesticide application, then note the presence of directive 2009/127/EC. The declaration of incorporation according to 2006/42/EC annex II 1.B. is shown in appendix B.

2006/95/EC — Low Voltage Directive (LVD)

2004/108/EC — Electromagnetic Compatibility (EMC)

2011/65/EU — Restriction of the use of certain Hazardous Substances (RoHS)

2012/19/EU — Waste of Electrical and Electronic Equipment (WEEE)

Declarations of conformity with the above Directives are included in the declaration of incorporation in appendix B.

A CE mark is affixed according to CE marking directives above. Regarding waste of electric and electronic equipment see chapter 7.

For information about standards applied during the development of the robot, see appendix C.

9 Warranties

9.1 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.

9.2 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.

A Stopping Time and Stopping Distance

The information about stopping times and distances is available for both CATEGORY 0 and CATEGORY 1 stops. This appendix includes the information regarding stop CATEGORY 0. Information on CATEGORY 1 stop is available on `http://support.universal-robots.com/`.

A.1 CATEGORY 0 stopping distances and times

The table below includes the stopping distances and times measured when a CATEGORY 0 stop is triggered. These measurements correspond to the following configuration of the robot:

- Extension: 100% (the robot arm is fully extended horizontally).
- Speed: 100% (the general speed of the robot is set to 100% and the movement is performed at a joint speed of 183 °/s).
- Payload: maximum payload handled by the robot attached to the TCP (10 kg).

The test on the Joint 0 was carried out by performing a horizontal movement, i.e. the axis of rotation was perpendicular to the ground. During the tests for Joint 1 and 2 the robot followed a vertical trajectory, i.e. the axes of rotation were parallel to the ground, and the stop was performed while the robot was moving downwards.

	Stopping Distance (rad)	Stopping time (ms)
Joint 0 (BASE)	0.98	750
Joint 1 (SHOULDER)	0.35	240
Joint 2 (ELBOW)	0.38	280

B Declarations and Certificates

B.1 CE Declaration of Incorporation (original)

According to European directive 2006/42/EC annex II 1.B.

The manufacturer
Universal Robots A/S
Energivej 25
5260 Odense S
Denmark
+45 8993 8989

hereby declares that the product described below

Industrial robot UR10
Robot serial number ______________________
Control box serial number ______________________

may not be put into service before the machinery in which it will be incorporated is declared to comply with the provisions of Directive 2006/42/EC, as amended by Directive 2009/127/EC, and with the regulations transposing it into national law.

The safety features of the product are prepared for compliance with all essential requirements of Directive 2006/42/EC under the correct incorporation conditions, see product manual. Compliance with all essential requirements of Directive 2006/42/EC relies on the specific robot installation and the final risk assessment.

Relevant technical documentation is compiled according to Directive 2006/42/EC annex VII part B.

Additionally the product declares in conformity with the following directives, according to which the product is CE marked:

2006/95/EC — Low Voltage Directive (LVD)
2004/108/EC — Electromagnetic Compatibility Directive (EMC)
2011/65/EU — Restriction of the use of certain hazardous substances (RoHS)

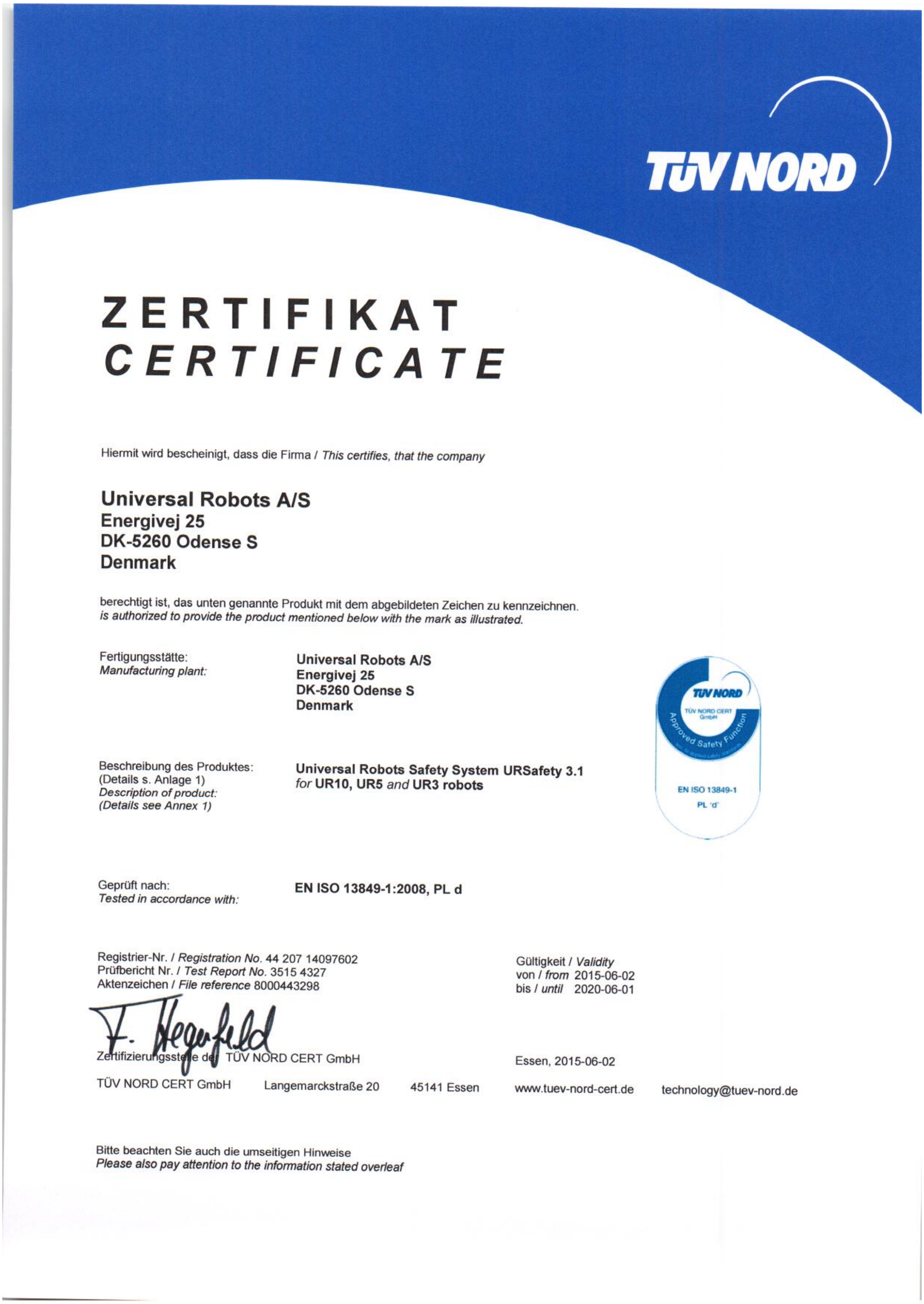
A complete list of applied harmonized standards, including associated specifications, is provided in the product manual. This list is valid for the product manual with the same serial numbers as this document and the product.

Odense, January 27th, 2015

R&D

Lasse Kieffer
Global Compliance Officer

B.2 Safety System Certificate



TÜV NORD

ZERTIFIKAT
CERTIFICATE

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

Universal Robots A/S
Energivej 25
DK-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** **DK-5260 Odense S** **Denmark**
Beschreibung des Produktes: (Details s. Anlage 1) *Description of product:* *(Details see Annex 1)*	**Universal Robots Safety System URSafety 3.1** *for* **UR10, UR5** *and* **UR3 robots**
Geprüft nach: *Tested in accordance with:*	**EN ISO 13849-1:2008, PL d**

EN ISO 13849-1
PL 'd'

Registrier-Nr. / *Registration No.* 44 207 14097602
Prüfbericht Nr. / *Test Report No.* 3515 4327
Aktenzeichen / *File reference* 8000443298

Gültigkeit / *Validity*
von / *from* 2015-06-02
bis / *until* 2020-06-01

F. Hegerfeld

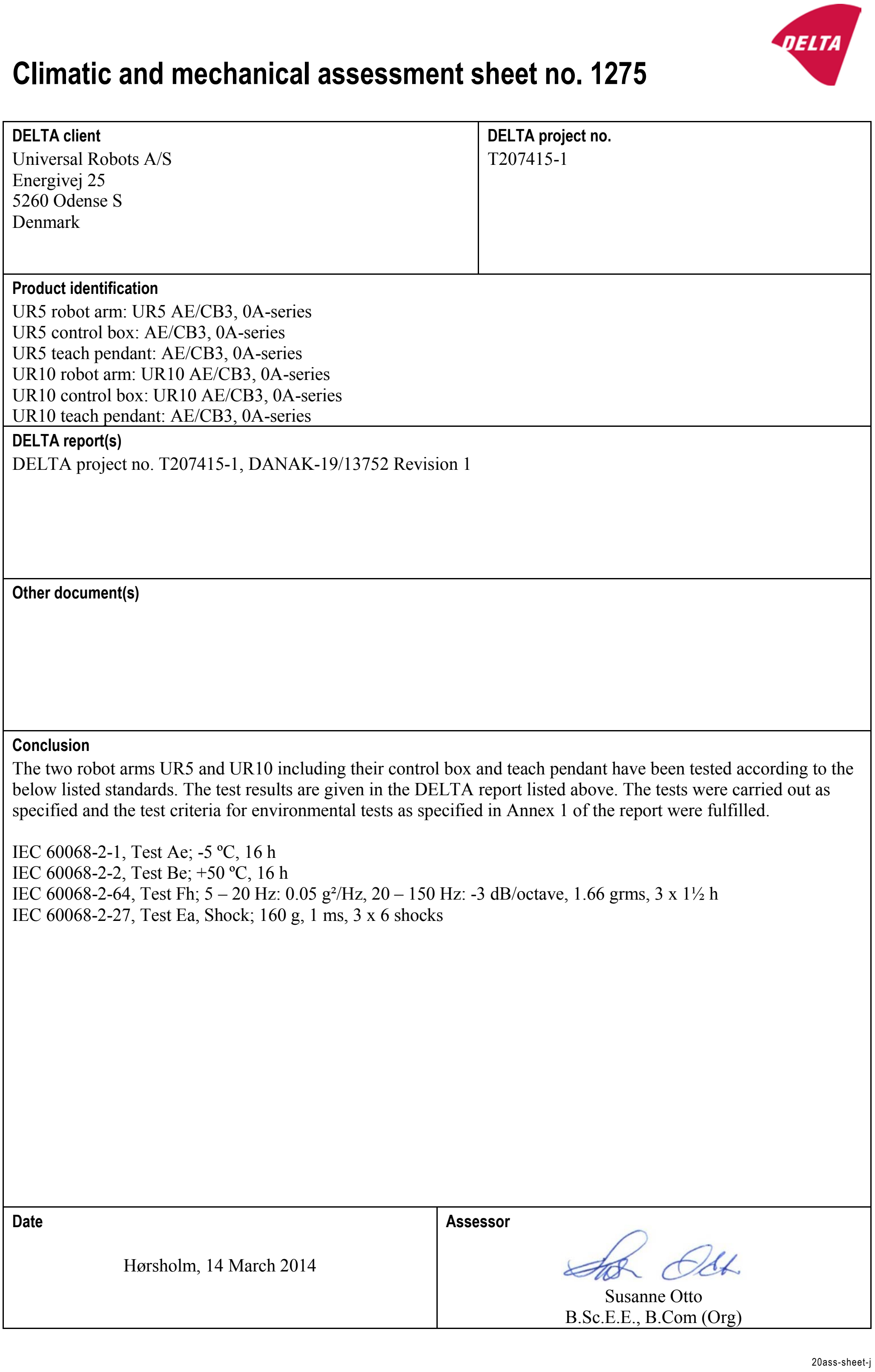
Zertifizierungsstelle der TÜV NORD CERT GmbH

Essen, 2015-06-02

TÜV NORD CERT GmbH Langemarckstraße 20 45141 Essen www.tuev-nord-cert.de technology@tuev-nord.de

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

B.3 Environmental Test Certificate



Climatic and mechanical assessment sheet no. 1275

DELTA client Universal Robots A/S Energivej 25 5260 Odense S Denmark	**DELTA project no.** T207415-1

Product identification

UR5 robot arm: UR5 AE/CB3, 0A-series
UR5 control box: AE/CB3, 0A-series
UR5 teach pendant: AE/CB3, 0A-series
UR10 robot arm: UR10 AE/CB3, 0A-series
UR10 control box: UR10 AE/CB3, 0A-series
UR10 teach pendant: AE/CB3, 0A-series

DELTA report(s)

DELTA project no. T207415-1, DANAK-19/13752 Revision 1

Other document(s)

Conclusion

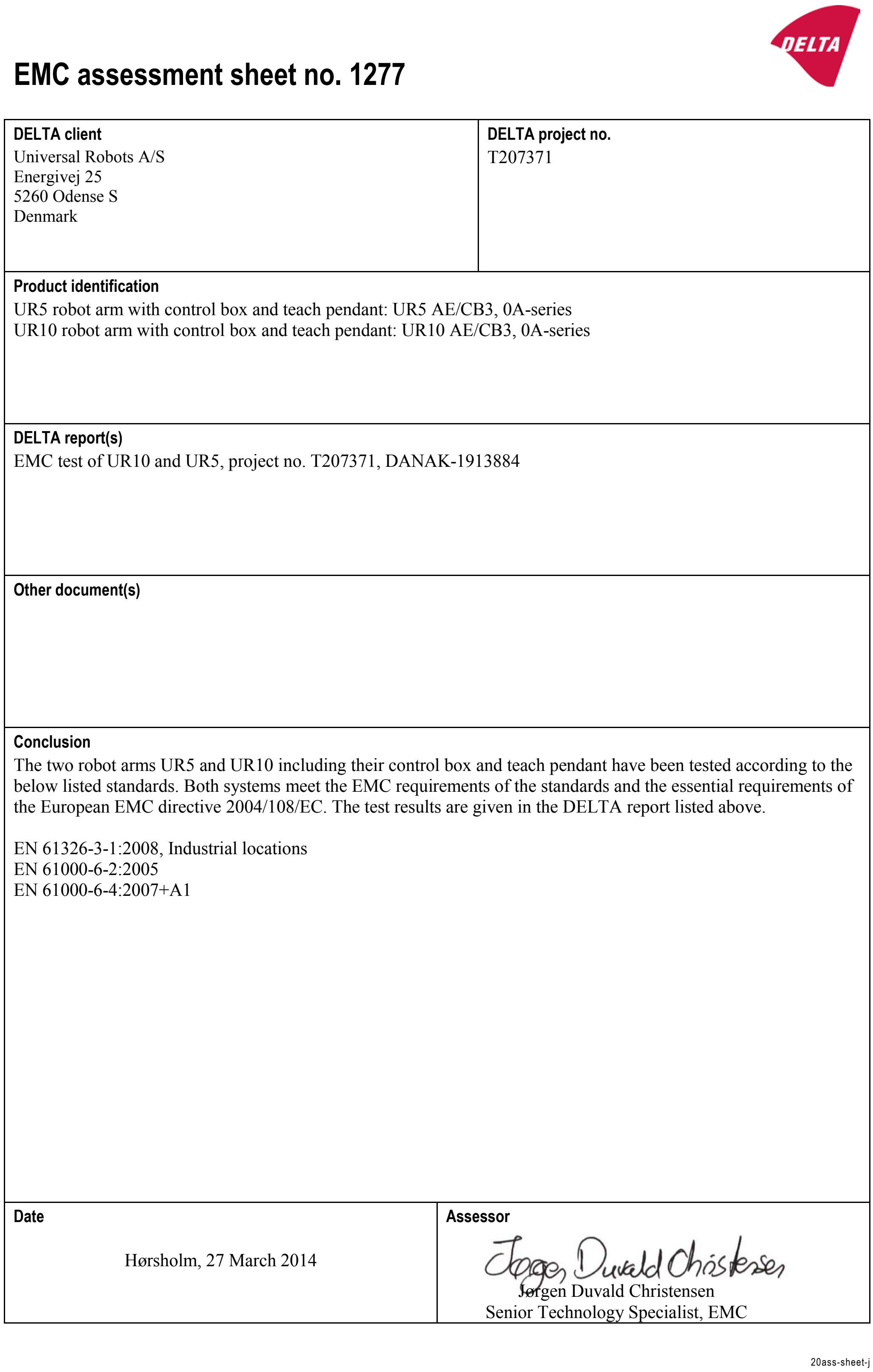
The two robot arms UR5 and UR10 including their control box and teach pendant have been tested according to the below listed standards. The test results are given in the DELTA report listed above. The tests were carried out as specified and the test criteria for environmental tests as specified in Annex 1 of the report were fulfilled.

IEC 60068-2-1, Test Ae; -5 ºC, 16 h
IEC 60068-2-2, Test Be; +50 ºC, 16 h
IEC 60068-2-64, Test Fh; 5 – 20 Hz: 0.05 g²/Hz, 20 – 150 Hz: -3 dB/octave, 1.66 grms, 3 x 1½ h
IEC 60068-2-27, Test Ea, Shock; 160 g, 1 ms, 3 x 6 shocks

Date Hørsholm, 14 March 2014	**Assessor** Susanne Otto B.Sc.E.E., B.Com (Org)

20ass-sheet-j

B.4 EMC Test Certificate



EMC assessment sheet no. 1277

DELTA client Universal Robots A/S Energivej 25 5260 Odense S Denmark	**DELTA project no.** T207371

Product identification

UR5 robot arm with control box and teach pendant: UR5 AE/CB3, 0A-series
UR10 robot arm with control box and teach pendant: UR10 AE/CB3, 0A-series

DELTA report(s)

EMC test of UR10 and UR5, project no. T207371, DANAK-1913884

Other document(s)

Conclusion

The two robot arms UR5 and UR10 including their control box and teach pendant have been tested according to the below listed standards. Both systems meet the EMC requirements of the standards and the essential requirements of the European EMC directive 2004/108/EC. The test results are given in the DELTA report listed above.

EN 61326-3-1:2008, Industrial locations
EN 61000-6-2:2005
EN 61000-6-4:2007+A1

Date Hørsholm, 27 March 2014	**Assessor** Jørgen Duvald Christensen Jørgen Duvald Christensen Senior Technology Specialist, EMC

20ass-sheet-j

C Applied Standards

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

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

Standard type abbreviations 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 13849-2:2012
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]
EN ISO 13850:2008 (E) [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.

The writers of the standard implicitly envisioned traditional industrial robots, which are traditionally safeguarded by fences and light curtains. UR robots are designed with force and power limiting enabled at all times. Therefore, some concepts are clarified and explained below.

If a UR robot is used in a hazardous application, additional safety measures might be required, see chapter 1 of this manual.

Clarification:

- "3.24.3 Safeguarded space" is defined by the perimeter safeguarding. Typically, the safeguarded space is a space behind a fence, which protects people from the hazardous traditional robots. UR robots are designed to work without a fence using a built-in power and force limiting safety function, where there is no hazardous safeguarded space defined by the perimeter of a fence.
- "5.4.2 Performance requirement". All safety functions are constructed as PLd according to ISO 13849-1:2006. The robot is constructed with redundant encoder systems in each joint, and the safety-rated I/Os are constructed with a category 3 structure. The safety-rated I/Os must be connected according to this manual to category 3 safety-rated equipment to form a PLd structure of the complete safety function.
- "5.7 Operating modes". UR robots do not have different operating modes and therefore they do not have a mode selector.
- "5.8 Pendant controls". This section defines protective features for the teach pendant, when it is to be used within a hazardous safeguarded space. Since UR robots are power and force limited, there is no hazardous safeguarded space like with traditional robots. UR robots are safer to teach than traditional robots. Instead of having to release a three-positioning enabling device, the operator can simply stop the robot with his hand.
- "5.10 Collaborative operation requirements". The power and force limiting function of UR robots is always active. The visual design of the UR robots indicates that the robots are capable of being used for collaborative operations. The power and force limiting function is designed according to clause 5.10.5.
- "5.12.3 Safety-rated soft axis and space limiting". This safety function is one of more safety functions configurable through software. A hash code is generated from the sittings of all these safety functions and is represented as a safety check identifier in the GUI.

ISO/DTS 15066 (Draft)

Robots and robotic devices – Safety requirements for industrial robots – Collaborative operation

This is a Technical Specification (TS) under preparation. A TS is not a standard. The purpose of a TS is to present a set of immature requirements to see if they are useful for a given industry.

This TS presents technologies and force-related safety limits for collaborative robots, where the robot and the human work together to perform a work task.

Universal Robots is an active member of the international committee that develops this TS (ISO/TC 184/SC 2). A final version might be published in 2016.

ANSI/RIA R15.06-2012

Industrial Robots and Robot Systems – Safety Requirements

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

Note that 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.

Note that 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
EN 60320-1/A1:2007 [2006/95/EC]

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 Hardware Installation Manual.

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 standards defines requirements for safety-rated communication buses.

IEC 60204-1/A1:2008
EN 60204-1/A1:2009 [2006/42/EC]

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 [2006/95/EC]

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, V1.10

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.

D Technical Specifications

Robot type	UR10
Weight	28.9 kg / 63.7 lb
Payload	10 kg / 22 lb
Reach	1300 mm / 51.2 in
Joint ranges	± 360 ° for all joints
Speed	Base and Shoulder joints: Max 120 °/s. All other joints: Max 180 °/s. Tool: Approx. 1 m/s / Approx. 39.4 in/s.
Repeatability	± 0.1 mm / ± 0.0039 in (4 mils)
Footprint	Ø190 mm / 7.5 in
Degrees of freedom	6 rotating joints
Control box size (W × H × D)	475 mm × 423 mm × 268 mm / 18.7 in × 16.7 in × 10.6 in
Control box I/O ports	16 digital in, 16 digital out, 2 analogue in, 2 analogue out
Tool I/O ports	2 digital in, 2 digital out, 2 analogue in
I/O power supply	24 V 2 A in control box and 12 V/24 V 600 mA in tool
Communication	TCP/IP 100 Mbit: IEEE 802.3u, 100BASE-TX Ethernet socket & Modbus TCP
Programming	PolyScope graphical user interface on 12” touchscreen with mounting
Noise	Comparatively noiseless
IP classification	IP54
Power consumption	Approx. 350 W using a typical program
Collaboration operation	Collaborative operation according to ISO 10218-1:2011
Temperature	The robot can work in a temperature range of 0-50 °C
Power supply	100-240 VAC, 50-60 Hz
Calculated operating life	35,000 hours
Cabling	Cable between robot and control box (6 m / 236 in) Cable between touchscreen and control box (4.5 m / 177 in)

Part II

PolyScope Manual

10 Introduction

The Universal Robot arm is composed of extruded aluminum tubes and joints. The joints with their usual names are shown in Figure 10.1. The *Base* is where the robot is mounted, and at the other end (*Wrist 3*) the tool of the robot is attached. By coordinating the motion of each of the joints, the robot can move its tool around freely, with the exception of the area directly above and directly below the base.

PolyScope is the graphical user interface (GUI) which lets you operate the robot arm and control box, execute robot programs and easily create new ones.

The following section gets you started with the robot. Afterwards, the screens and functionality of PolyScope are explained in more detail.

10.1 Getting Started

Before using PolyScope, the robot arm and control box must be installed and the control box switched on.

10.1.1 Installing the Robot Arm and Control Box

To install the robot arm and control box, do the following:

1. Unpack the robot arm and the control box.
2. Mount the robot arm on a sturdy and vibration-free surface.
3. Place the control box on its foot.
4. Plug on the robot cable between the robot and the control box.
5. Plug in the mains plug of the control box.

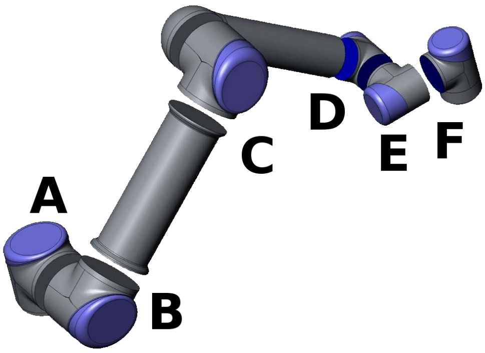


Figure 10.1: Joints of the robot. A: *Base*, B: *Shoulder*, C: *Elbow* and D, E, F: *Wrist 1, 2, 3*

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

Detailed installation instructions can be found in the Hardware Installation Manual. Note that a risk assessment is required before using the robot arm to do any work.

10.1.2 Turning the Control Box On and Off

The control box is turned on by pressing the power button at the front side of the panel with the touch screen. This panel is usually referred to as the *teach pendant*. When the control box is turned on, text from the underlying operating system will appear on the touch screen. After about one minute, a few buttons appear on the screen and a popup guides the user to the initialization screen (see 10.4).

To shut down the control box, press the green power button on the screen, or use the `Shut Down` button on the welcome screen (see 10.3).

WARNING:
Shutting down by pulling the power cord from the wall socket may cause corruption of the robot's file system, which may result in robot malfunction.

10.1.3 Turning the Robot Arm On and Off

The robot arm can be turned on if the control box is turned on, and if no emergency stop button is activated. Turning the robot arm on is done in the initialization screen (see 10.4) by touching the `ON` button on that screen, and then pressing `Start`. When a robot is started, it makes a sound and moves a little while releasing the brakes.

The power to the robot arm can be turned off by touching the `OFF` button on the initialization screen. The robot arm is also powered off automatically when the control box shuts down.

10.1.4 Quick Start

To quickly start up the robot after it has been installed, perform the following steps:

1. Press the Emergency Stop button on the front side of the teach pendant.
2. Press the power button on the teach pendant.
3. Wait a minute while the system is starting up, displaying text on the touch screen.
4. When the system is ready, a popup will be shown on the touch screen, stating that the robot needs to be initialized.
5. Touch the button on the popup dialog. You will be taken to the initialization screen.

6. Wait for the `Confirmation of applied Safety Configuration` dialog and press the `Confirm Safety Configuration` button. This applies an initial set of safety parameters that need to be adjusted based on a risk assessment.
7. Unlock the Emergency Stop button. The robot state changes from `Emergency Stopped` to `Power off`.
8. Step outside the reach (workspace) of the robot.
9. Touch the `On` button on the touch screen. Wait a few seconds until robot state changes to `Idle`.
10. Verify that the payload mass and selected mounting are correct. You will be notified if the mounting detected based on sensor data does not match the selected mounting.
11. Touch the `Start` button on the touch screen. The robot now makes a sound and moves a little while releasing the brakes.
12. Touch the `OK` button, bringing you to the `Welcome` screen.

10.1.5 The First Program

A program is a list of commands telling the robot what to do. PolyScope allows people with only little programming experience to program the robot. For most tasks, programming is done entirely using the touch panel without typing in any cryptic commands.

Since tool motion is an important part of a robot program, a way of teaching the robot how to move is essential. In PolyScope, motions of the tool are given using a series of *waypoints*, i.e. points in the robot's workspace. A waypoint can be given by moving the robot to a certain position, or it can be calculated by software. In order to move the robot arm to a certain position, use either the Move tab (see 12.1), or simply pull the robot arm into place while holding the *Freedrive* button at the back side of the teach pendant.

Besides moving through waypoints, the program can 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.

To create a simple program on a robot that has been started up, do the following:

1. Touch the `Program Robot` button and select `Empty Program`.
2. Touch the `Next` button (bottom right) so that the `<empty>` line is selected in the tree structure on the left side of the screen.
3. Go to the `Structure` tab.
4. Touch the `Move` button.
5. Go to the `Command` tab.
6. Press the `Next` button, to go to the `Waypoint` settings.
7. Press the `Set this waypoint` button next to the `‘‘?’’` picture.
8. On the `Move` screen, move the robot by pressing the various blue arrows, or move the robot by holding the `Freedrive` button, placed on the backside of the teach pendant, while pulling the robot arm.

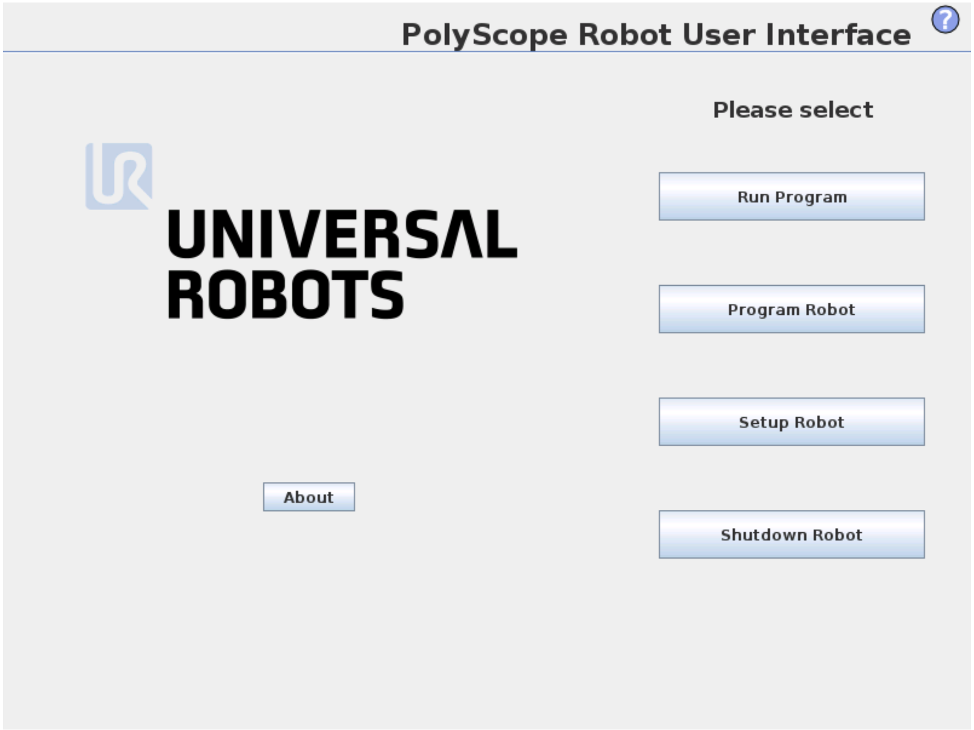
9. Press `OK`.
10. Press `Add waypoint before`.
11. Press the `Set this waypoint` button next to the ``` ``?'' ``` picture.
12. On the `Move` screen, move the robot by pressing the various blue arrows, or move the robot by holding the `Freedrive` button while pulling the robot arm.
13. Press `OK`.with
14. Your program is ready. The robot will move between the two points when you press the “Play” symbol. Stand clear, hold on to the emergency stop button and press “Play”.
15. Congratulations! You have now produced your first robot program that moves the robot between the two given waypoints.

WARNING:

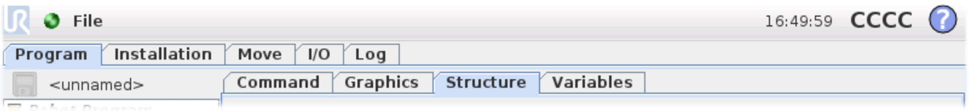
1. Do not drive the robot into itself or anything else as this may cause damage to the robot.
2. Keep your head and torso outside the reach (workspace) of the robot. Do not place fingers where they can be caught.
3. This is only a quick start guide to show how easy it is to use a UR robot. It assumes 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.

10.2 PolyScope Programming Interface

PolyScope runs on the touch sensitive screen attached to the control box.

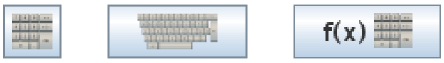

PolyScope Robot User Interface
Please select
UNIVERSAL ROBOTS
Run Program
Program Robot
Setup Robot
About
Shutdown Robot

The picture above shows the Welcome Screen. The bluish areas of the screen are buttons that can be pressed by pressing a finger or the backside of a pen against the screen. PolyScope has a hierarchical structure of screens. In the programming environment, the screens are arranged in *tabs*, for easy access on the screens.


File
16:49:59
CCCC
Program
Installation
Move
I/O
Log
<unnamed>
Command
Graphics
Structure
Variables

In this example, the `Program` tab is selected at the top level, and under that the `Structure` tab is selected. The `Program` tab holds information related to the currently loaded program. If the `Move` tab is selected, the screen changes to the *Move* screen, from where the robot arm can be moved. Similarly, by selecting the `I/O` tab, the current state of the electrical I/O can be monitored and changed.

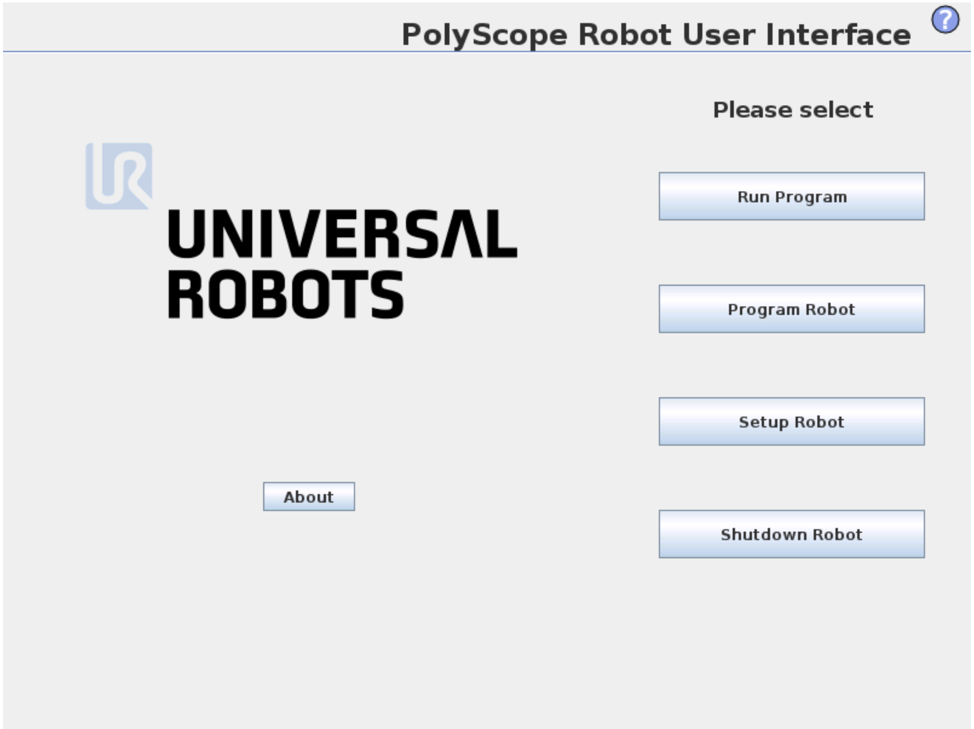
It is possible to connect a mouse and a keyboard to the control box or the teach pendant; however, this is not required. Almost all text fields are touch-enabled, so touching them launches an on-screen keypad or keyboard. Non-touchable text fields have an editor icon next to them that launches the associated input editor.



The icons of the on-screen keypad, keyboard and expression editor are shown above.

The various screens of PolyScope are described in the following sections.

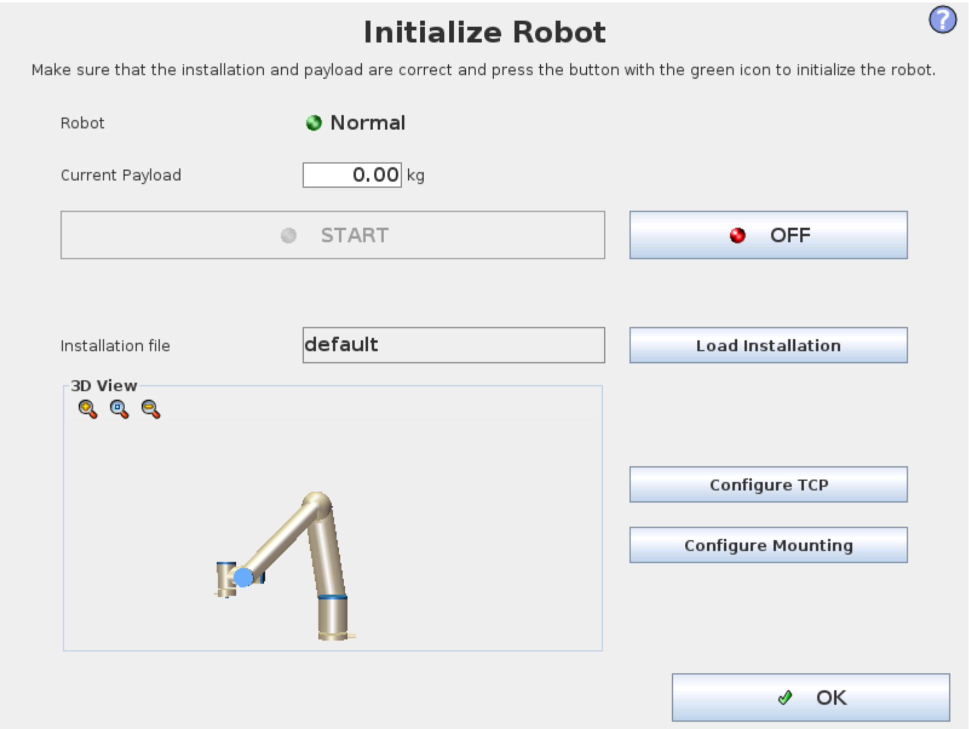
10.3 Welcome Screen


PolyScope Robot User Interface
Please select
UNIVERSAL ROBOTS
Run Program
Program Robot
Setup Robot
About
Shutdown Robot

After booting up the controller PC, the welcome screen is shown. The screen offers the following options:

- **Run Program:** Choose and run an existing program. This is the simplest way to operate the robot arm and control box.
- **Program Robot:** Change a program, or create a new program.
- **Setup Robot:** Set passwords, upgrade software, request support, calibrate the touch screen, etc.
- **Shutdown Robot:** Powers off the robot arm and shuts down the control box.

10.4 Initialization Screen



Initialize Robot

Make sure that the installation and payload are correct and press the button with the green icon to initialize the robot.

Robot ● Normal

Current Payload 0.00 kg

Installation file default

On this screen you control the initialization of the robot arm.

Robot arm state indicator

The status LED gives an indicaton of the robot arm's running state:

- A bright red LED indicates that the robot arm is currently in a stopped state where the reasons can be several.
- A bright yellow LED indicates that the robot arm is powered on, but is not ready for normal operation.
- Finally, a green LED indicates that the robot arm is powered on, and ready for normal operation.

The text appearing next to the LED further specifies the current state of the robot arm.

Active payload and installation

When the robot arm is powered on, the payload mass used by the controller when operating the robot arm is shown in the small white text field. This value can be modified by tapping the text field and entering a new value. Note that setting this value does not modify the payload in the robot's installation (see 12.6), it only sets the payload mass to be used by the controller.

Similarly, the name of the installation file that is currently loaded is shown in the grey text field. A different installation can be loaded by tapping the text field or by using the `Load` button next to it. Alternatively, the loaded installation can be customized using the buttons next to the 3D view in the lower part of the screen.

Before starting up the robot arm, it is very important to verify that both the active payload and the active installation correspond to the actual situation the robot arm is currently in.

Initializing the robot arm

DANGER:
Always verify that the actual payload and installation are correct before starting up the robot arm. If these settings are wrong, the robot arm and control box will not function correctly and may become dangerous to people or equipment around them.

CAUTION:
Great care should be taken if the robot arm is touching an obstacle or table, since driving the robot arm into the obstacle might damage a joint gearbox.

The large button with the green icon on it serves to perform the actual initialization of the robot arm. The text on it, and the action it performs, change depending on the current state of the robot arm.

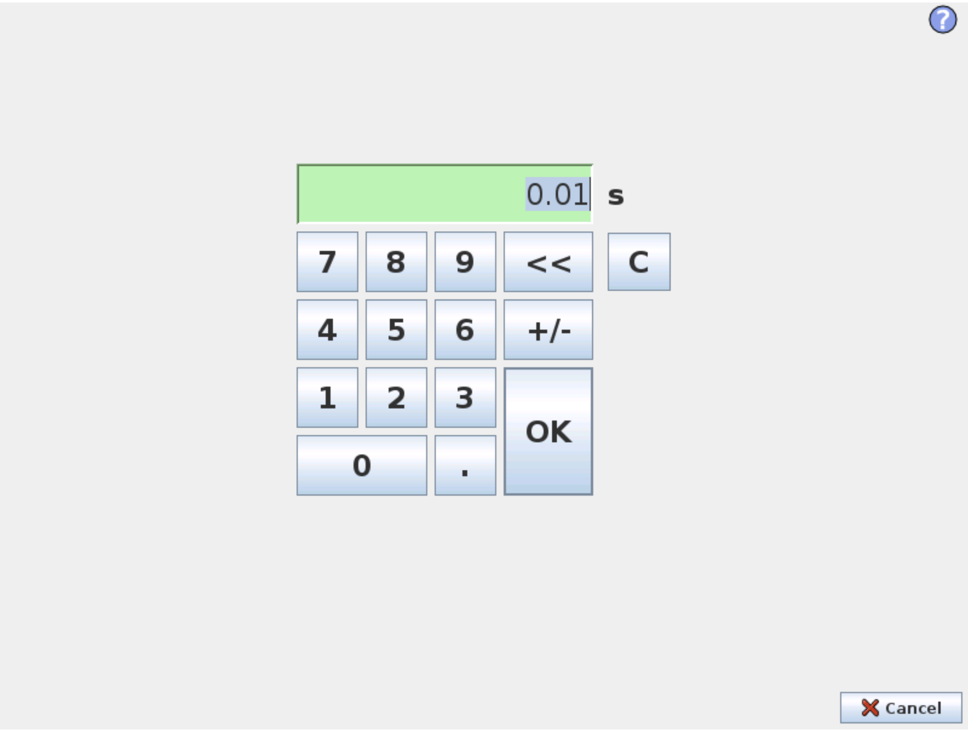
- After the controller PC boots up, the button needs to be tapped once to power the robot arm on. The robot arm state then turns to *Power on* and subsequently to *Idle*. Note that when an emergency stop is in place, the robot arm cannot be powered on, so the button will be disabled.
- When the robot arm state is *Idle*, the button needs to be tapped once again to start the robot arm up. 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 verification passes, tapping the button releases all joint brakes and the robot arm becomes ready for normal operation. Note that the robot makes a sound and moves a little while releasing the brakes.
- If the robot arm violates one of the safety limits after it starts up, it operates in a special *Recovery mode*. In this mode, tapping the button switches to a recovery move screen where the robot arm can be moved back within the safety limits.
- If a fault occurs, the controller can be restarted using the button.
- If the controller is currently not running, tapping the button starts it.

Finally, the smaller button with the red icon on it serves to power off the robot arm.

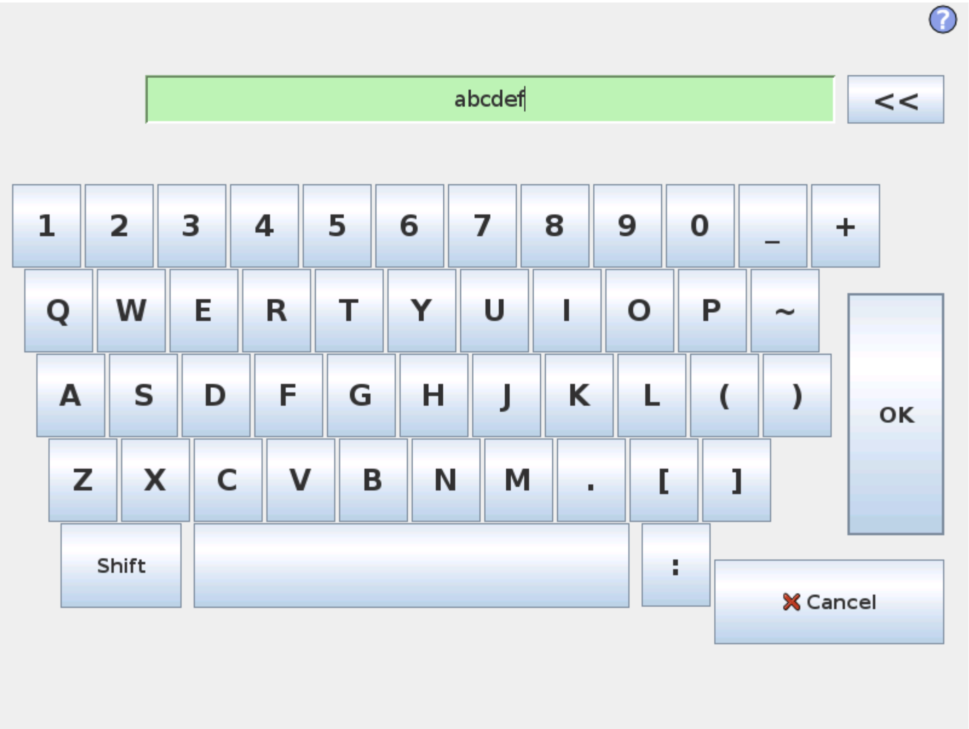
11 On-screen Editors

11.1 On-screen Keypad


0.01
s
7
8
9
<<
C
4
5
6
+/-
1
2
3
OK
0
.
Cancel

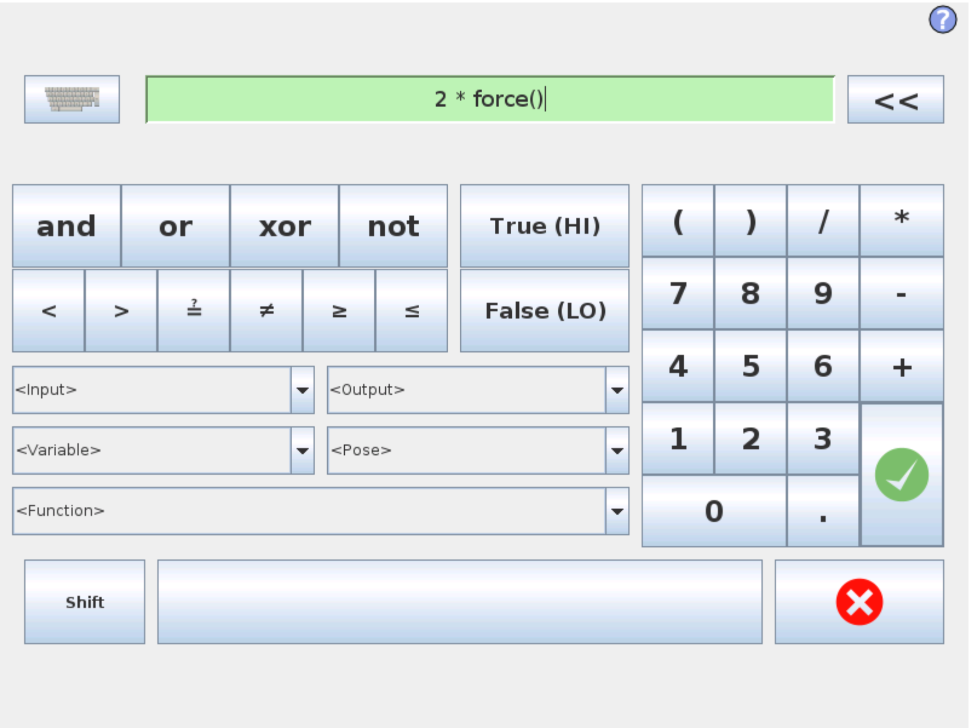
Simple number typing and editing facilities. In many cases, the unit of the typed value is displayed next to the number.

11.2 On-screen Keyboard


abcdef
<<
1 2 3 4 5 6 7 8 9 0 _ +
Q W E R T Y U I O P ~
A S D F G H J K L ( )
Z X C V B N M . [ ]
Shift
:
OK
Cancel

Simple text typing and editing facilities. The `Shift` key can be used to get some additional special characters.

11.3 On-screen Expression Editor


2 * force()
<<
and
or
xor
not
True (HI)
False (LO)
<
>
≟
≠
≥
≤
<Input>
<Output>
<Variable>
<Pose>
<Function>
( ) / *
7 8 9 -
4 5 6 +
1 2 3
0 .
Shift

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 $\le$ for less than or equal to. The keyboard symbol button in the top right 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`.

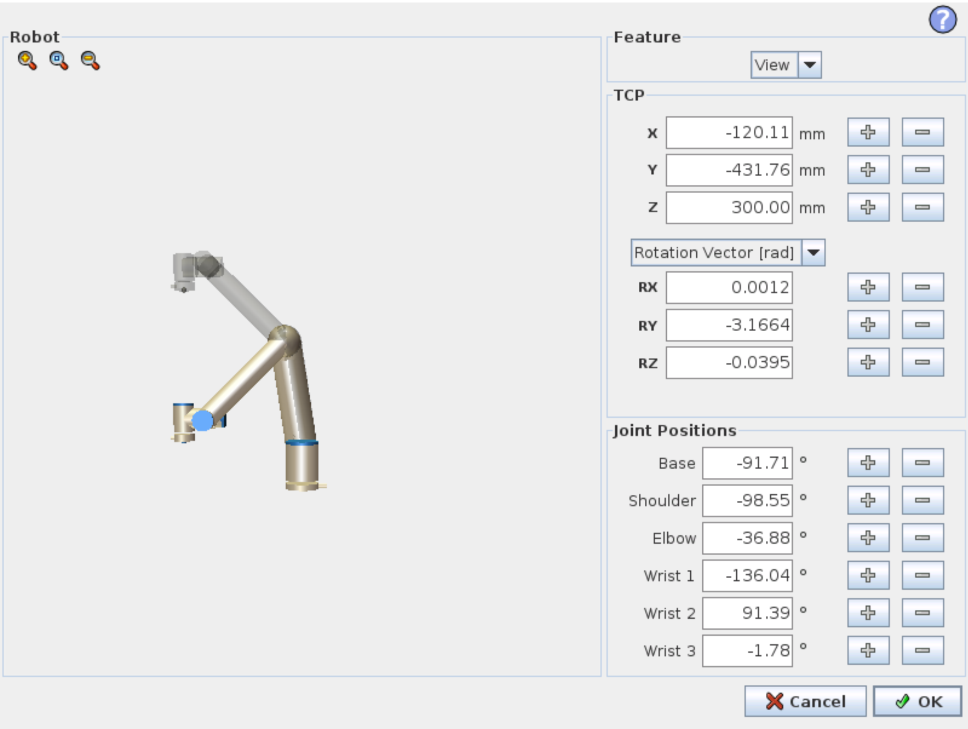
The expression is checked for grammatical errors when the `Ok` button is pressed. The `Cancel` button leaves the screen, discarding all changes.

An expression can look like this:

digital_in[1]$\overset{?}{=}$True and analog_in[0]<0.5

11.4 Pose Editor Screen

On this screen you can specify target joint positions, or a target pose (position and orientation) of the robot tool. This screen is "offline" and does not control the robot arm directly.



Feature: View

X: -120.11 mm
Y: -431.76 mm
Z: 300.00 mm

Rotation Vector [rad]

RX: 0.0012
RY: -3.1664
RZ: -0.0395

Base: -91.71 °
Shoulder: -98.55 °
Elbow: -36.88 °
Wrist 1: -136.04 °
Wrist 2: 91.39 °
Wrist 3: -1.78 °

Robot

The current position of the robot arm and the specified new target position are shown in 3D graphics. The 3D drawing of the robot arm shows the current position of the robot arm, and the "shadow" of the robot arm shows the target position of the robot arm controlled by the specified values on the right hand side of the screen. Push the magnifying glass icons to zoom in/out or drag a finger across 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 15.11), 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 15.5) 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 target TCP is in violation or very close to violating a boundary limit, the visualization of the limit turns red.

Feature and tool position

In the top right corner of the screen, the feature selector can be found. The feature selector defines which feature to control the robot arm relative to

Below the feature selector, the name of the currently active Tool Center Point (TCP) is displayed. For further information about configuring several named TCPs, see 12.6. The text boxes show the full coordinate values of that TCP relative to the selected feature. `X`, `Y` and `Z` control the position of the tool, while `RX`, `RY` and `RZ` control the orientation of the tool.

Use the drop down menu above the `RX`, `RY` and `RZ` boxes to choose the orientation representation. Available types are:

- **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:

  $$R_{rpy}(\gamma, \beta, \alpha) = R_Z(\alpha) \cdot R_Y(\beta) \cdot R_X(\gamma)$$
- **RPY `[°]`** *Roll*, *pitch* and *yaw* (*RPY*) angles, where angles are in degrees.

Values can be edited by clicking on the coordinate. Clicking on the + or - buttons just to the right of a box allows you to add or subtract an amount to/from the current value. Pressing and holding down a button will directly increase/decrease the value. The longer the button is down, the larger the increase/decrease will be.

Joint positions

Allows the individual joint positions to be specified directly. Each joint position can have a value in the range from $-360°$ to $+360°$, which are the *joint limits*. Values can be edited by clicking on the joint position. Clicking on the + or - buttons just to the right of a box allows you to add or subtract an amount to/from the current value. Pressing and holding down a button will directly increase/decrease the value. The longer the button is down, the larger the increase/decrease will be.

OK button

If this screen was activated from the `Move` tab (see 12.1), clicking the `OK` button will return to the `Move` tab, where the robot arm will move to the specified target. If the last specified value was a tool coordinate, the robot arm will move to the target position using the *MoveL* movement type, while the robot arm will move to the target position using the *MoveJ* movement type, if a joint position was specified last. The different movement types are described in 13.5.

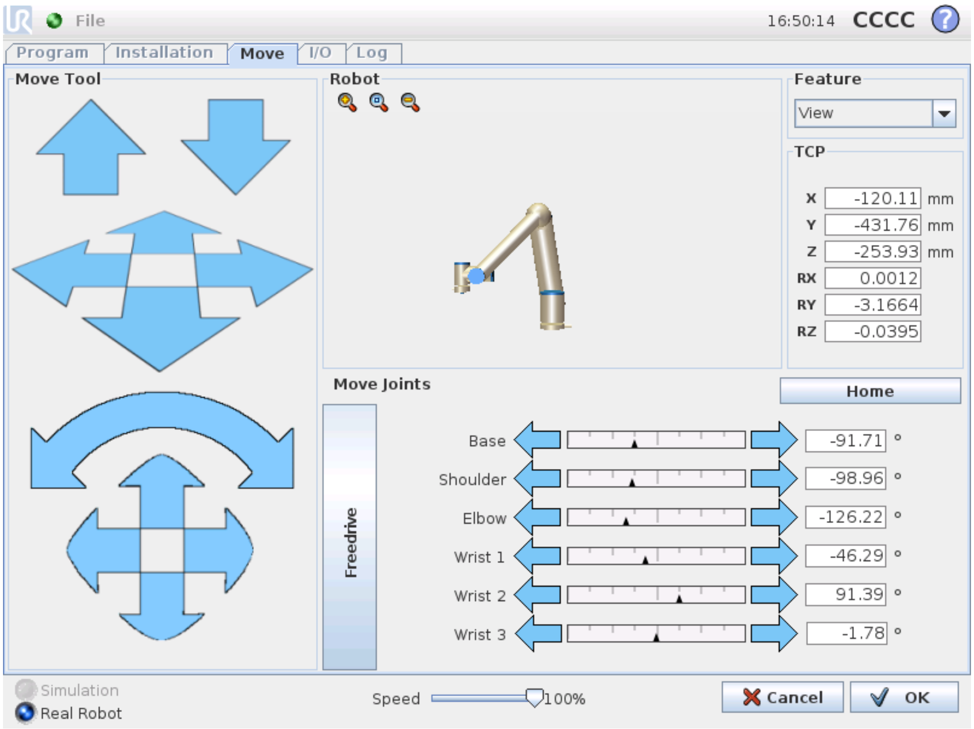
Cancel button

Clicking the `Cancel` button leaves the screen discarding all changes.

12 Robot Control

12.1 Move Tab

On this screen you can always move (jog) the robot arm directly, either by translating/rotating the robot tool, or by moving robot joints individually.



Feature: View

X -120.11 mm
Y -431.76 mm
Z -253.93 mm
RX 0.0012
RY -3.1664
RZ -0.0395

Base -91.71 °
Shoulder -98.96 °
Elbow -126.22 °
Wrist 1 -46.29 °
Wrist 2 91.39 °
Wrist 3 -1.78 °

Speed 100%

12.1.1 Robot

The current position of the robot arm is shown in 3D graphics. Push the magnifying glass icons to zoom in/out or drag a finger across to change the view. To get the best feel for controlling the robot arm, select the `View` feature and rotate the viewing angle of the 3D drawing to match your view of the real robot arm.

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 15.11), a 3D representation of the proximate boundary limit is shown. Note that 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 15.5) 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 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.

12.1.2 Feature and Tool Position

In the top right corner of the screen, the feature selector can be found. It defines which feature to control the robot arm relative to.

The name of the currently active Tool Center Point (TCP) is displayed below the feature selector. The text boxes show the full coordinate values of that TCP relative to the selected feature. For further information about configuring several named TCPs, see 12.6.

Values can be edited manually by clicking on the coordinate or the joint position. This will take you to the pose editor screen (see 11.4) where you can specify a target position and orientation for the tool or target joint positions.

12.1.3 Move Tool

- Holding down a `translate arrow` (top) will move the tool-tip of the robot in the direction indicated.
- Holding down a `rotate arrow` (button) will change the orientation of the robot tool in the indicated direction. The point of rotation 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.

Note: *Release the button to stop the motion at any time!*

12.1.4 Move Joints

Allows the individual joints to be controlled directly. Each joint can move from $-360°$ to $+360°$, which are the default *joint limits* illustrated by the horizontal bar for each joint. If a joint reaches its joint limit, it cannot be driven any further. If the limits for a joint have been configured with a position range different from the default (see 15.10), this range is indicated with red in the horizontal bar.

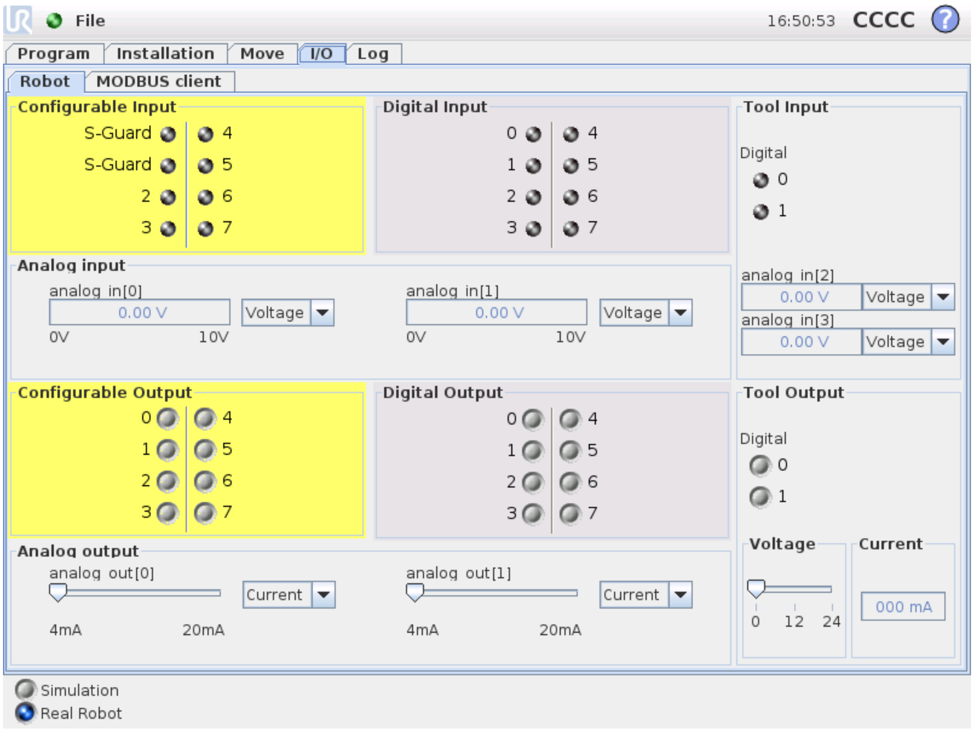
12.1.5 Freedrive

While the *Freedrive* button is held down, it is possible to physically grab the robot arm and pull it to where you want it to be. If the gravity setting (see 12.7) in the `Setup` tab is wrong, or the robot arm carries a heavy load, the robot arm might start moving (falling) when the *Freedrive* button is pressed. In that case, just release the *Freedrive* button again.

WARNING:

1. Make sure to use the correct installation settings (e.g. Robot mounting angle, weight in TCP, TCP offset). Save and load the installation files along with the program.
2. Make sure that the TCP settings and the robot mounting settings are set correctly before operating the `Freedrive` button. If these settings are not correct, the robot arm will move when the `Freedrive` button is activated.
3. The freedrive function (impedance/backdrive) shall only be used in installations where the risk assessment allows it. Tools and obstacles shall not have sharp edges or pinch points. Make sure that all personnel remain outside the reach of the robot arm.

12.2 I/O Tab


File
16:50:53
CCCC
Program
Installation
Move
I/O
Log
Robot
MODBUS client
Configurable Input
S-Guard
S-Guard
2
3
4
5
6
7
Digital Input
0
1
2
3
4
5
6
7
Tool Input
Digital
0
1
Analog input
analog in[0]
0.00 V
Voltage
0V
10V
analog in[1]
0.00 V
Voltage
0V
10V
analog in[2]
0.00 V
Voltage
analog in[3]
0.00 V
Voltage
Configurable Output
0
1
2
3
4
5
6
7
Digital Output
0
1
2
3
4
5
6
7
Tool Output
Digital
0
1
Voltage
0
12
24
Current
000 mA
Analog output
analog out[0]
Current
4mA
20mA
analog out[1]
Current
4mA
20mA
Simulation
Real 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, inluding 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.

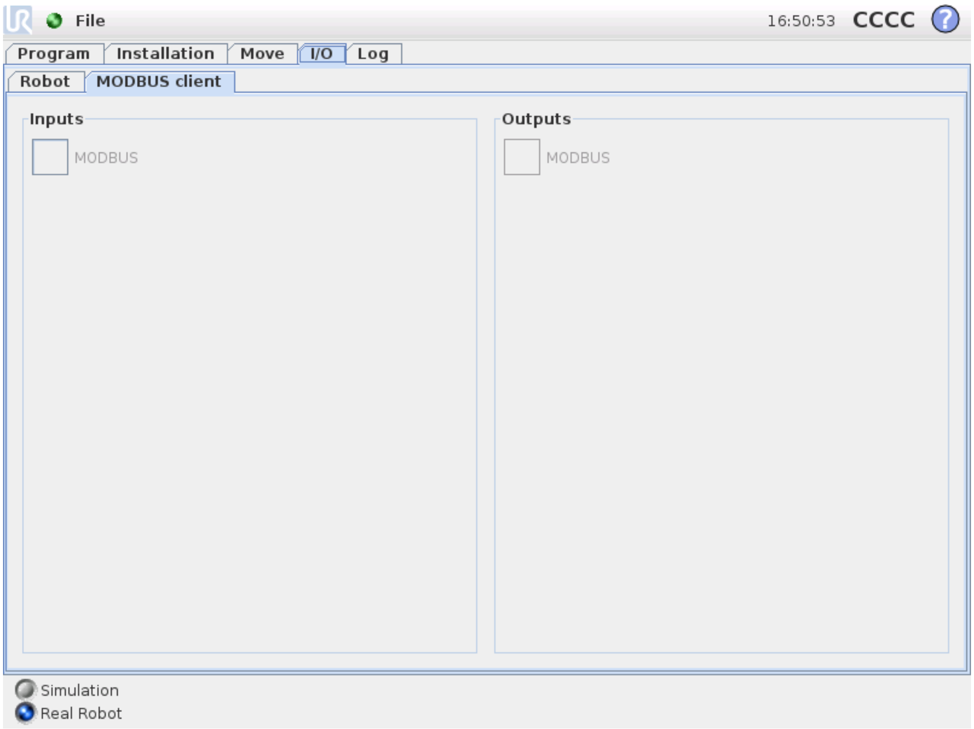
Configurable I/O's can be reserved for special safety settings defined in the safety I/O configuration section of the installaton (see 15.12); those which are reserved will have the name of the safety function in place of the default or user defined name. Configurable outputs that are reserved for safety settings are not togglable and will be displaed as LED's only.

The electrical details of the signals are described in the user manual.

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.

12.3 MODBUS client I/O

Here, the digital MODBUS client I/O signals as set up in the installation are shown. If the signal connection is lost, the corresponding entry on this screen is disabled.


File
16:50:53
CCCC
Program
Installation
Move
I/O
Log
Robot
MODBUS client
Inputs
☐ MODBUS
Outputs
☐ MODBUS
Simulation
Real Robot

Inputs

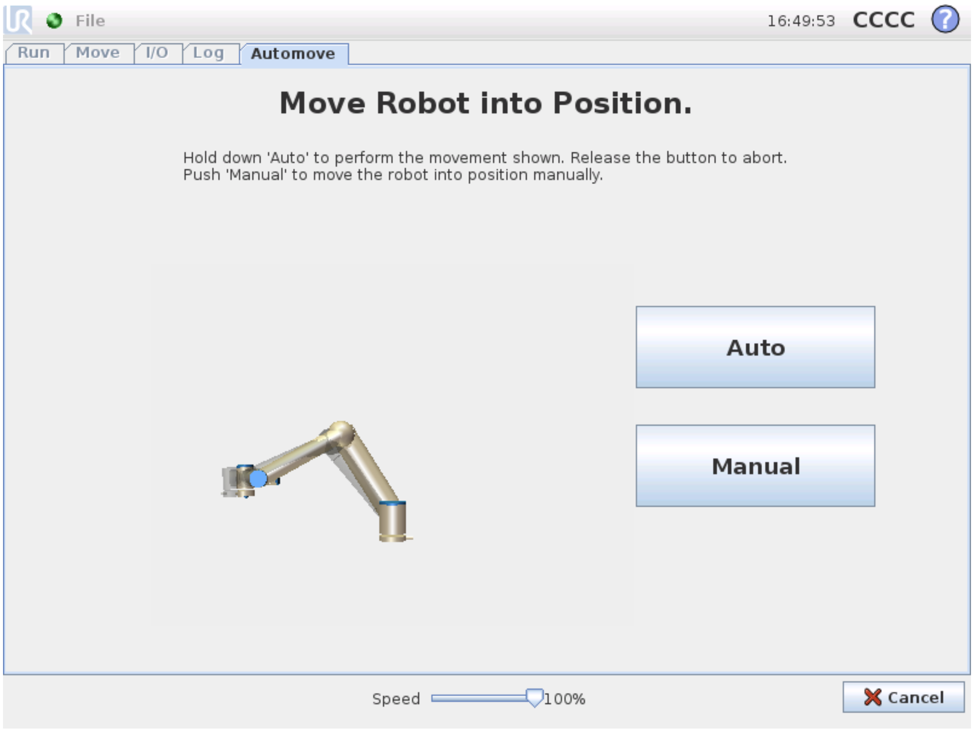
View the state of digital MODBUS client inputs.

Outputs

View and toggle the state of digital MODBUS client outputs. A signal can only be toggled if the choice for I/O tab control (described in 12.8) allows it.

12.4 AutoMove Tab

The AutoMove tab is used when the robot arm has to move to a specific position in its workspace. Examples are when the robot arm has to move to the start position of a program before running it, or when moving to a waypoint while modifying a program.



Move Robot into Position.

Hold down 'Auto' to perform the movement shown. Release the button to abort.
Push 'Manual' to move the robot into position manually.

Animation

The animation shows the movement the robot arm is about to perform.

CAUTION:
Compare the animation with the position of the real robot arm and make sure that the robot arm can safely perform the movement without hitting any obstacles.

CAUTION:
The automove function moves along the robot along the shadow trajectory. Collision might damage the robot or other equipment.

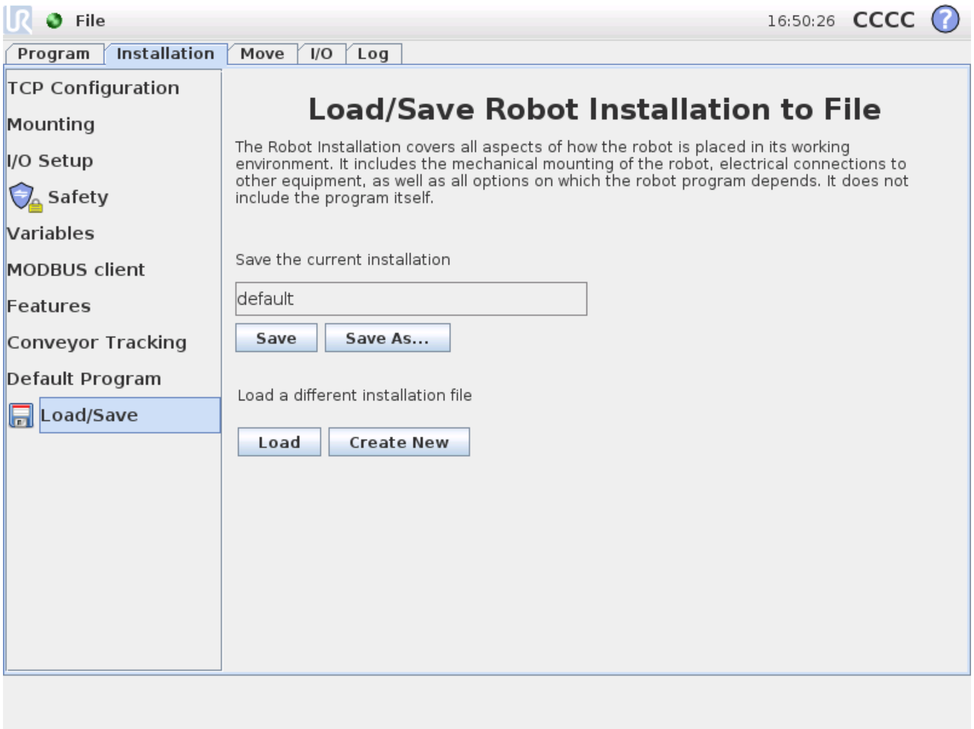
Auto

Hold down the `Auto` button to move the robot arm as shown in the animation. Note: *Release the button to stop the motion at any time!*

Manual

Pushing the `Manual` button will take you to the `MoveTab` where the robot arm can be moved manually. This is only needed if the movement in the animation is not preferable.

12.5 Installation → Load/Save



Load/Save Robot Installation to File

The Robot Installation covers all aspects of how the robot is placed in its working environment. It includes the mechanical mounting of the robot, electrical connections to other equipment, as well as all options on which the robot program depends. It does not include the program itself.

Save the current installation

default

Load a different installation file

The Robot Installation covers all aspects of how the robot arm and control box are placed in the working environment. It includes the mechanical mounting of the robot arm, electrical connections to other equipment, as well as all options on which the robot program depends. It does not include the program itself.

These settings can be set using the various screens under the `Installation` tab, except for the I/O domains which are set in the `I/O` tab (see 12.2).

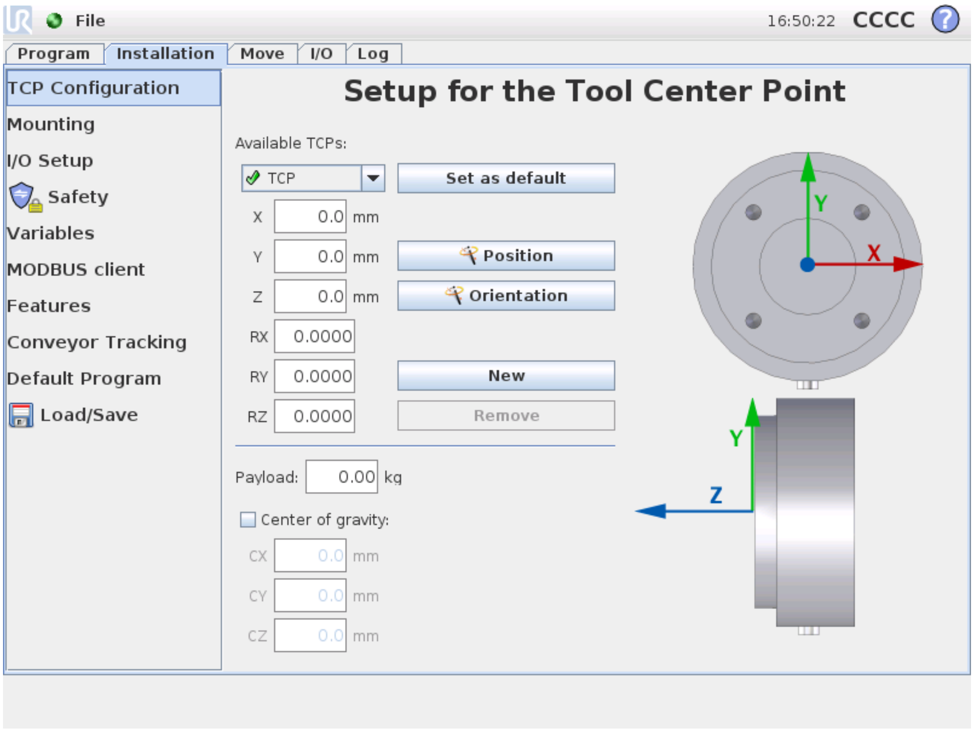
It is possible to have more than one installation file for the robot. Programs created will use the active installation, and will load this installation automatically when used.

Any changes to an installation need to be saved to be preserved after power down. If there are unsaved changes in the installation, a floppy disk icon is shown next to the `Load/Save` text on the left side of the `Installation` tab.

Saving an installation can be done by pressing the `Save` or `Save As...` button. Alternatively, saving a program also saves the active installation. To load a different installation file, use the `Load` button. The `Create New` button resets all of the settings in the Robot Installation to their factory defaults.

CAUTION:
Using the robot with an installation loaded from a USB drive is not recommended. To use an installation stored on a USB drive, first load it and then save it in the local `programs` folder using the `Save As...` button.

12.6 Installation → TCP Configuration


File
16:50:22
CCCC
Program
Installation
Move
I/O
Log
TCP Configuration
Mounting
I/O Setup
Safety
Variables
MODBUS client
Features
Conveyor Tracking
Default Program
Load/Save
Set as default
Position
Orientation
New
Remove
Y
X
Y
Z

Setup for the Tool Center Point

Available TCPs: TCP

X 0.0 mm

Y 0.0 mm

Z 0.0 mm

RX 0.0000

RY 0.0000

RZ 0.0000

Payload: 0.00 kg

☐ Center of gravity:

CX 0.0 mm

CY 0.0 mm

CZ 0.0 mm

A *Tool Center Point* (TCP) is a characteristic point on the robot's tool. Several named TCPs can be defined on this screen. Each TCP contains a translation and a rotation relative to the center of the tool output flange, as indicated on the on-screen graphics. The position coordinates, `X`, `Y` and `Z`, specify the position of the TCP, while `RX`, `RY` and `RZ` specify its orientation. When all of the specified values are zero, the TCP coincides with the center point on the tool output flange and adopts the coordinate system depicted on the right side of the screen.

12.6.1 Adding, modifying and removing TCPs

To define a new TCP, hit the `New` button. The created TCP then automatically receives a unique name and becomes selected in the drop-down menu.

The translation and rotation of the selected TCP can be modified by tapping the respective white text fields and entering new values.

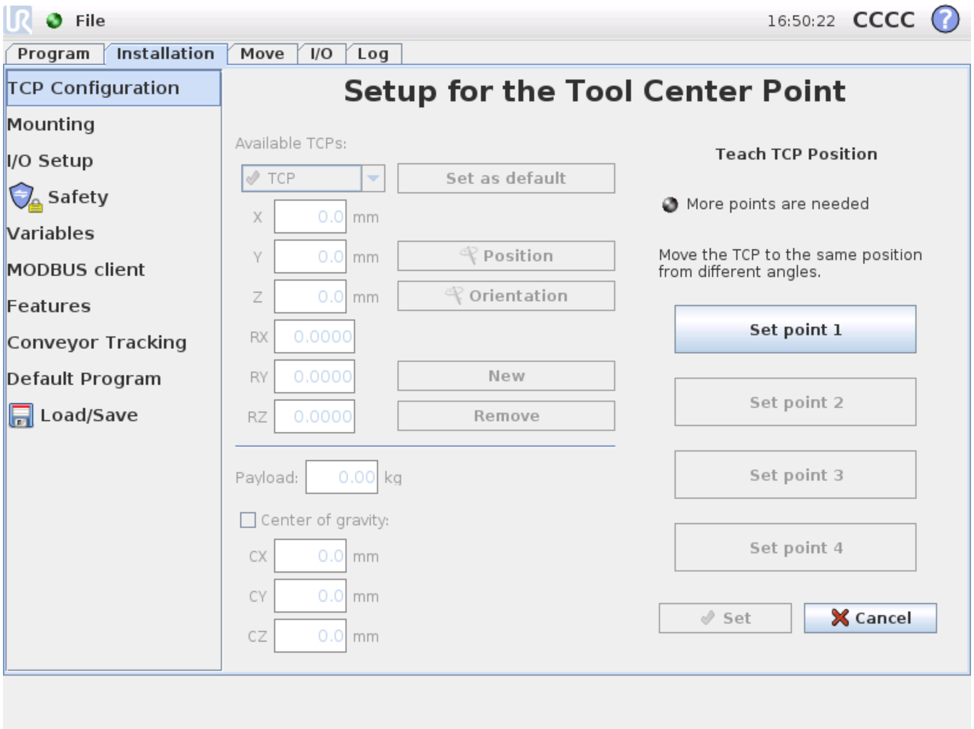
To remove the selected TCP, simply tap the `Remove` button. The last remaining TCP cannot be deleted.

12.6.2 The default and the active TCP

Precisely one of the configured TCPs is the *default* one. The default TCP is marked by a green icon to the left of its name in the drop-down TCP menu. To set the currently selected TCP as the default one, hit the `Set as default` button.

One TCP offset is always used as the *active* one to determine all linear motions in Cartesian space. Also, it is the motion of the active TCP that is visualized on the Graphics tab (see 13.28). Before any program is run, as well as before the start of a program, the default TCP is set as the active one. Within a program, any of the specified TCPs can be set as the active one for a particular movement of the robot (see 13.5 and 13.10).

12.6.3 Teaching TCP position


File
16:50:22
CCCC
Program
Installation
Move
I/O
Log
TCP Configuration
Mounting
I/O Setup
Safety
Variables
MODBUS client
Features
Conveyor Tracking
Default Program
Load/Save
Setup for the Tool Center Point
Set as default
Position
Orientation
New
Remove
Teach TCP Position
Set point 1
Set point 2
Set point 3
Set point 4
Set
Cancel

Available TCPs: TCP

X 0.0 mm
Y 0.0 mm
Z 0.0 mm
RX 0.0000
RY 0.0000
RZ 0.0000

Payload: 0.00 kg

☐ Center of gravity:

CX 0.0 mm
CY 0.0 mm
CZ 0.0 mm

● More points are needed

Move the TCP to the same position from different angles.

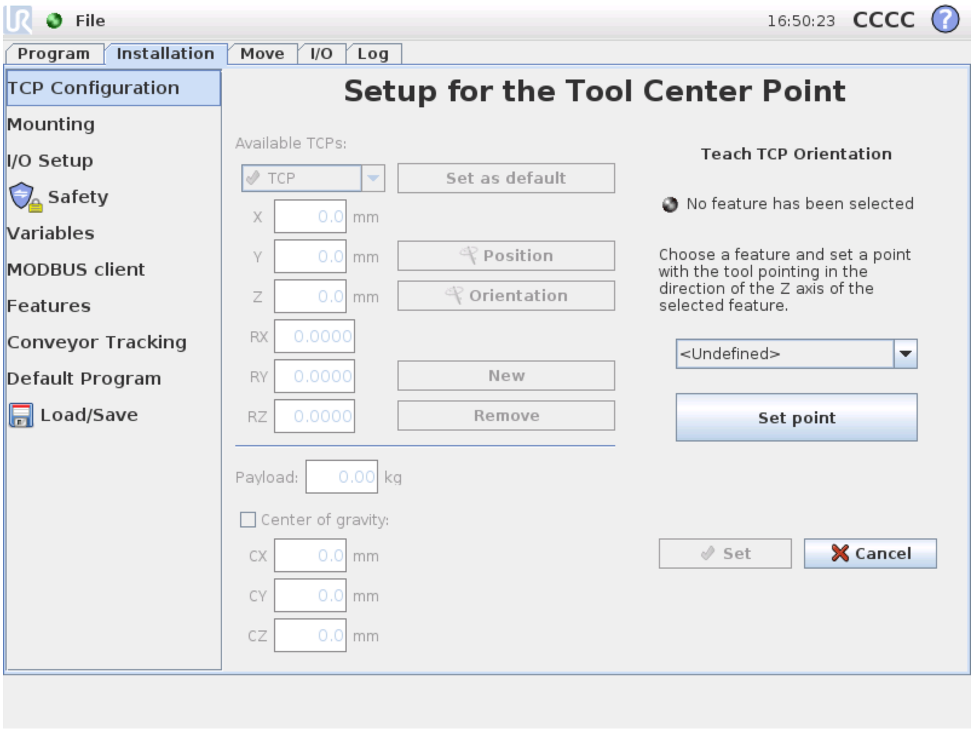
TCP position coordinates can be calculated automatically as follows:

1. Tap the `Position` button.
2. Choose a fixed point in the workspace of the robot.
3. Use the buttons on the right side of the screen to move the TCP to the chosen point from at least three different angles and to save the corresponding positions of the tool output flange.
4. Verify the calculated TCP coordinates and set them onto the selected TCP using the `Set` button.

Note that the positions must be sufficiently diverse for the calculation to work correctly. If they are not, the status LED above the buttons turns red.

Furthermore, even though three positions are usually sufficient to determine the correct TCP, the fourth position can be used to further verify that 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 respective button.

12.6.4 Teaching TCP orientation


File
16:50:23 CCCC
Program
Installation
Move
I/O
Log
TCP Configuration
Mounting
I/O Setup
Safety
Variables
MODBUS client
Features
Conveyor Tracking
Default Program
Load/Save
Setup for the Tool Center Point
Available TCPs:
Set as default
Position
Orientation
New
Remove
Teach TCP Orientation
Set point
Set
Cancel

TCP
X 0.0 mm
Y 0.0 mm
Z 0.0 mm
RX 0.0000
RY 0.0000
RZ 0.0000
Payload: 0.00 kg
☐ Center of gravity:
CX 0.0 mm
CY 0.0 mm
CZ 0.0 mm

No feature has been selected

Choose a feature and set a point with the tool pointing in the direction of the Z axis of the selected feature.

<Undefined>

TCP orientation can be calculated automatically as follows:

1. Tap the `Orientation` button.
2. Select a feature from the drop-down list. For additional information about how new features can be defined, see 12.12.
3. Use the button below to move to a position in which the orientation of the tool corresponding to the TCP coincides with the coordinate system of the selected feature.
4. Verify the calculated TCP orientation and set it onto the selected TCP using the `Set` button.

12.6.5 Payload

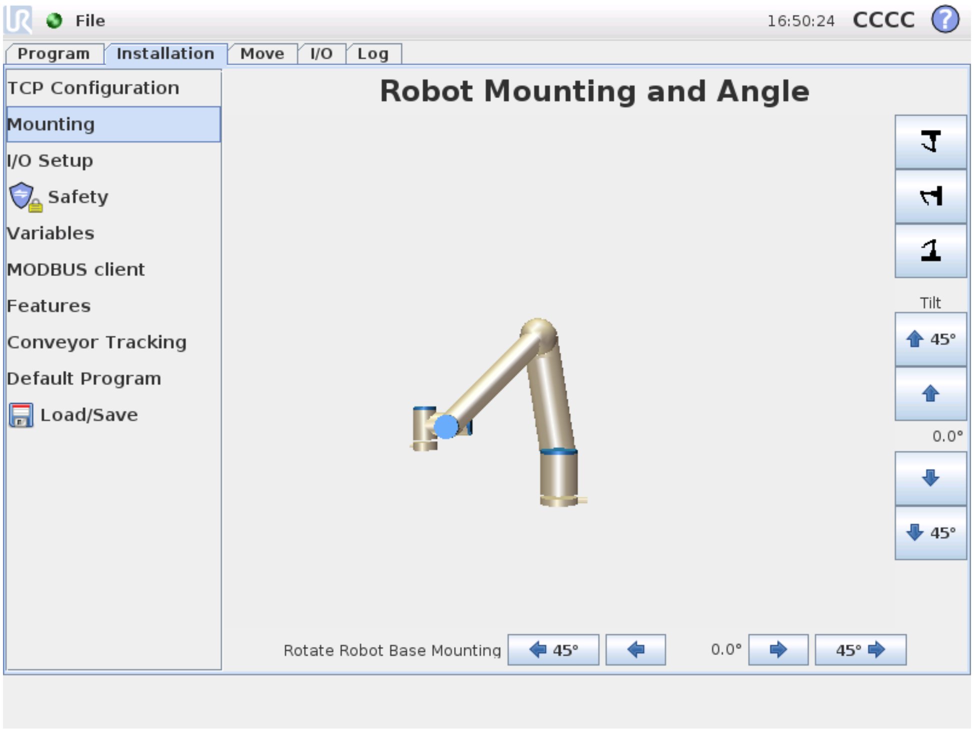
The weight of the tool of the robot is specified in the lower part of the screen. To change this setting, simply tap the white text field and enter a new weight. The setting applies to all defined TCPs.

12.6.6 Center of gravity

The center of gravity of the tool may optionally be specified using the fields `CX`, `CY` and `CZ`. The tool center point is assumed to be the tool's center of gravity if nothing else has been specified. The setting applies to all defined TCPs.

WARNING:
Make sure to use the correct installation settings. Save and load the installation files along with the program.

12.7 Installation → Mounting


File
16:50:24
CCCC
Program
Installation
Move
I/O
Log
TCP Configuration
Mounting
I/O Setup
Safety
Variables
MODBUS client
Features
Conveyor Tracking
Default Program
Load/Save
Robot Mounting and Angle
Tilt
45°
0.0°
45°
Rotate Robot Base Mounting
45°
0.0°
45°

Here the mounting of the robot arm can be specified. This serves two purposes:

1. Making the robot arm look right on the screen.
2. Telling the controller about the direction of gravity.

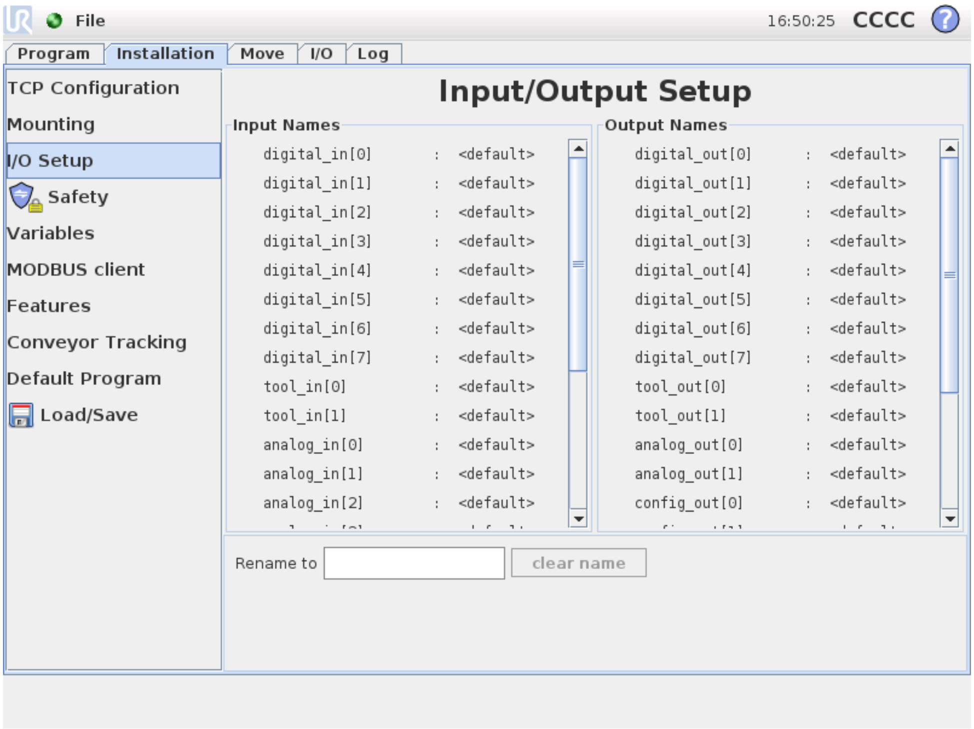
The controller uses an advanced dynamics model to give the robot arm smooth and precise motions, and to make the robot arm hold itself when in *Freedrive* mode. For this reason, it is very important that the mounting of the robot arm be set correctly.

WARNING:
Failure to set robot arm's mounting correctly might result in frequent protective stops, and/or a possibility that the robot arm will move when the freedrive button is pressed.

The default is that the robot arm is mounted on a flat table or floor, in which case 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 push-buttons. The buttons on the right side of the screen are for setting the angle of the robot arm's mounting. The three top right side buttons set the angle to *ceiling* (180°), *wall* (90°), *floor* (0°). The `Tilt` buttons can be used to 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:
Make sure to use the correct installation settings. Save and load the installation files along with the program.

12.8 Installation → I/O Setup


File
16:50:25
CCCC
Program
Installation
Move
I/O
Log
TCP Configuration
Mounting
I/O Setup
Safety
Variables
MODBUS client
Features
Conveyor Tracking
Default Program
Load/Save
Rename to
clear name

Input/Output Setup

Input Names

digital_in[0]	<default>
digital_in[1]	<default>
digital_in[2]	<default>
digital_in[3]	<default>
digital_in[4]	<default>
digital_in[5]	<default>
digital_in[6]	<default>
digital_in[7]	<default>
tool_in[0]	<default>
tool_in[1]	<default>
analog_in[0]	<default>
analog_in[1]	<default>
analog_in[2]	<default>

Output Names

digital_out[0]	<default>
digital_out[1]	<default>
digital_out[2]	<default>
digital_out[3]	<default>
digital_out[4]	<default>
digital_out[5]	<default>
digital_out[6]	<default>
digital_out[7]	<default>
tool_out[0]	<default>
tool_out[1]	<default>
analog_out[0]	<default>
analog_out[1]	<default>
config_out[0]	<default>

Input and output signals can be given names. This can make it easier to remember what the signal does when working with the robot. Select an I/O by clicking on it, and set the name using the on screen keyboard. You can set the name back by setting it to only blank characters.

The eight standard digital inputs and the two tool inputs may configured to trigger an action. Available actions include the ability to start the current program, stop the current program, pause the current program and enter/leave `Freedrive` mode when the input is high/low (like the `Freedrive` button on the back of the Teach Pendant).

The default behavior of outputs is that their values are preserved after a program stops running. It is also possible to configure an output with a default value that is applied whenever no program is running.

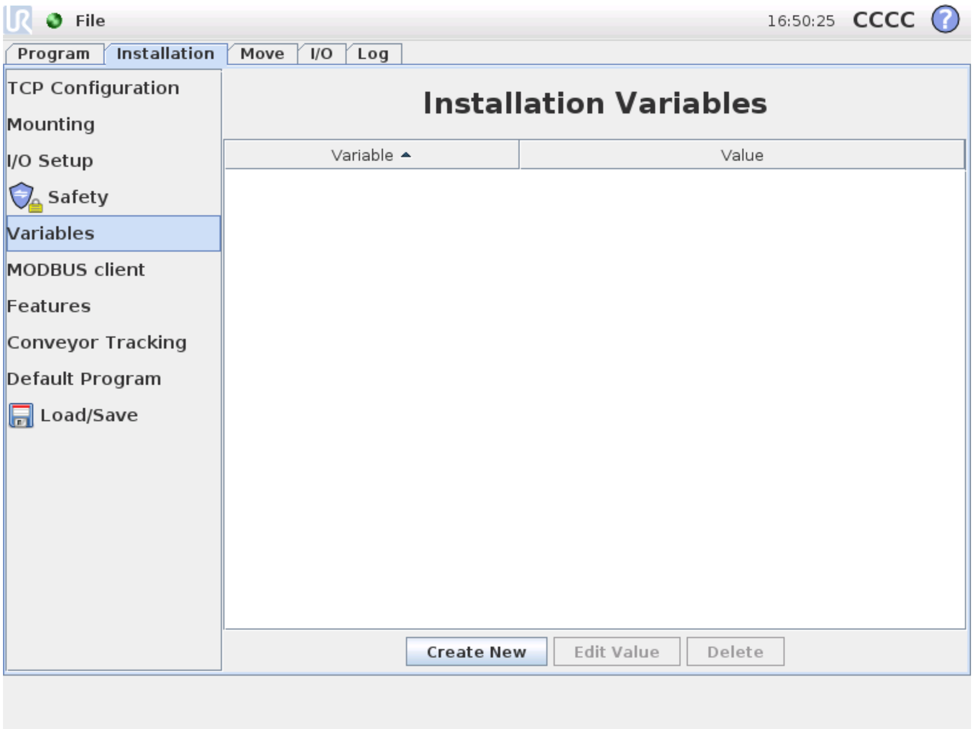
The eight digital standard outputs and the two tool outputs may furthermore be configured to reflect whether a program is currently running, so that the output is high when a program is running and otherwise low.

Finally, it is also possible to specify whether an output can be controlled on the I/O tab (by either programmers, or both operators and programmers) or if it is only robot programs that may alter the output value.

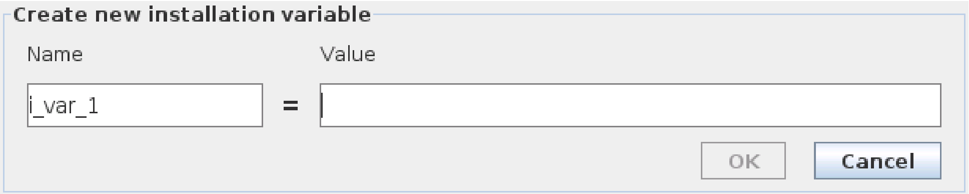
12.9 Installation → Safety

See chapter 15.

12.10 Installation → Variables


File
16:50:25
CCCC
Program
Installation
Move
I/O
Log
TCP Configuration
Mounting
I/O Setup
Safety
Variables
MODBUS client
Features
Conveyor Tracking
Default Program
Load/Save
Installation Variables
Variable
Value
Create New
Edit Value
Delete

Variables created here are called installation variables and can be used just like normal program variables. Installation variables are special because they keep their value even if a program is stopped and then started again, and when the robot arm and/or control box is powered down and powered up again. Their names and values are stored with the installation, so it is possible to use the same variable in multiple programs.


Create new installation variable
Name
Value
=
OK
Cancel

i_var_1

Pressing `Create New` will bring up a panel with a suggested name for the new variable. The name may be changed and its value may be entered by touching either text field. The `OK`-button can only clicked if the new name is unused in this installation.

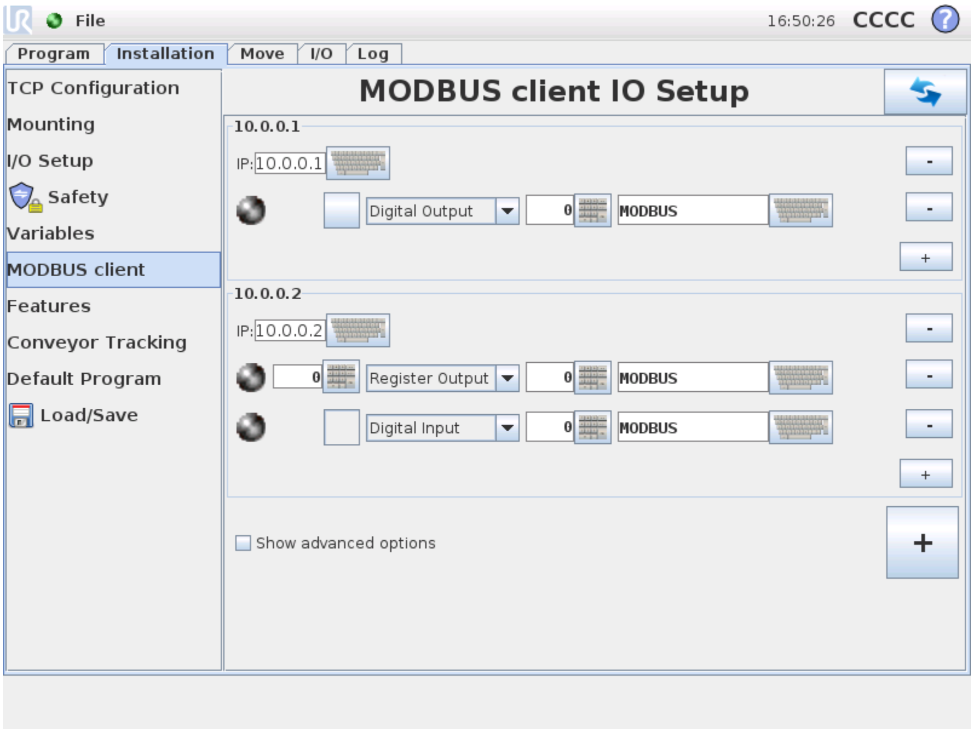
It is possible to change the value of an installation variable by highlighting the variable in the list and then clicking on `Edit Value`.

To delete a variable, select it in the list, then click `Delete`.

After configuring the installation variables, the installation itself must be saved to keep this configuration, see 12.5. The installation variables and their values are also saved automatically every 10 minutes.

If a program or an installation is loaded and one or more of the program variables have the same name as the installation variables, the user is presented with two options to resolve the issue: either use the installation variables of the same name instead of the program variable or have the conflicting variables renamed automatically.

12.11 Installation → MODBUS client I/O Setup


File
16:50:26 CCCC
Program
Installation
Move
I/O
Log
TCP Configuration
Mounting
I/O Setup
Safety
Variables
MODBUS client
Features
Conveyor Tracking
Default Program
Load/Save
MODBUS client IO Setup
-
+
☐ Show advanced options
+

10.0.0.1

IP: 10.0.0.1

Digital Output 0 MODBUS

10.0.0.2

IP: 10.0.0.2

0 Register Output 0 MODBUS

Digital Input 0 MODBUS

Here, the MODBUS client (master) signals can be set up. Connections to MODBUS servers (or slaves) on specified IP addresses can be created with input/output signals (registers or digital). Each signal has a unique name so it can be used in programs.

Refresh

Push this button to refresh all MODBUS connections.

Add unit

Push this button to add a new MODBUS unit.

Delete unit

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

Set unit IP

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

Add signal

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

Delete signal

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

Set signal type

Use this drop down menu to choose the signal type. Available types are:

- **Digital input:** A digital input (coil) is a one-bit quantity which is read from the MODBUS unit on the coil specified in the address field of the signal. Function code 0x02 (Read Discrete Inputs) is used.
- **Digital output:** A digital output (coil) is a one-bit quantity which can be set to either high or low. Before the value of this output has been set by the user, the value is read from the remote MODBUS unit. This means that function code 0x01 (Read Coils) is used. When the output has been set by a robot program or by pressing the “set signal value” button, the function code 0x05 (Write Single Coil) is used onwards.
- **Register input:** A register input is a 16-bit quantity read from the address specified in the address field. The function code 0x04 (Read Input Registers) is used.
- **Register output:** A register output is a 16-bit quantity which can be set by the user. Before the value of the register has been set, the value of it is read from the remote MODBUS unit. This means that function code 0x03 (Read Holding Registers) is used. When the output has been set by a robot program or by specifying a signal value in the “set signal value” field, function code 0x06 (Write Single Register) is used to set the value on the remote MODBUS unit.

Set signal address

This field shows the address on the remote MODBUS server. Use the on-screen keypad to choose a different address. Valid addresses depends on the manufacturer and configuration of the remote MODBUS unit.

Set signal name

Using the on-screen keyboard, the user can give the signal a name. This name is used when the signal is used in programs.

Signal value

Here, the current value of the signal is shown. For register signals, the value is expressed as an unsigned integer. For output signals, the desired signal value can be set using the button. Again, for a register output, the value to write to the unit must be supplied as an unsigned integer.

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 enterd signal value is valid for the specified address on the remote MODBUS server.
- **E4** SLAVE DEVICE FAILURE (0x04): An unrecoverable error occurred while the server (or slave) was attempting to perform the requested action.
- **E5** ACKNOWLEDGE (0x05): Specialized use in conjunction with programming commands sent to the remote MODBUS unit.
- **E6** SLAVE DEVICE BUSY (0x06): Specialized use in conjunction with programming commands sent to the remote MODBUS unit, the slave (server) is not able to respond now.

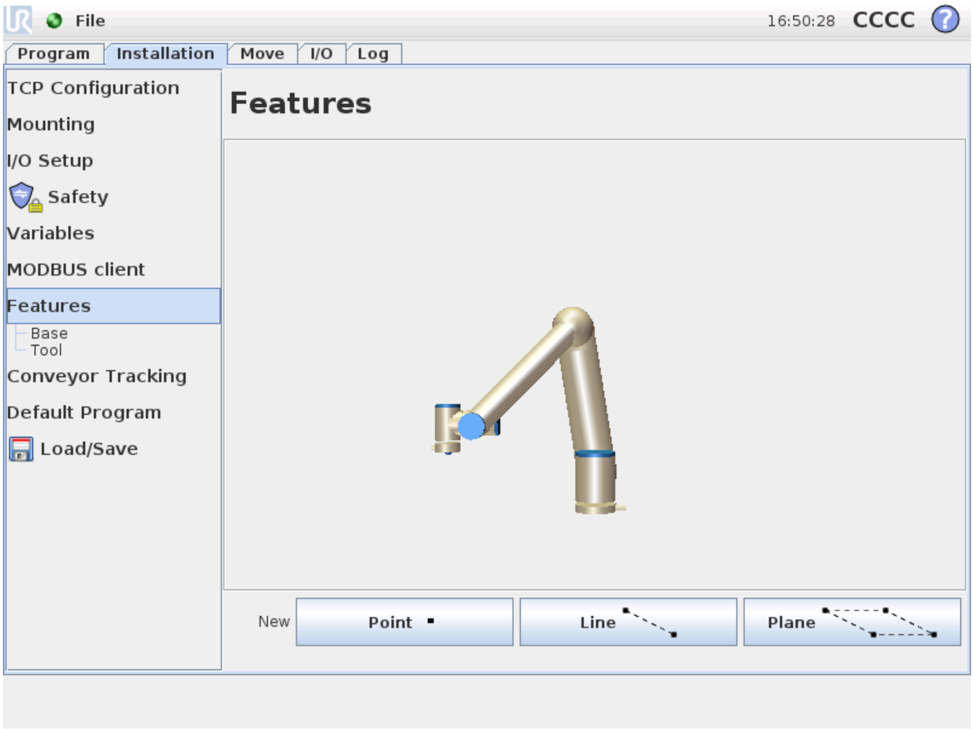
Show Advanced Options

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

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.
- **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.

12.12 Installation → Features


File
16:50:28 CCCC
Program
Installation
Move
I/O
Log
TCP Configuration
Mounting
I/O Setup
Safety
Variables
MODBUS client
Features
Base
Tool
Conveyor Tracking
Default Program
Load/Save
Features
New
Point
Line
Plane

Customers that buy industrial robots generally want to be able to control or manipulate a robot arm, and to program the robot arm, relative to various objects and boundaries in the surroundings of the robot arm, such as machines, objects or blanks, fixtures, conveyers, pallets or vision systems. Traditionally, this is done by defining "frames" (coordinate systems) that relate the internal coordinate system of the robot arm (the base coordinate system) to the relevant object's coordinate system. Reference can both be made to "tool coordinates" and to "base coordinates" of the robot arm.

A problem with such frames is that a certain level of mathematical knowledge is required to be able to define such coordinate systems and also that it takes a considerable ammount of time to do this, even for a person skilled in the art of robot programming and installation. Often this task involves the calculation of 4x4 matrices. Particularly, the representation of orientation is complicated for a person that lacks the required experience to understand this problem.

Questions often asked by customers are for instance:

- Will it be possible to move the robot 4 cm away from the claw of my computerised numerically controlled (CNC) machine?
- Is it possible to rotate the tool of the robot 45 degrees relative to the table?
- Can we make the robot arm move vertically downwards with the object, let the object loose, and then move the robot arm vertically upward again?

The meaning of such and similar questions is very straightforward to an average customer who intends to use a robot arm for instance at various stations in a production plant, and it may seem annoying and incomprehensible to the customer to be told that there may not be a simple answer to such *relevant* questions. There are several complicated reasons for this being the case, and in order to address these problems, Universal Robots has developed unique and simple ways for a customer to specify the location of various objects relative to the robot arm. Within a few steps, it is therefore possible to do exactly what was asked for in the above questions.

Rename

This button makes it possible to rename a feature.

Delete

This button deletes the selected feature and, if any, all sub-features.

Show Axes

Choose whether the coordinate axes of the selected feature shall be visible on the 3D graphics. The choice applies on this screen and on the Move screen.

Joggable

Select whether the selected feature shall be joggable. This determines whether the feature will appear in the feature menu on the Move screen.

Variable

Select whether the selected feature can be used as a variable. If this option is selected a variable named the name of the feature suceeded by "_var" will then be available when editing robot programs, and this variable can be assigned a new value in a program, which can then be used to control waypoints that depend on the value of a feature.

Set or Change Position

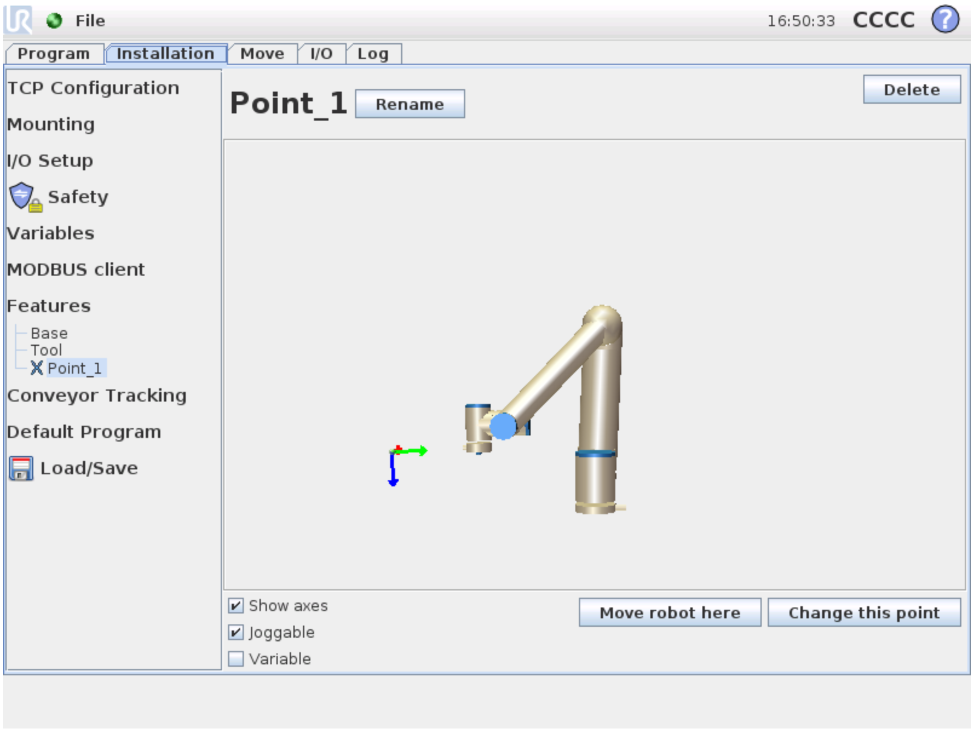
Use this button to set or change the selected feature. The Move screen will appear and a new position of the feature can be set.

Move Robot to Feature

Pressing this button will 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, except for a 180 degree rotation about the x-axis.

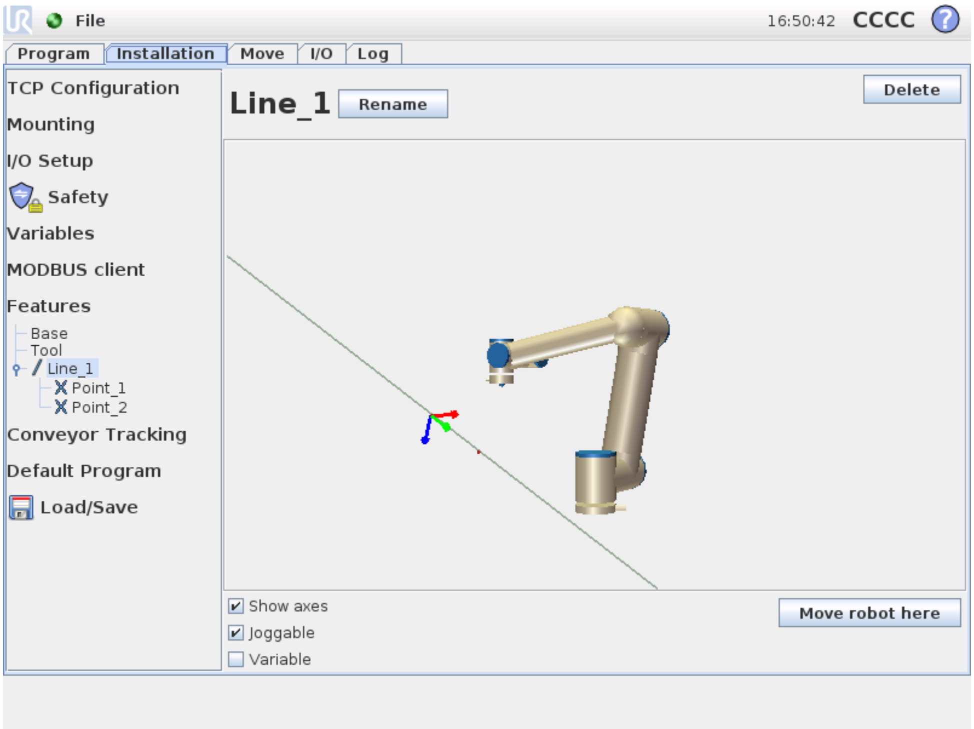
Add Point

Push this button to add a point feature to the installation. The position of a point feature is defined as the position of the TCP at that point. The orientation of the point feature is the same as the TCP orientation, except that the feature coordinate system is rotated 180 degrees about its x-axis. This makes the z-axis of the point feature directed opposite than that of the TCP at that point.


File
16:50:33
CCCC
Program
Installation
Move
I/O
Log
TCP Configuration
Mounting
I/O Setup
Safety
Variables
MODBUS client
Features
Base
Tool
Point_1
Conveyor Tracking
Default Program
Load/Save
Point_1
Rename
Delete
☑ Show axes
☑ Joggable
☐ Variable
Move robot here
Change this point

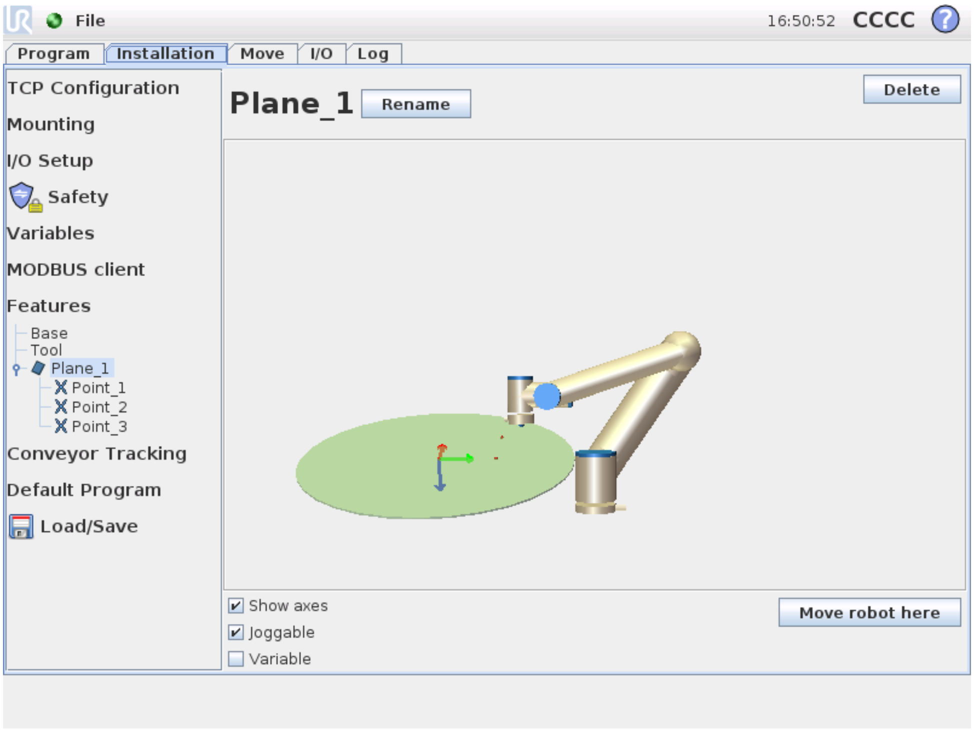
Add Line

Push this button to add a line feature to the installation. A line is defined as an axis between two point features. This axis, directed from the first point towards the second point, will constitute the y-axis of the line coordinate system. The z-axis will be defined by the projection of the z-axis of the first sub point onto the plane perpendicular to the line. The position of the line coordinate system is the same as the position for the first sub point.


File
16:50:42
CCCC
Program
Installation
Move
I/O
Log
TCP Configuration
Mounting
I/O Setup
Safety
Variables
MODBUS client
Features
Base
Tool
Line_1
Point_1
Point_2
Conveyor Tracking
Default Program
Load/Save
Line_1
Rename
Delete
☑ Show axes
☑ Joggable
☐ Variable
Move robot here

Add Plane

Push this button to add a plane feature to the installation. A plane is defined by three sub point features. The position of the coordinate system is the same as the position for the first sub point. The z-axis is the plane normal, and the y-axis is directed from the first point towards the second. The positive direction of the z-axis is set so that the angle between the z-axis of the plane and the z-axis of the first point is less than 180 degrees.


File
16:50:52
CCCC
Program
Installation
Move
I/O
Log
TCP Configuration
Mounting
I/O Setup
Safety
Variables
MODBUS client
Features
Base
Tool
Plane_1
Point_1
Point_2
Point_3
Conveyor Tracking
Default Program
Load/Save
Plane_1
Rename
Delete
Move robot here

☑ Show axes
☑ Joggable
☐ Variable

12.13 Conveyor Tracking Setup

When using a conveyor, the robot can be configured to track the movement of the conveyor. The Conveyor Tracking Setup provides options for configuring the robot to work with absolute and incremental encoders, and linear and circular conveyors.

Conveyor Parameters

`Incremental` encoders can be connected to digital inputs 0 to 3. Using a `Quadrature` encoder (requiring two inputs), the robot is able to determine the speed and direction of the conveyor. If the direction of the coneyor is constant, a single input can be used, counting on `Rising`, `Falling`, or `Rise and Fall` edges to determine the speed of the conveyor.

`Absolute` encoders can be used when connected through a MODBUS signal. This requires that a digital MODBUS input register is preconfigured in the 12.11.

Linear conveyors

When a linear conveyor is selected, a line feature must be configured to determine the direction of the conveyor. The line feature should be parallel to the direction of the conveyor, and there should be a large distance between the two points defining the line feature. It is recommended to configure the line feature by by placing the tool firmly against the side of the conveyor when teaching the two points.

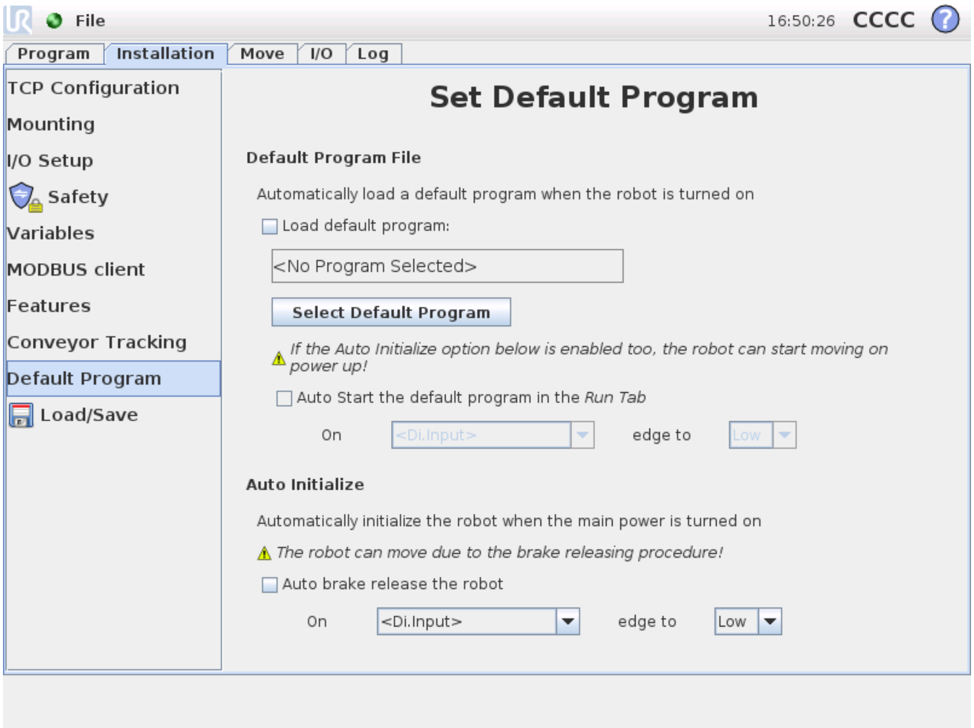
The value of `Ticks per meter` must be the number of ticks the encoder generates when the conveyor moves one meter.

$$\text{Ticks per meter} = \frac{\text{ticks per revolution of encoder}}{2\pi \cdot \text{radius of encoder disc[m]}} \tag{12.1}$$

Circular conveyors

When tracking a circular conveyor, the center point of the conveyor (circle) must be defined. The value of `Ticks per revolution` must be the number of ticks the encoder generates when the conveyor rotates one full revolution.

12.14 Installation → Default Program


File
16:50:26 CCCC
Program
Installation
Move
I/O
Log
TCP Configuration
Mounting
I/O Setup
Safety
Variables
MODBUS client
Features
Conveyor Tracking
Default Program
Load/Save
Select Default Program

Set Default Program

Default Program File

Automatically load a default program when the robot is turned on

☐ Load default program:

<No Program Selected>

⚠ *If the Auto Initialize option below is enabled too, the robot can start moving on power up!*

☐ Auto Start the default program in the *Run Tab*

On <Di.Input> edge to Low

Auto Initialize

Automatically initialize the robot when the main power is turned on

⚠ *The robot can move due to the brake releasing procedure!*

☐ Auto brake release the robot

On <Di.Input> edge to Low

This screen contains settings for automatically loading and starting a default program, and for auto initializing the robot arm on power up.

WARNING:

If auto load, auto start and auto initialize all three are enabled, the robot will start running the selected program as soon as the control box is powered up.

12.14.1 Loading a Default Program

A default program can be chosen to be loaded when the control box is powered up. Furthermore, the default program will also be auto loaded when the *Run Program* screen (see 10.3) is entered and no program is loaded.

12.14.2 Starting a Default Program

The default program can be 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 will be started automatically.

Note, on startup the current input signal level is undefined and chosing a transition that matches the signal level on startup will start the program immediately. Furthermore, leaving the *Run Program* screen or pressing the stop button in the *Dashboard* will disable the auto starting feature until the run button has been pressed again.

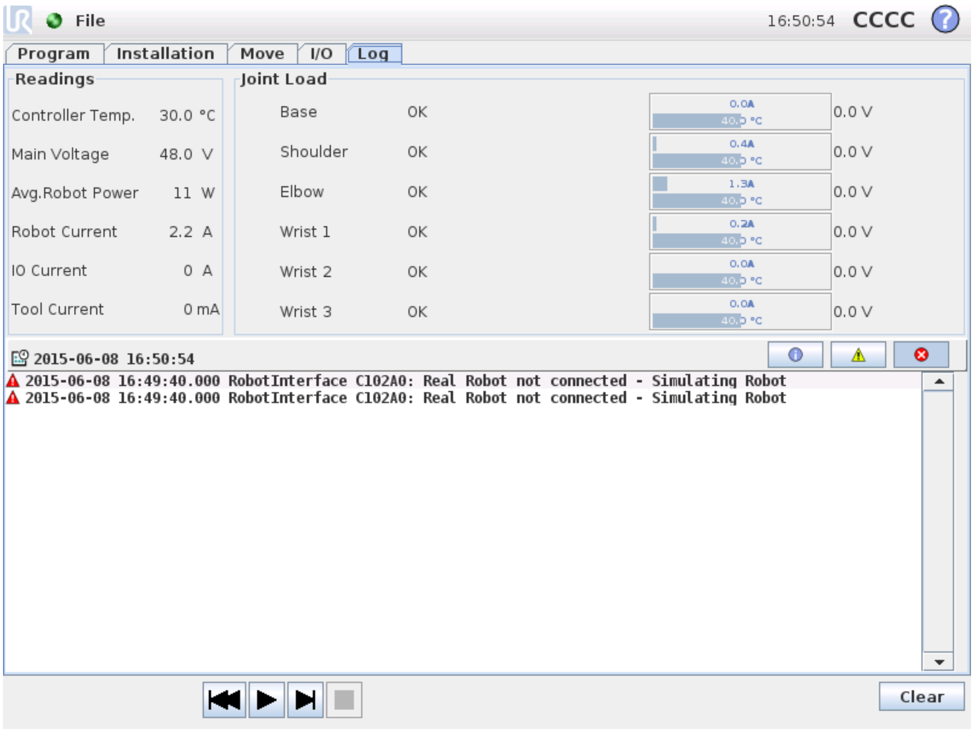
12.14.3 Auto Initialization

The robot arm can be automatically initialized, for instance when the control box is powered up. On the specified external input signal edge transition, the robot arm will be completely initialized, irrespective of the visible screen.

The final stage of initialization is *brake release*. When the robot is releasing the brakes, it moves a bit and makes a sound. Furthermore, the brakes cannot be automatically released if the configured mounting does not match the mounting detected based on sensor data. In this case, the robot needs to be initialized manually in the initialization screen (see 10.4).

Note, on startup the current input signal level is undefined and chosing a transition that matches the signal level on startup will initialize the robot arm immediately.

12.15 Log Tab



Readings

Controller Temp.	30.0 °C
Main Voltage	48.0 V
Avg.Robot Power	11 W
Robot Current	2.2 A
IO Current	0 A
Tool Current	0 mA

Joint Load

Base	OK	0.0A	40.0 °C	0.0 V
Shoulder	OK	0.4A	40.0 °C	0.0 V
Elbow	OK	1.3A	40.0 °C	0.0 V
Wrist 1	OK	0.2A	40.0 °C	0.0 V
Wrist 2	OK	0.0A	40.0 °C	0.0 V
Wrist 3	OK	0.0A	40.0 °C	0.0 V

```
2015-06-08 16:50:54
⚠ 2015-06-08 16:49:40.000 RobotInterface C102A0: Real Robot not connected - Simulating Robot
⚠ 2015-06-08 16:49:40.000 RobotInterface C102A0: Real Robot not connected - Simulating Robot
```

Robot Health The top half of the screen displays the health of the robot arm and control box. The left part shows information related to the control box of the robot, while the right part shows information about each robot joint. Each robot joint shows information for temperaure of the motor and electronics, the load of the joint and the voltage at the joint.

Robot Log On the bottom half of the screen log messages are shown. The first column categorizes the severity of the log entry. The second column shows the time of arrival of the message. The next column shows the sender of the message. While the last column shows the message itself. Messages can be filtered by selecting the toggle buttons which correspond to the severity. The figure above now shows that errors will be displayed while information and warning messages will be filtered. Some log messages are designed to provide more information, this can be accessed by selecting the log entry.

12.16 Load Screen

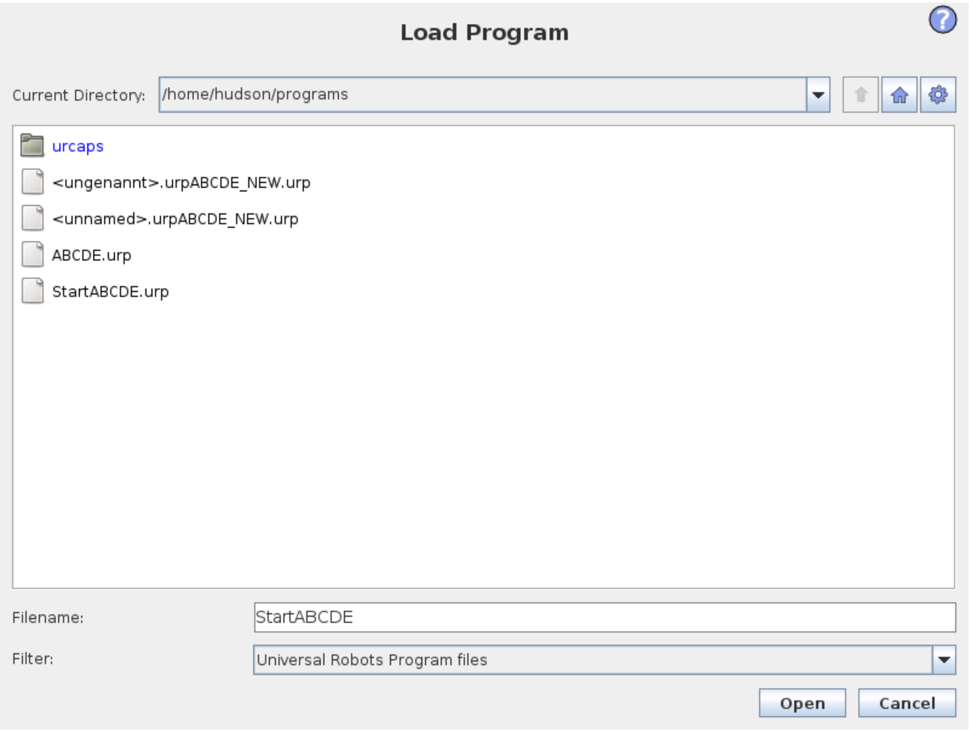
On this screen you choose which program to load. There are two versions of this screen: one that is to be used when you just want to load a program and execute it, and one that is used when you want to actually edit a program.

NOTE:
Running a program from a USB drive is not recommended. To run a program stored on a USB drive, first load it and then save it in the local `programs` folder using the `Save As...` option in the `File` menu.

The main difference lies in which actions are available to the user. In the basic load screen, the user will only be able to access files - not modify or delete them. Furthermore, the user is not allowed to leave the directory structure that descends from the `programs` folder. The user can descend to a sub-directory, but he cannot get any higher than the `programs` folder.

Therefore, all programs should be placed in the programs folder and/or sub folders under the programs folder.

Screen layout



Current Directory: /home/hudson/programs

urcaps
<ungenannt>.urpABCDE_NEW.urp
<unnamed>.urpABCDE_NEW.urp
ABCDE.urp
StartABCDE.urp

Filename: StartABCDE

Filter: Universal Robots Program files

This image shows the actual load screen. It consists of the following important areas and buttons:

Path history The path history shows a list of the paths leading up to the present location. This means that all parent directories up to the root of the computer are shown. Here you will notice that you may not be able to access all the directories above the programs folder.

By selecting a folder name in the list, the load dialog changes to that directory and displays it in the file selection area 12.16.

File selection area In this area of the dialog the contents of the actual area is present. It gives the user the option to select a file by single clicking on its name or to open the file by double clicking on its name.

In the case that the user double-clicks on a directory, the dialog descends into this folder and presents its contents.

File filter By using the file filter, one can limit the files shown to include the type of files that one wishes. By selecting "Backup Files" the file selection area will display the latest 10 saved versions of each program, where `.old0` is the newest and `.old9` is the oldest.

File field Here the currently selected file is shown. The user has the option to manually enter the file name of a file by clicking on the keyboard icon to the right of the field. This will cause an on-screen keyboard to pop up where the user can enter the file name directly on the screen.

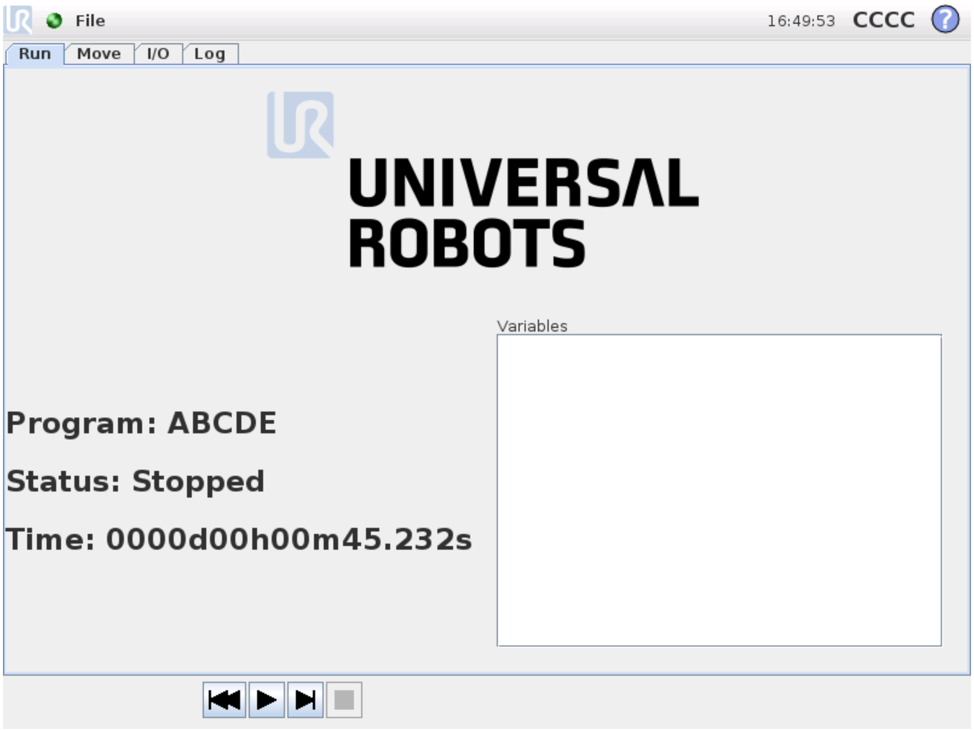
Open button Clicking on the Open button, will open the currently selected file and return to the previous screen.

Cancel button Clicking on the Cancel button will abort the current loading process and cause the screen to switch to the previous image.

Action buttons A series of buttons gives the user the ability to perform some of the actions that normally would be accessible by right-clicking on a file name in a conventional file dialog. Added to this is the ability to move up in the directory structure and directly to the program folder.

- Parent: Move up in the directory structure. The button will not be enabled in two cases: when the current directory is the top directory or if the screen is in the limited mode and the current directory is the program folder.
- Go to program folder: Go home
- Actions: Actions such as create directory, delete file etc.

12.17 Run Tab



Program: ABCDE

Status: Stopped

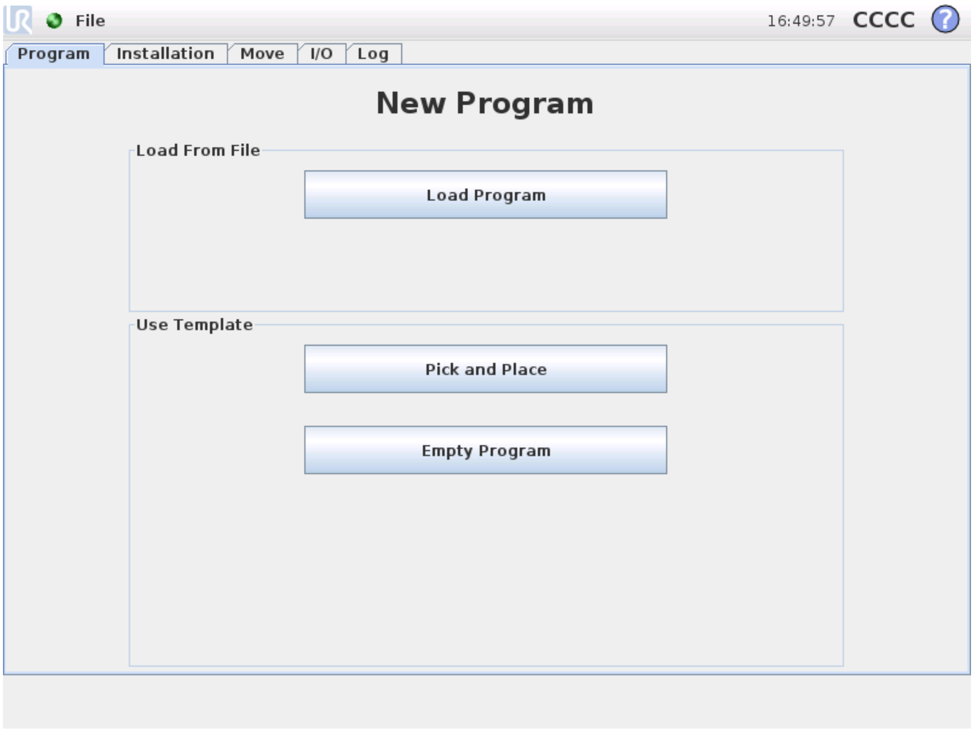
Time: 0000d00h00m45.232s

This tab provides a very simple way of operating the robot arm and control box, with as few buttons and options as possible. This can be usefully combined with password protecting the programming part of PolyScope (see 14.3), to make the robot into a tool that can run exclusively pre-written programs.

Furthermore, in this tab a default program can be automatically loaded and started based on an external input signal edge transition (see 12.14). The combination of auto loading and starting of a default program and auto initialization on power up can, for instance, be used to integrate the robot arm into other machinery.

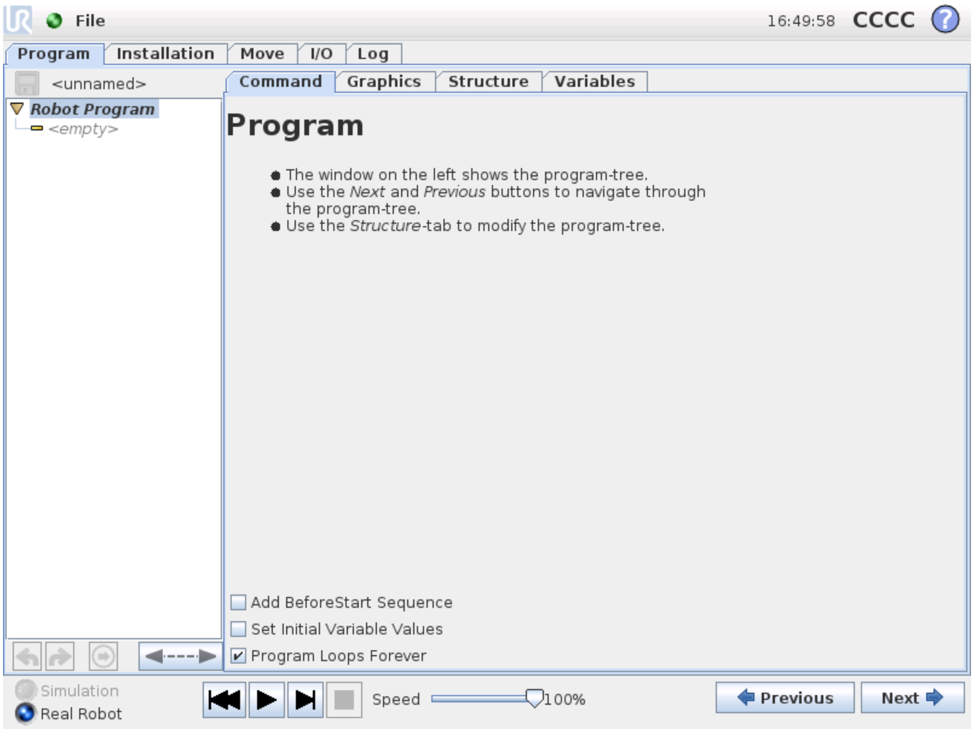
13 Programming

13.1 New Program


File
16:49:57
CCCC
Program
Installation
Move
I/O
Log
New Program
Load From File
Load Program
Use Template
Pick and Place
Empty Program

A new robot program can start from either a *template* or from an existing (saved) robot program. A *template* can provide the overall program structure, so only the details of the program need to be filled in.

13.2 Program Tab


File
16:49:58 CCCC
Program
Installation
Move
I/O
Log
<unnamed>
Command
Graphics
Structure
Variables
Robot Program
<empty>
Simulation
Real Robot
Speed
100%
Previous
Next

Program

- The window on the left shows the program-tree.
- Use the *Next* and *Previous* buttons to navigate through the program-tree.
- Use the *Structure*-tab to modify the program-tree.

☐ Add BeforeStart Sequence
☐ Set Initial Variable Values
☑ Program Loops Forever

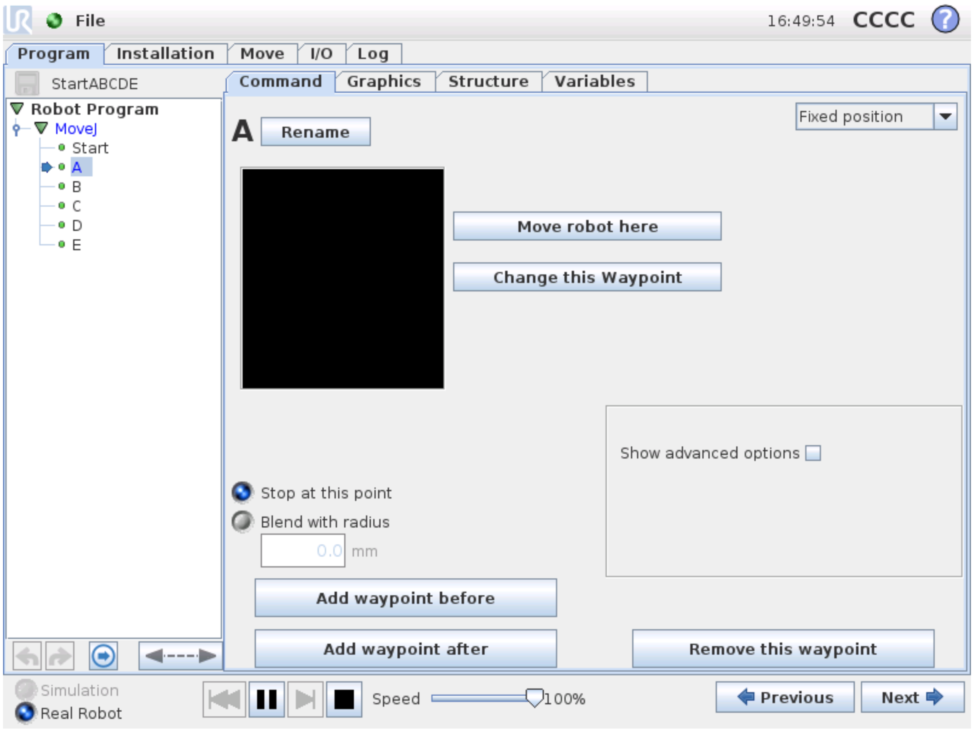
The program tab shows the current program being edited.

13.2.1 Program Tree

The *program tree* on the left side of the screen displays the program as a list of commands, while the area on the right side of the screen displays information relating to the current command. The current command is selected by clicking the command list, or by using the `Previous` and `Next` buttons on the bottom right of the screen. Commands can be inserted or removed using the `Structure` tab, described in 13.29. The program name is shown directly above the command list, with a small disk icon that can be clicked to quickly save the program.

In the program tree, the command that is currently being executed is highlighted as described in 13.2.2.

13.2.2 Program Execution Indication


File
16:49:54 CCCC
Program
Installation
Move
I/O
Log
StartABCDE
Robot Program
MoveJ
Start
A
B
C
D
E
Command
Graphics
Structure
Variables
A
Rename
Fixed position
Move robot here
Change this Waypoint
Show advanced options ☐
● Stop at this point
○ Blend with radius
0.0 mm
Add waypoint before
Add waypoint after
Remove this waypoint
○ Simulation
● Real Robot
Speed
100%
Previous
Next

The program tree contains visual cues informing about the command currently being executed by the robot controller. A small ➡ indicator icon is displayed to the left of the command icon, and the name of the executing command and any commands of which this command is a sub-command (typically identified by the ▷/▽ command icons) are highlighted with blue. This aids the user in locating the executing command in the tree.

For example, if the robot arm is moving towards a waypoint, the corresponding waypoint sub-command is marked with the ➡ icon and its name together with the name of the Move command (see 13.5) to which it belongs to are shown in blue.

If the program is paused, the program execution indicator icon marks the last command that was in the process of being executed.

Clicking the button with the ⊙ icon below the program tree jumps to the current executing or the last executed command in the tree. If a command is clicked while a program is running, the `Command` tab will keep displaying the information related to the selected command. Pressing the ⊙ button will make the `Command` tab continuously show information about the currently executing commands again.

13.2.3 Undo/Redo Buttons

The buttons with icons ↶ and ↷ below the program tree serve to undo and redo changes made in the program tree and in the commands it contains.

13.2.4 Program Dashboard

The lowest part of the screen is the *Dashboard*. The *Dashboard* features a set of buttons similar to an old-fashioned tape recorder, from which programs can be started and stopped, single-stepped and restarted. The *speed slider* allows you to adjust the program speed at any time, which directly affects the speed at which the robot arm moves. Additionally, the *speed slider* shows in real time the relative speed at which the robot arm moves taking into account the safety settings. The indicated percentage is the maximum achievable speed for the running program without violating the safety limits.

To the left of the *Dashboard* the `Simulation` and `Real Robot` buttons toggle between running the program in a simulation, or running it on the real robot. When running in simulation, the robot arm does not move and thus cannot damage itself or any nearby equipment in collisions. Use simulation to test programs if unsure about what the robot arm will do.

DANGER:

1. Make sure to stay outside the robot workspace when the `Play` button is pressed. The movement you programmed may be different than expected.
2. Make sure to stay outside the robot workspace when the `Step` button is pressed. The function of the `Step` button can be difficult to understand. Only use it when it is absolutely necessary.
3. Make sure to always test your program by reducing the speed with the speed slider. Logic programming errors made by the integrator might cause unexpected movements of the robot arm.
4. When a emergency stop or protective stop has occured, the robot program will stop. It can be resumed as long as no joint has moved more than 10°. When pressing play, the robot will move slowly back onto the trajectory, and continue program execution.

While the program is being written, the resulting motion of the robot arm is illustrated using a 3D drawing on the `Graphics` tab, described in 13.28.

Next to each program command is a small icon, which is either red, yellow or green. A red icon means that there is an error in that command, yellow means that the command is not finished, and green means that all is OK. A program can only be run when all commands are green.

13.3 Variables

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

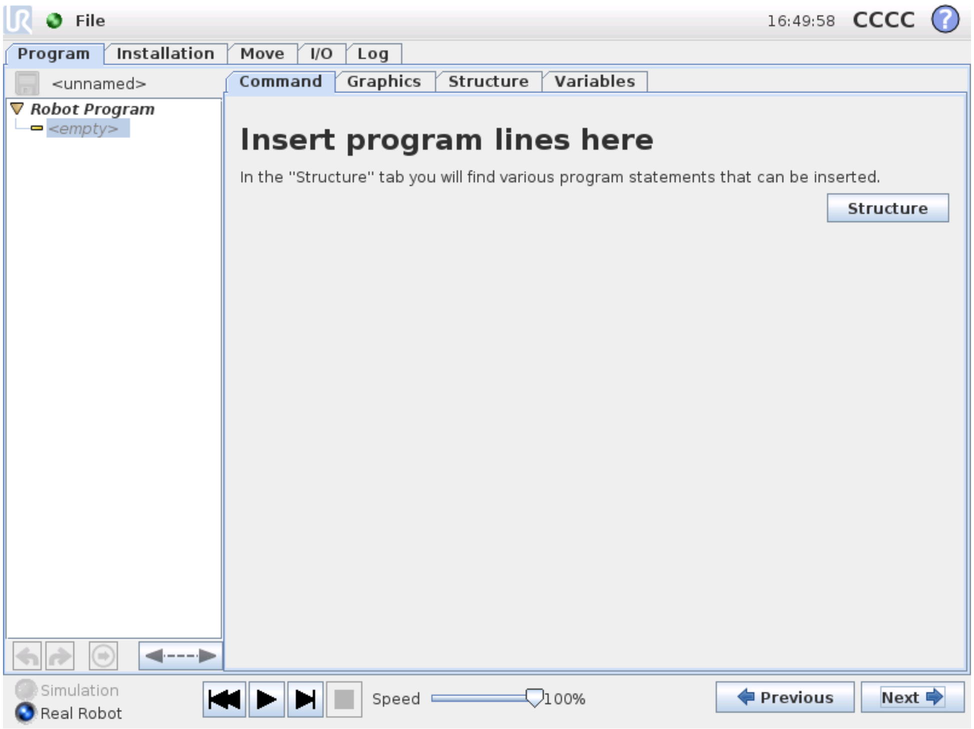
Installation variables: These can be used by multiple programs and their names and values are persisted together with the robot installation (see 12.10 for further details);

Regular program variables: These are available to the running program only and their values are lost as soon as the program is stopped.

The following variable types are available:

bool	A boolean variable whose value is either `True` or `False`.
int	A whole number in the range from -32768 to 32767.
float	A floating point number (decimal).
string	A sequence of characters.
pose	A vector describing the location and orientation in Cartesian space. It is a combination of a position vector (x, y, z) and a rotation vector (rx, ry, rz) representing the orientation, written `p[x, y, z, rx, ry, rz]`.
list	A sequence of variables.

13.4 Command: Empty



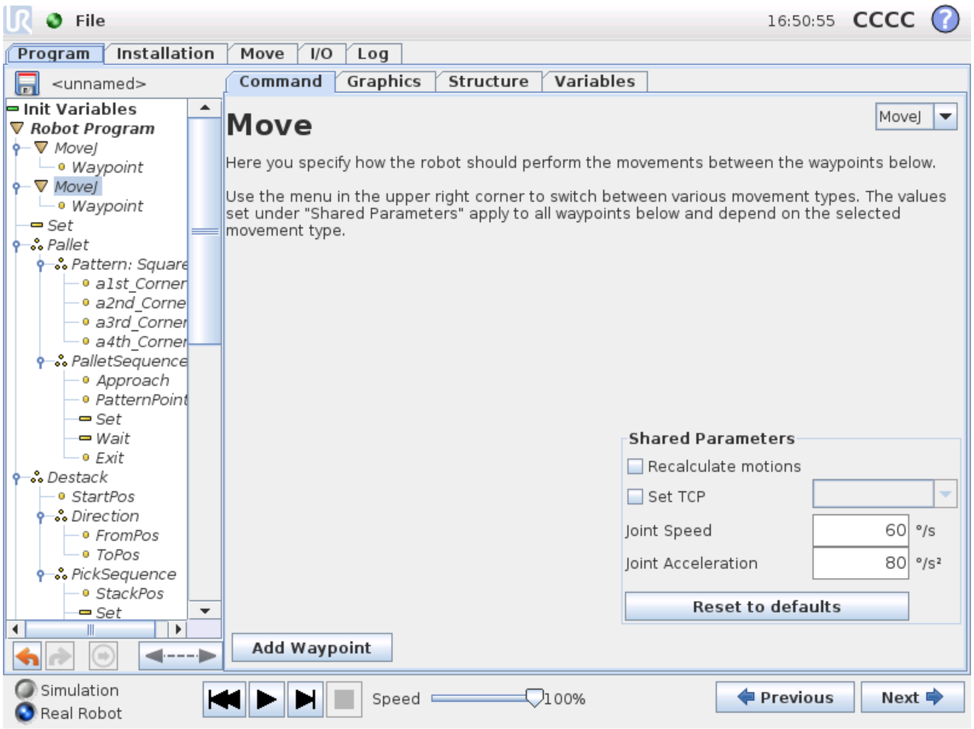
Insert program lines here

In the "Structure" tab you will find various program statements that can be inserted.

File
16:49:58
CCCC
Program
Installation
Move
I/O
Log
<unnamed>
Command
Graphics
Structure
Variables
Robot Program
<empty>
Structure
Simulation
Real Robot
Speed
100%
Previous
Next

Program commands need to be inserted here. Press the `Structure` button to go to the structure tab, where the various selectable program lines can be found. A program cannot run before all lines are specified and defined.

13.5 Command: Move



File
16:50:55 CCCC
Program
Installation
Move
I/O
Log
<unnamed>
Command
Graphics
Structure
Variables
Init Variables
Robot Program
MoveJ
Waypoint
MoveJ
Waypoint
Set
Pallet
Pattern: Square
a1st_Corner
a2nd_Corne
a3rd_Corner
a4th_Corner
PalletSequence
Approach
PatternPoint
Set
Wait
Exit
Destack
StartPos
Direction
FromPos
ToPos
PickSequence
StackPos
Set
MoveJ
Add Waypoint
Reset to defaults
Simulation
Real Robot
Speed
100%
Previous
Next

Move

Here you specify how the robot should perform the movements between the waypoints below.

Use the menu in the upper right corner to switch between various movement types. The values set under "Shared Parameters" apply to all waypoints below and depend on the selected movement type.

Shared Parameters

☐ Recalculate motions

☐ Set TCP

Joint Speed 60 °/s

Joint Acceleration 80 °/s²

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

It is possible to select one of three types of movements: *MoveJ*, *MoveL* and *MoveP* each explained below.

- **moveJ** will make movements that are calculated in the *joint space* of the robot arm. Each joint is controlled to reach the desired end location 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 to use for the movement calculations, 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 favorable choice.
- **moveL** will make the tool move 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. The selected feature will determine in which feature space the tool positions of the waypoints are represented in. Of specific interest concerning feature spaces are variable features and variable waypoints. Variable features can be used when the tool position of a waypoint need to be

determined by the actual value of the variable feature when the robot program runs.

- **moveP** will move 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.

  A **Circle Move** can be added to a moveP command, consisting of two waypoints: the first one specifying a via point on the circular arc, and the second one being the endpoint of the movement. The robot will start the circle movement from its current position, and then move through the two specified waypoints. The orientation change of the tool through the circle move is determined only by the starting orientation and the orientation at the endpoint, so the orientation of the via point does not influence the circle move. A Circle Move must always be preceeded by a waypoint under the same moveP.

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.

Recalculate motions

Tick this check box if the positions within this move command should be adjusted based on the active TCP.

TCP selection

The TCP used for the waypoints under this Move command can be selected by ticking the check box and selecting a TCP from the drop-down menu. The selected TCP is then set as active each time the robot arm moves to one of the Waypoints under this Move command. If the check box is not ticked, then the active TCP is not modified in any way. If the active TCP for this motion is determined during runtime of the program, it needs to be set dynamically using the Set command (see 13.10) or by using script commands. For further information about configuring named TCPs, see 12.6.

Feature selection

For *MoveL* and *MoveP*, it is possible to select which feature space the waypoints under the Move command should be represented when specifying these waypoints. 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:

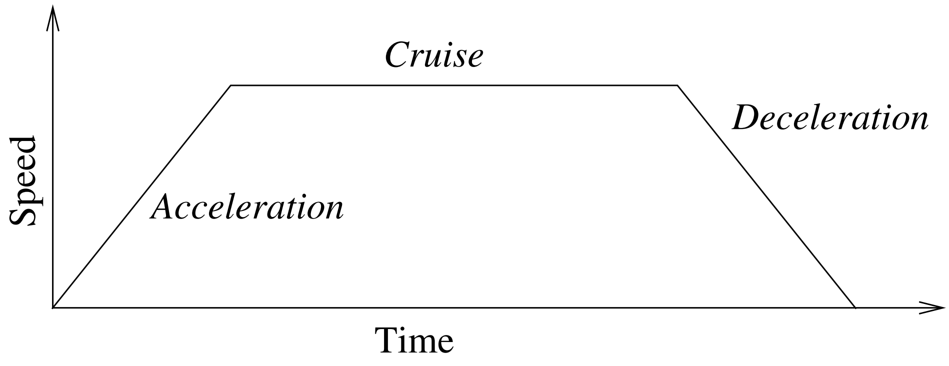


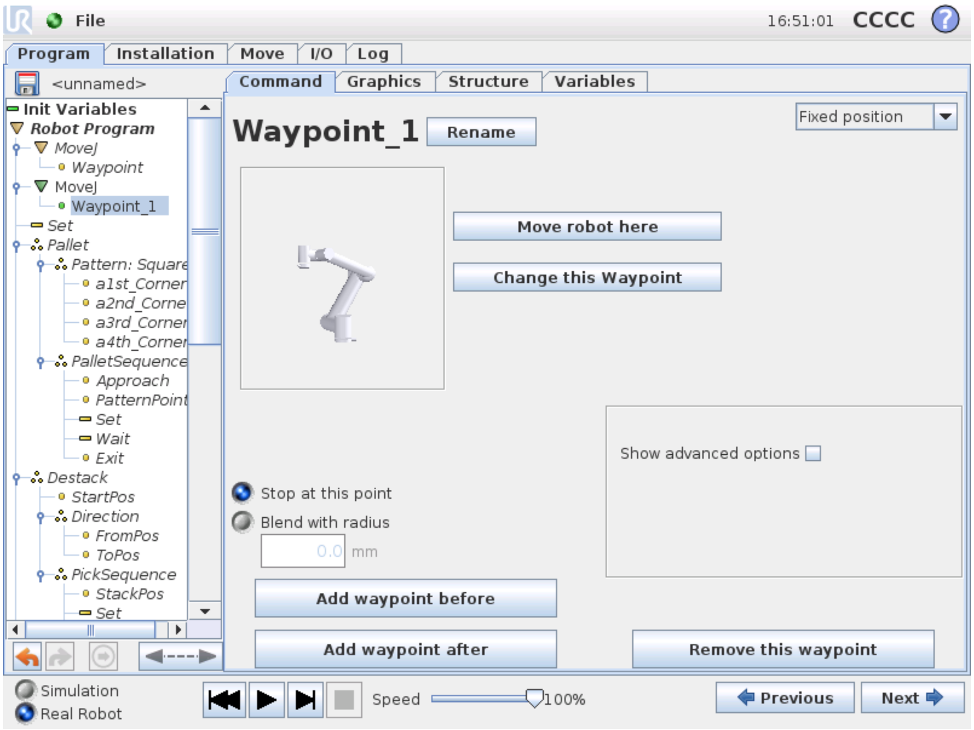
Figure 13.1: Speed profile for a motion. The curve is divided into three segments: *acceleration*, *cruise* and *deceleration*. The level of the *cruise* phase is given by the speed setting of the motion, while the steepness of the *acceleration* and *deceleration* phases is given by the acceleration parameter.

Relative waypoints: The selected feature has no effect on relative waypoints. The relative movement is always performed w.r.t. to orientation of the *Base*.

Variable waypoints: When the robot arm moves to a variable waypoint, the tool target position 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.

Variable feature: If any of the features in the currently loaded installation are selected as variable, these corresponding variables are also selectable in the feature selection menu. If a feature variable (named with the name of the feature suffixed by "_var") is selected, robot arm movements (except to *Relative* waypoints) are relative to the actual value of the variable when the program is running. The initial value of a feature variable is the value of the actual feature as configured in the installation. If this value is modified, then the movements of the robot change.

13.6 Command: Fixed Waypoint


File
16:51:01
CCCC
Program
Installation
Move
I/O
Log
<unnamed>
Command
Graphics
Structure
Variables
Init Variables
Robot Program
MoveJ
Waypoint
MoveJ
Waypoint_1
Set
Pallet
Pattern: Square
a1st_Corner
a2nd_Corne
a3rd_Corner
a4th_Corner
PalletSequence
Approach
PatternPoint
Set
Wait
Exit
Destack
StartPos
Direction
FromPos
ToPos
PickSequence
StackPos
Set
Waypoint_1
Rename
Fixed position
Move robot here
Change this Waypoint
Show advanced options ☐
● Stop at this point
○ Blend with radius
0.0 mm
Add waypoint before
Add waypoint after
Remove this waypoint
○ Simulation
● Real Robot
Speed
100%
Previous
Next

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 given by physically moving the robot arm to the position.

Setting the waypoint

Press this button to enter the Move screen where you can specify the robot arm's position for this waypoint. If the waypoint is placed under a Move command in linear space (`moveL` or `moveP`), there needs to be a valid feature selected at that Move command, in order for this button to be pressable.

Waypoint names

Waypoint names can be changed. Two waypoints with the same name is always the same waypoint. Waypoints are numbered as they are specified.

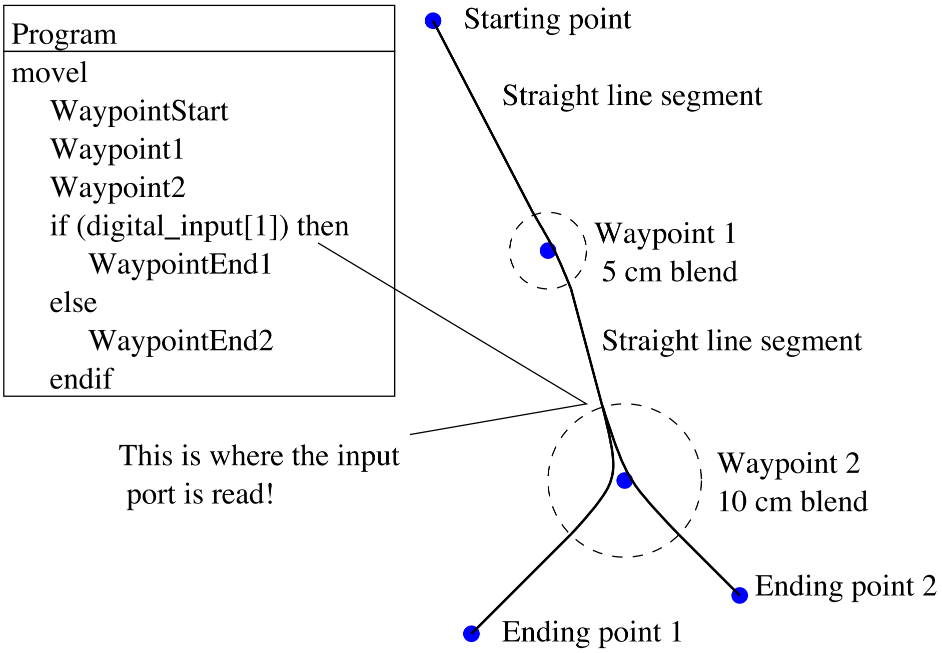
Blend radius

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 a blend radius for a previous or following waypont. A stop point is a waypoint with a blend radius of $0.0mm$.

Note on I/O Timing

If a waypoint is a stop point with an I/O command as the next command, the I/O command is executed when the robot arm stops at the waypoint. However, if the waypoint has a blend radius, the following I/O command is executed when the robot arm enters the blend.

Example

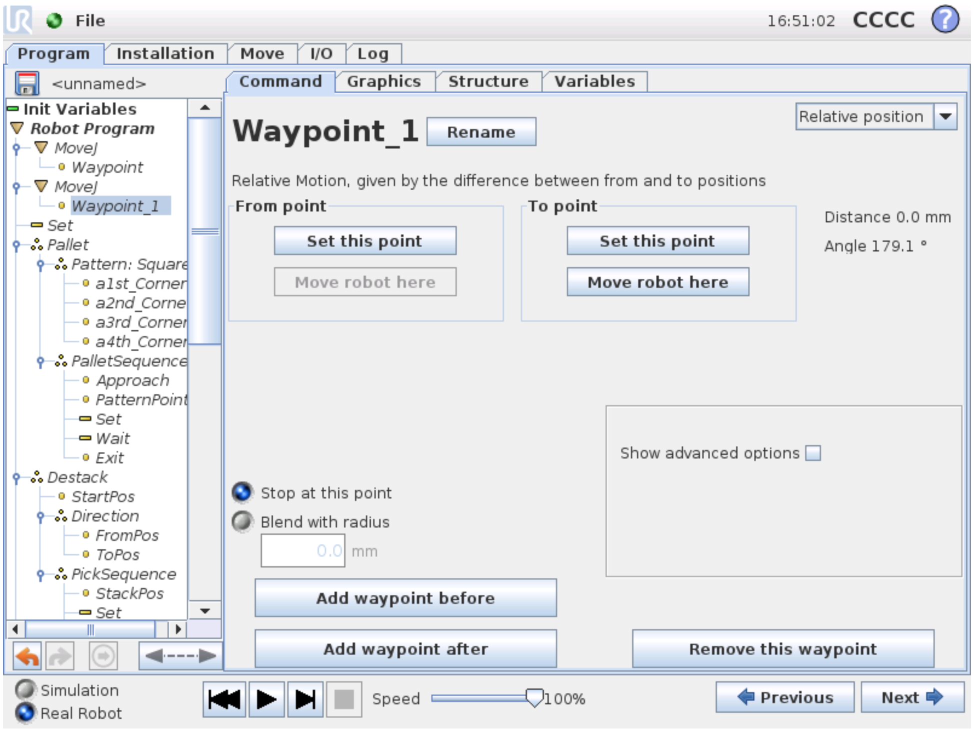


```
Program
movel
    WaypointStart
    Waypoint1
    Waypoint2
    if (digital_input[1]) then
        WaypointEnd1
    else
        WaypointEnd2
    endif
```

Starting point
Straight line segment
Waypoint 1
5 cm blend
Straight line segment
This is where the input port is read!
Waypoint 2
10 cm blend
Ending point 2
Ending point 1

A small example in which a robot program moves the tool from a starting position to one of two ending positions, depending on the state of `digital_input[1]`. Notice that the tool trajectory (thick black line) moves in straight lines outside the blend areas (dashed circles), while the tool trajectory deviates from the straight line path inside the blend areas. Also notice that the state of the `digital_input[1]` sensor is read just as the robot arm is about to enter the blend area around `Waypoint 2`, even though the `if...then` command is after `Waypoint 2` in the program sequence. This is somewhat counter-intuitive, but is necessary to select the right blend path.

13.7 Command: Relative Waypoint


File
16:51:02 CCCC
Program
Installation
Move
I/O
Log
<unnamed>
Init Variables
Robot Program
MoveJ
Waypoint
MoveJ
Waypoint_1
Set
Pallet
Pattern: Square
a1st_Corner
a2nd_Corne
a3rd_Corner
a4th_Corner
PalletSequence
Approach
PatternPoint
Set
Wait
Exit
Destack
StartPos
Direction
FromPos
ToPos
PickSequence
StackPos
Set
Command
Graphics
Structure
Variables
Waypoint_1
Rename
From point
Set this point
Move robot here
To point
Set this point
Move robot here
Show advanced options ☐
Blend with radius
0.0 mm
Add waypoint before
Add waypoint after
Remove this waypoint
Simulation
Real Robot
Speed
100%
Previous
Next

Relative position

Relative Motion, given by the difference between from and to positions

Distance 0.0 mm

Angle 179.1 °

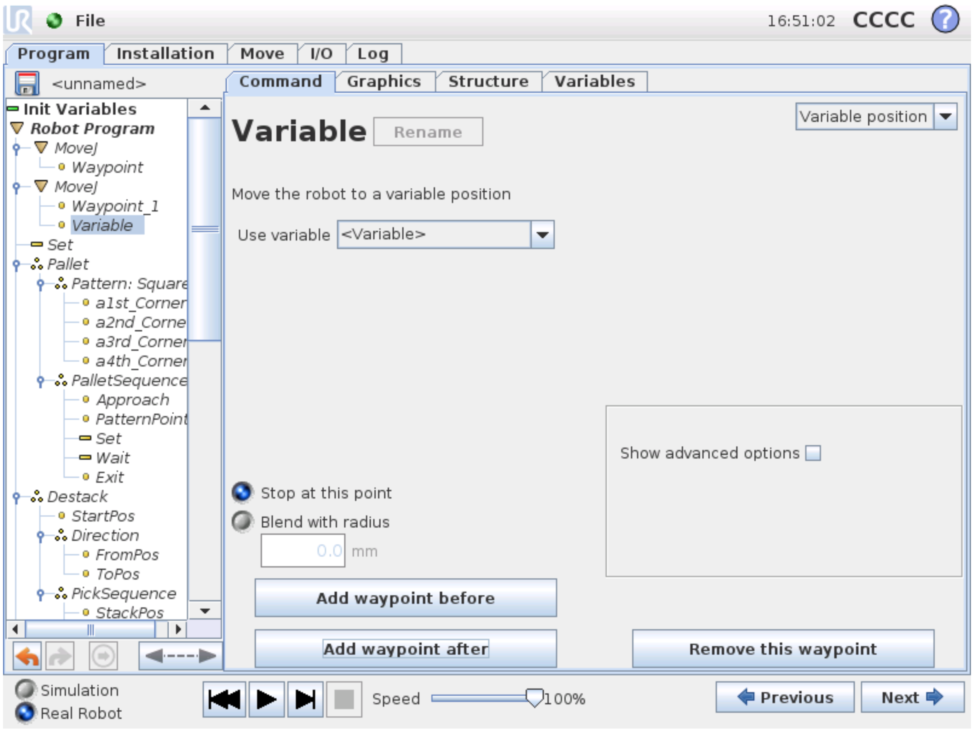
● Stop at this point

○ Blend with radius

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 that 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.

13.8 Command: Variable Waypoint


File
16:51:02 CCCC
Program
Installation
Move
I/O
Log
<unnamed>
Command
Graphics
Structure
Variables
Init Variables
Robot Program
MoveJ
Waypoint
MoveJ
Waypoint_1
Variable
Set
Pallet
Pattern: Square
a1st_Corner
a2nd_Corne
a3rd_Corner
a4th_Corner
PalletSequence
Approach
PatternPoint
Set
Wait
Exit
Destack
StartPos
Direction
FromPos
ToPos
PickSequence
StackPos
Variable
Rename
Variable position
Show advanced options ☐
Add waypoint before
Add waypoint after
Remove this waypoint
○ Simulation
● Real Robot
Speed
100%
Previous
Next

Move the robot to a variable position

Use variable <Variable>

● Stop at this point

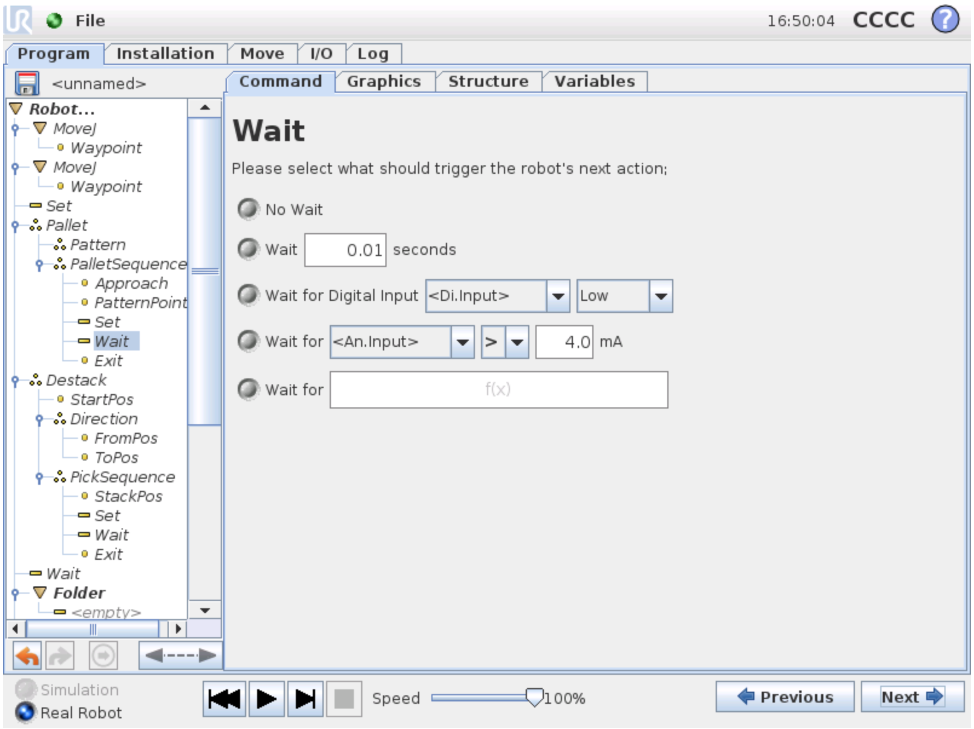
○ Blend with radius 0.0 mm

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[0.5,0.0,0.0,3.14,0.0,0.0]`. 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. The robot arm always moves linearly to a variable waypoint.

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]
MoveL
  Waypoint_1 (varibale position):
    Use variable=var_1, Feature=Tool
```

13.9 Command: Wait



File
16:50:04 CCCC
Program
Installation
Move
I/O
Log
<unnamed>
Command
Graphics
Structure
Variables
Robot...
MoveJ
Waypoint
MoveJ
Waypoint
Set
Pallet
Pattern
PalletSequence
Approach
PatternPoint
Set
Wait
Exit
Destack
StartPos
Direction
FromPos
ToPos
PickSequence
StackPos
Set
Wait
Exit
Wait
Folder
<empty>
Simulation
Real Robot
Speed
100%
Previous
Next

Wait

Please select what should trigger the robot's next action;

○ No Wait

○ Wait 0.01 seconds

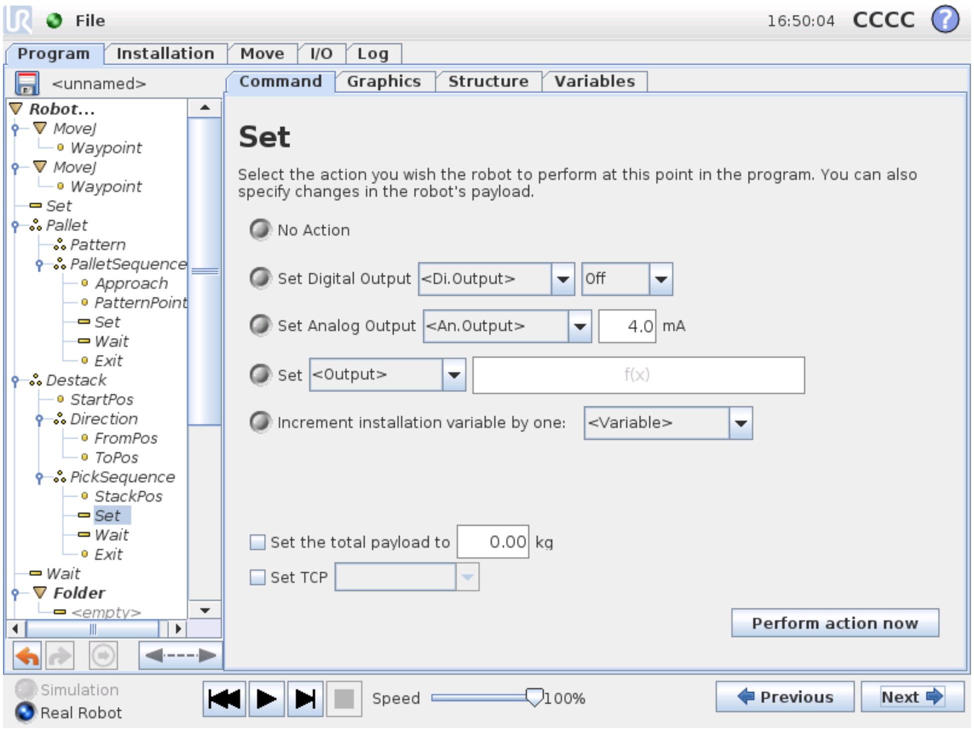
○ Wait for Digital Input <Di.Input> Low

○ Wait for <An.Input> > 4.0 mA

○ Wait for f(x)

Waits for a given amount of time or for an I/O signal.

13.10 Command: Set



File
16:50:04 CCCC
Program
Installation
Move
I/O
Log
<unnamed>
Command
Graphics
Structure
Variables
Robot...
MoveJ
Waypoint
MoveJ
Waypoint
Set
Pallet
Pattern
PalletSequence
Approach
PatternPoint
Set
Wait
Exit
Destack
StartPos
Direction
FromPos
ToPos
PickSequence
StackPos
Set
Wait
Exit
Wait
Folder
<empty>
Perform action now
Simulation
Real Robot
Speed
100%
Previous
Next

Set

Select the action you wish the robot to perform at this point in the program. You can also specify changes in the robot's payload.

○ No Action

○ Set Digital Output <Di.Output> Off

○ Set Analog Output <An.Output> 4.0 mA

○ Set <Output> f(x)

○ Increment installation variable by one: <Variable>

☐ Set the total payload to 0.00 kg

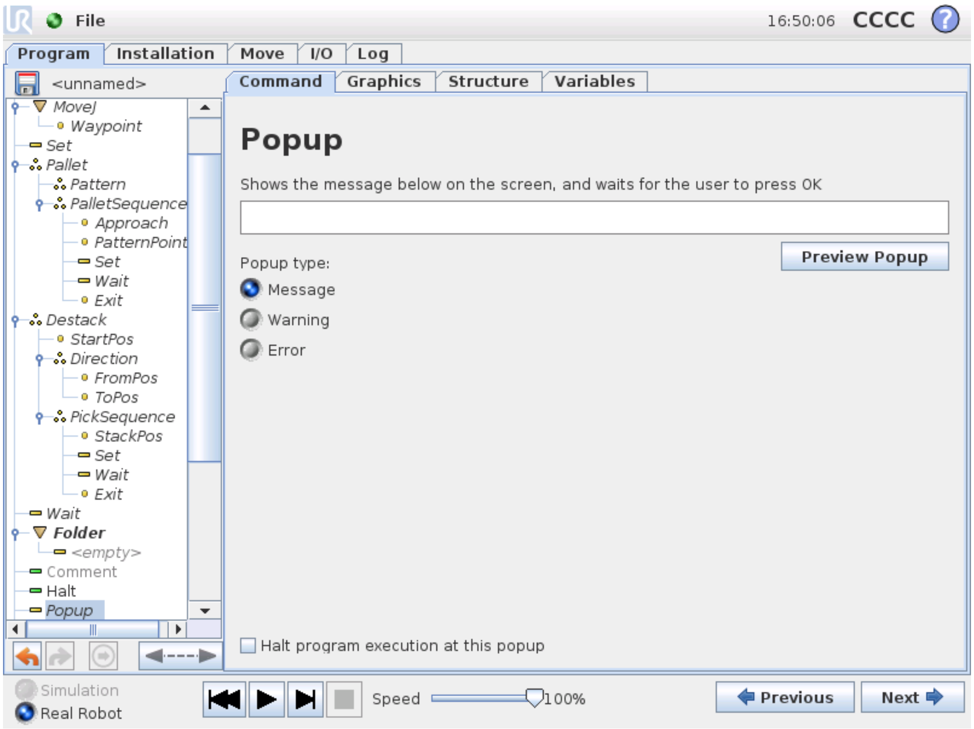
☐ Set TCP

Sets either digital or analog outputs to a given value.

Can also be used to set the payload of the robot arm, for example the weight that is picked up as a consequence of this action. Adjusting the weight can be neccesary to prevent the robot from triggering a protective stop unexpectedly, when the weight at the tool is different from the expected one.

The active TCP can also be modified using a Set command. Simply tick the check box and select one of the TCP offsets from the menu. If the active TCP for a particular motion is known at the time of writing of the program, consider using the TCP selection on the Move card instead (see 13.5). For further information about configuring named TCPs, see 12.6.

13.11 Command: Popup


File
16:50:06
CCCC
Program
Installation
Move
I/O
Log
<unnamed>
Command
Graphics
Structure
Variables
MoveJ
Waypoint
Set
Pallet
Pattern
PalletSequence
Approach
PatternPoint
Set
Wait
Exit
Destack
StartPos
Direction
FromPos
ToPos
PickSequence
StackPos
Set
Wait
Exit
Wait
Folder
<empty>
Comment
Halt
Popup
Preview Popup
Simulation
Real Robot
Speed
100%
Previous
Next

Popup

Shows the message below on the screen, and waits for the user to press OK

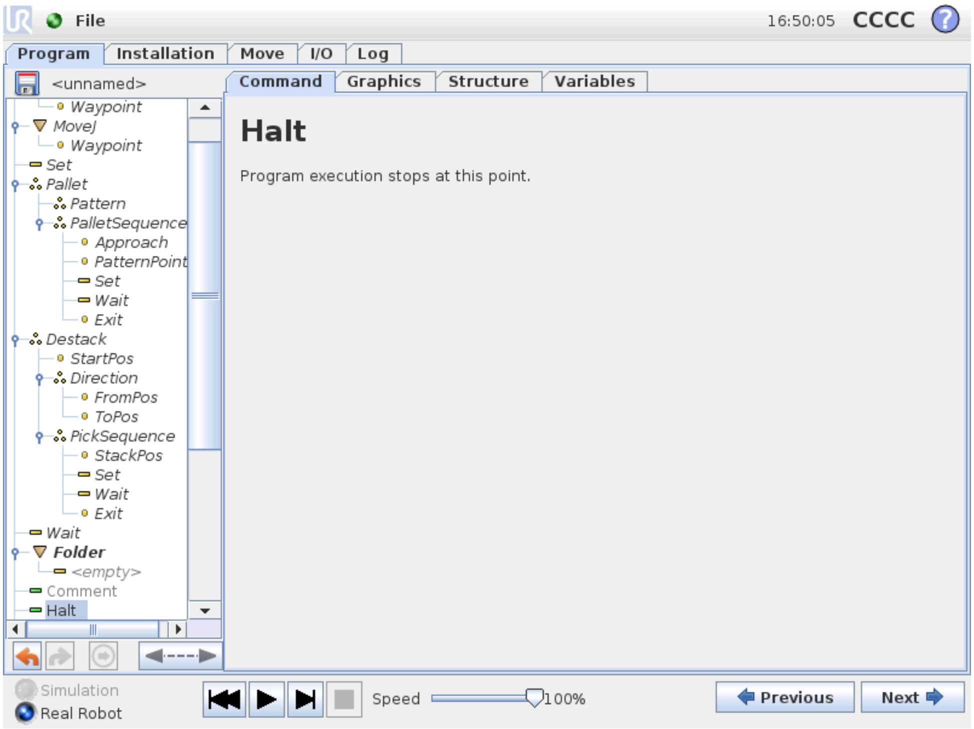
Popup type:

● Message
○ Warning
○ Error

☐ Halt program execution at this popup

The popup is a message that appears on the screen when the program reaches this command. The style of the message can be selected, and the text itself can be given using the on-screen keyboard. The robot waits for the user/operator to press the “OK” button under the popup before continuing the program. If the “Halt program execution” item is selected, the robot program halts at this popup.

13.12 Command: Halt



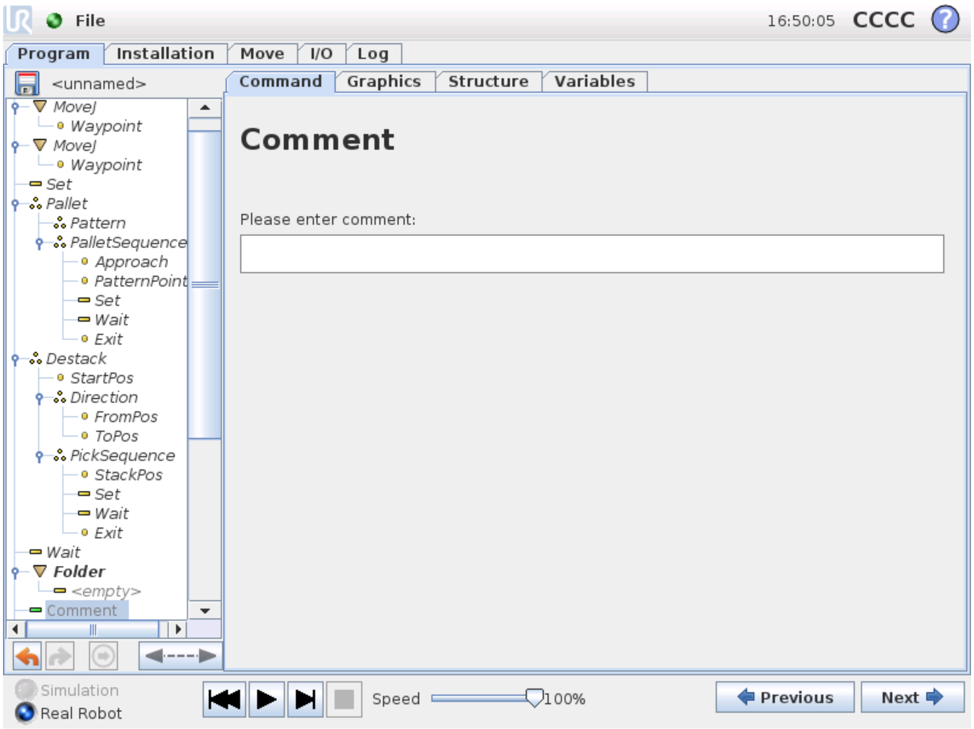
Halt

Program execution stops at this point.

File
16:50:05
CCCC
Program
Installation
Move
I/O
Log
<unnamed>
Command
Graphics
Structure
Variables
Waypoint
MoveJ
Waypoint
Set
Pallet
Pattern
PalletSequence
Approach
PatternPoint
Set
Wait
Exit
Destack
StartPos
Direction
FromPos
ToPos
PickSequence
StackPos
Set
Wait
Exit
Wait
Folder
<empty>
Comment
Halt
Simulation
Real Robot
Speed
100%
Previous
Next

The program execution stops at this point.

13.13 Command: Comment



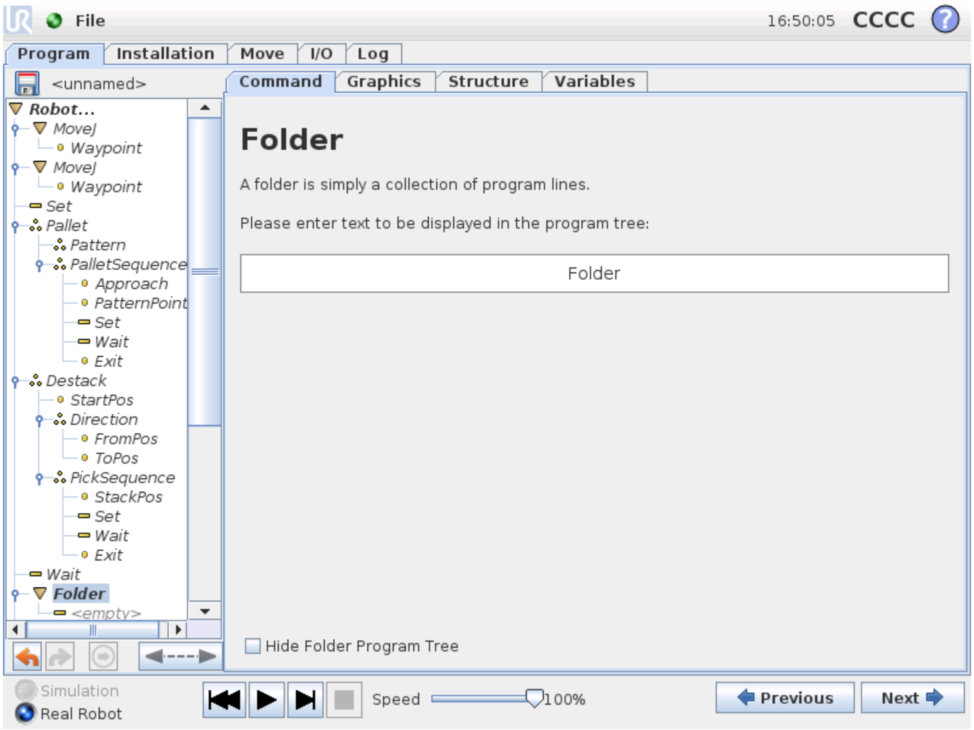
Comment

Please enter comment:

File
16:50:05
CCCC
Program
Installation
Move
I/O
Log
<unnamed>
Command
Graphics
Structure
Variables
MoveJ
Waypoint
MoveJ
Waypoint
Set
Pallet
Pattern
PalletSequence
Approach
PatternPoint
Set
Wait
Exit
Destack
StartPos
Direction
FromPos
ToPos
PickSequence
StackPos
Set
Wait
Exit
Wait
Folder
<empty>
Comment
Simulation
Real Robot
Speed
100%
Previous
Next

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.

13.14 Command: Folder


File
16:50:05
CCCC
Program
Installation
Move
I/O
Log
<unnamed>
Command
Graphics
Structure
Variables
Robot...
MoveJ
Waypoint
MoveJ
Waypoint
Set
Pallet
Pattern
PalletSequence
Approach
PatternPoint
Set
Wait
Exit
Destack
StartPos
Direction
FromPos
ToPos
PickSequence
StackPos
Set
Wait
Exit
Wait
Folder
<empty>
Simulation
Real Robot
Speed
100%
Previous
Next

Folder

A folder is simply a collection of program lines.

Please enter text to be displayed in the program tree:

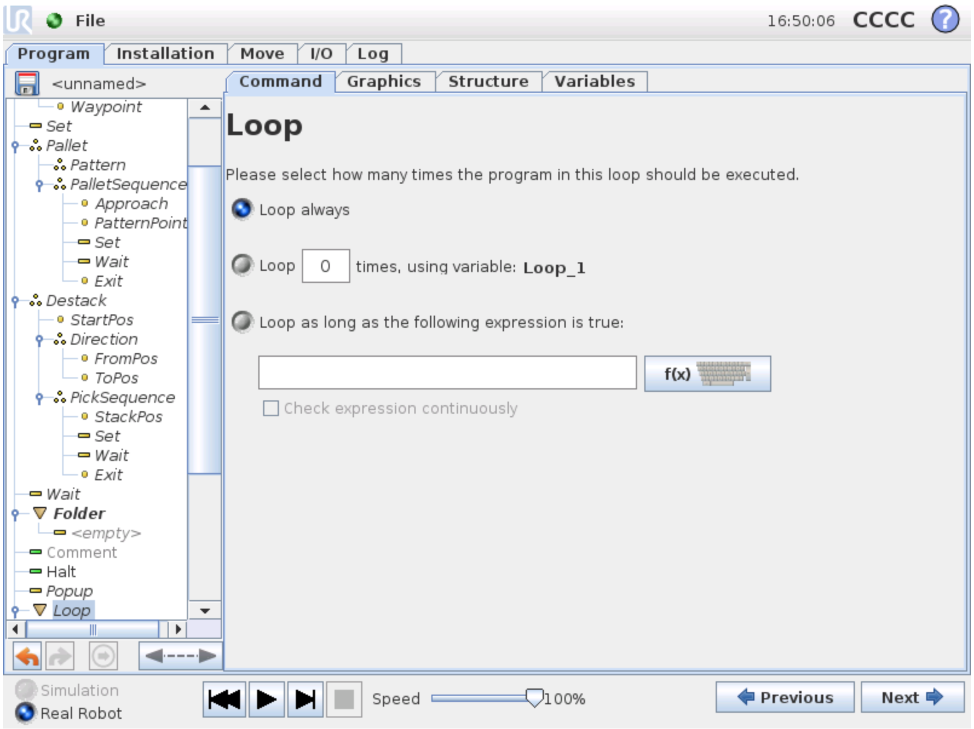
Folder

☐ Hide Folder Program Tree

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.

A folder does not in itself do anything.

13.15 Command: Loop


File
16:50:06
CCCC
Program
Installation
Move
I/O
Log
<unnamed>
Command
Graphics
Structure
Variables
Waypoint
Set
Pallet
Pattern
PalletSequence
Approach
PatternPoint
Set
Wait
Exit
Destack
StartPos
Direction
FromPos
ToPos
PickSequence
StackPos
Set
Wait
Exit
Wait
Folder
<empty>
Comment
Halt
Popup
Loop
f(x)
Simulation
Real Robot
Speed
100%
Previous
Next

Loop

Please select how many times the program in this loop should be executed.

● Loop always

○ Loop 0 times, using variable: Loop_1

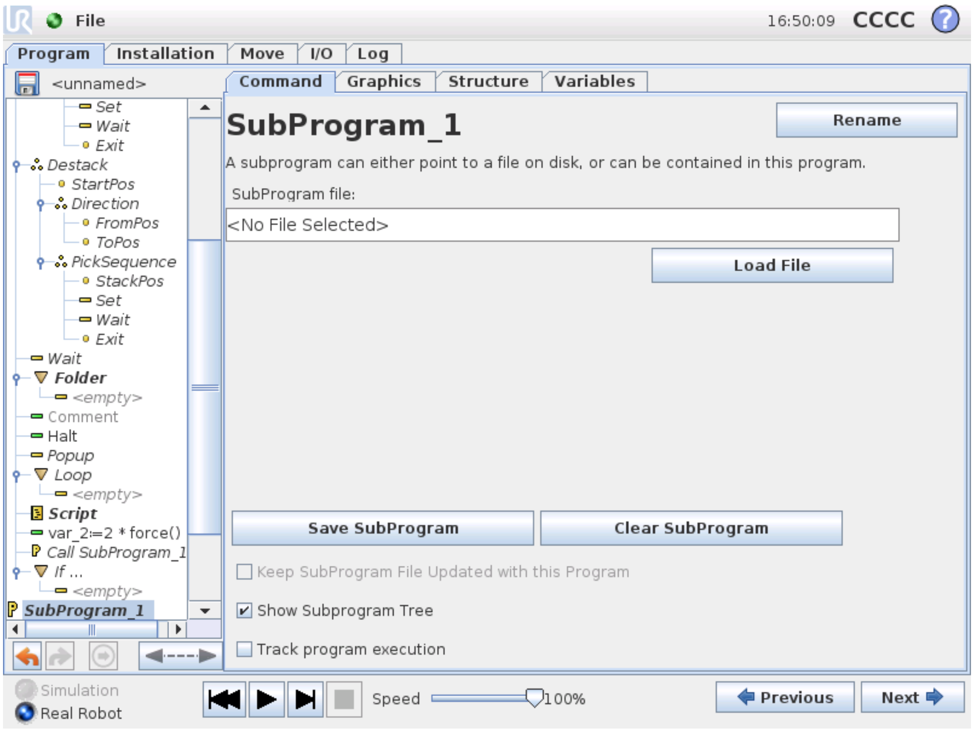
○ Loop as long as the following expression is true:

☐ Check expression continuously

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

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.

13.16 Command: SubProgram


File
16:50:09
CCCC
Program
Installation
Move
I/O
Log
<unnamed>
Set
Wait
Exit
Destack
StartPos
Direction
FromPos
ToPos
PickSequence
StackPos
Set
Wait
Exit
Wait
Folder
<empty>
Comment
Halt
Popup
Loop
<empty>
Script
var_2:=2 * force()
Call SubProgram_1
If ...
<empty>
SubProgram_1
Command
Graphics
Structure
Variables
SubProgram_1
Rename
Load File
Save SubProgram
Clear SubProgram
Simulation
Real Robot
Speed
100%
Previous
Next

A subprogram can either point to a file on disk, or can be contained in this program.

SubProgram file: <No File Selected>

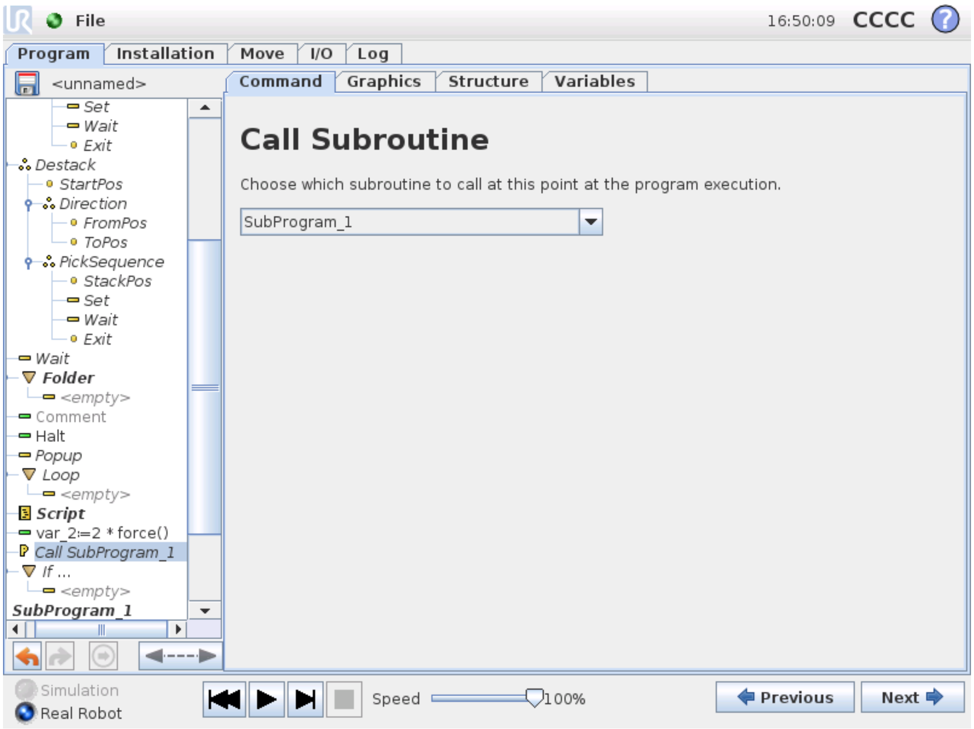
☐ Keep SubProgram File Updated with this Program

☑ Show Subprogram Tree

☐ Track program execution

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

Command: Call SubProgram


File
16:50:09 CCCC
Program
Installation
Move
I/O
Log
<unnamed>
Command
Graphics
Structure
Variables
Set
Wait
Exit
Destack
StartPos
Direction
FromPos
ToPos
PickSequence
StackPos
Set
Wait
Exit
Wait
Folder
<empty>
Comment
Halt
Popup
Loop
<empty>
Script
var_2:=2 * force()
Call SubProgram_1
If ...
<empty>
SubProgram_1
Simulation
Real Robot
Speed
100%
Previous
Next

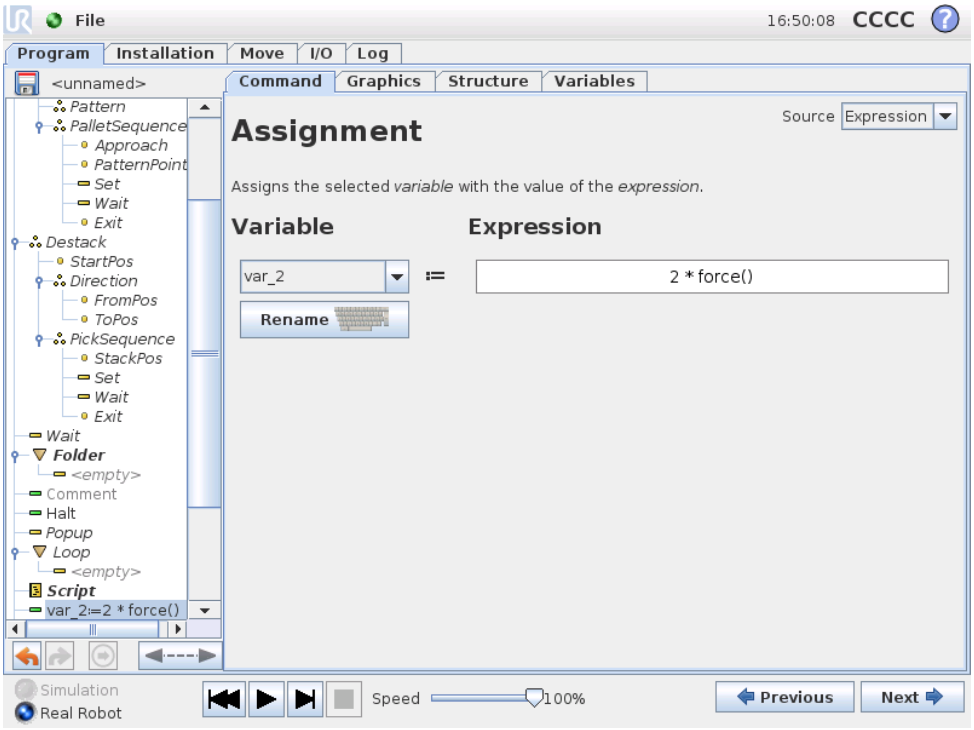
Call Subroutine

Choose which subroutine to call at this point at the program execution.

SubProgram_1

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

13.17 Command: Assignment


File
16:50:08 CCCC
Program
Installation
Move
I/O
Log
<unnamed>
Command
Graphics
Structure
Variables
Pattern
PalletSequence
Approach
PatternPoint
Set
Wait
Exit
Destack
StartPos
Direction
FromPos
ToPos
PickSequence
StackPos
Set
Wait
Exit
Wait
Folder
<empty>
Comment
Halt
Popup
Loop
<empty>
Script
var_2:=2 * force()
Rename
Simulation
Real Robot
Speed
100%
Previous
Next

Assignment

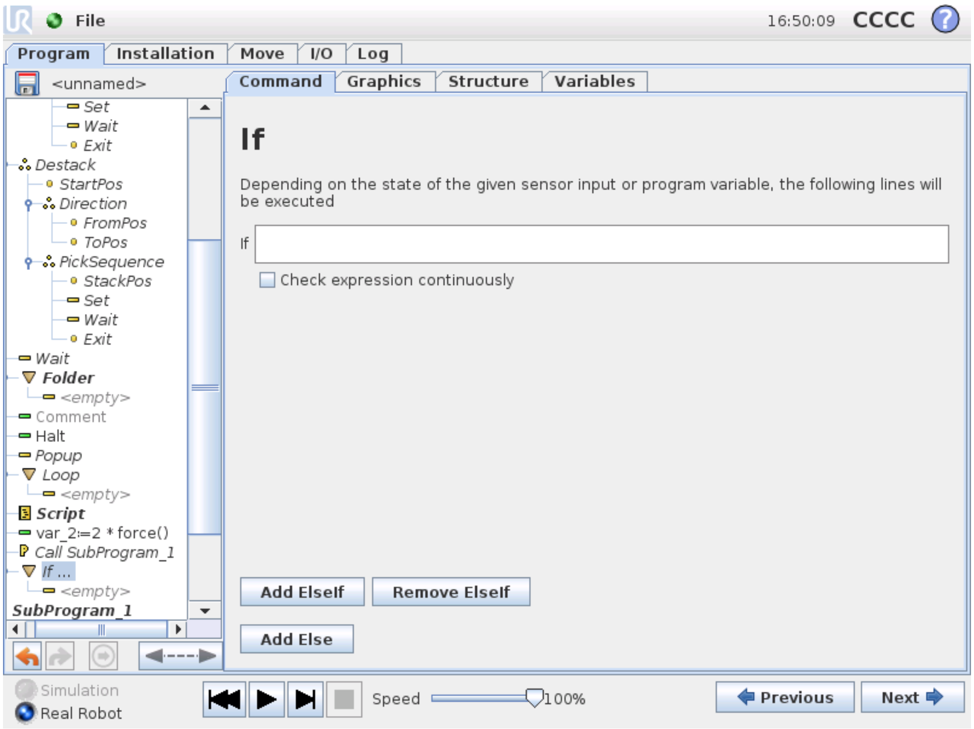
Source: Expression

Assigns the selected *variable* with the value of the *expression*.

Variable		Expression
var_2	:=	2 * force()

Assigns values to variables. An assignment puts the computed value of the right hand side into the variable on the left hand side. This can be useful in complex programs.

13.18 Command: If


File
16:50:09
CCCC
Program
Installation
Move
I/O
Log
<unnamed>
Command
Graphics
Structure
Variables
Set
Wait
Exit
Destack
StartPos
Direction
FromPos
ToPos
PickSequence
StackPos
Set
Wait
Exit
Wait
Folder
<empty>
Comment
Halt
Popup
Loop
<empty>
Script
var_2:=2 * force()
Call SubProgram_1
If ...
<empty>
SubProgram_1
Add ElseIf
Remove ElseIf
Add Else
Simulation
Real Robot
Speed
100%
Previous
Next

If

Depending on the state of the given sensor input or program variable, the following lines will be executed

If

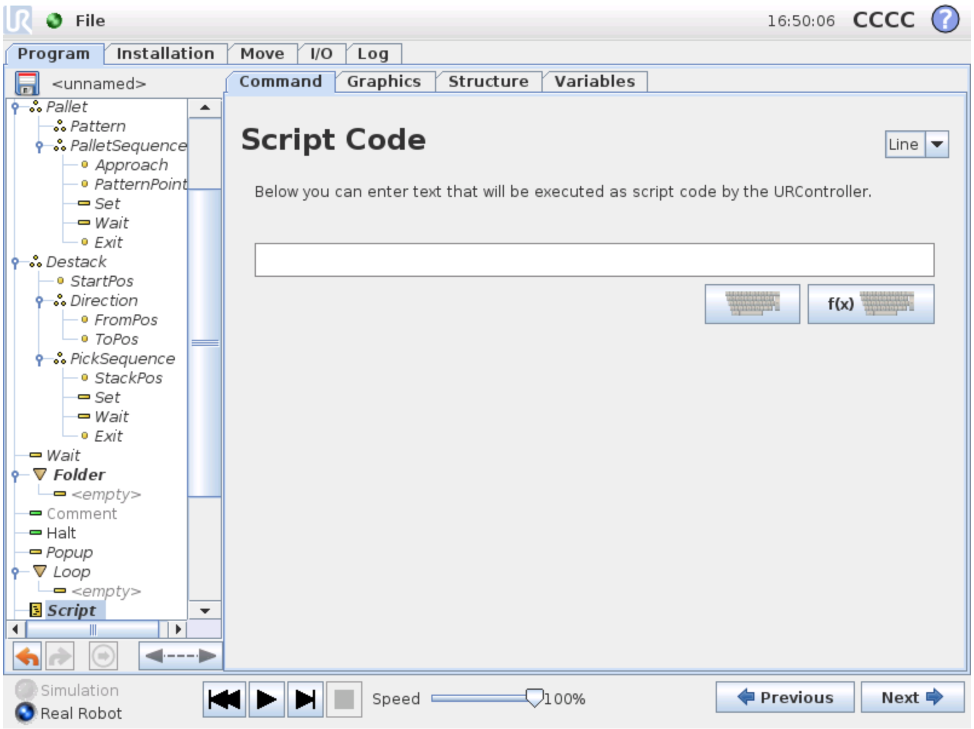
☐ Check expression continuously

An "if...else" construction can make the robot change its behavior based on sensor inputs or variable values. Use the expression editor to describe the condition under which the robot should proceed to the sub-commands of this `If`. If the condition is evaluated to `True`, the lines inside this `If` are executed.

Each `If` can have several `ElseIf` and one `Else` command. These can be added using the buttons on the screen. An `ElseIf` command can be removed from the screen for that command.

The open `Check Expression Continuously` allow the conditions of the `If` and `ElseIf` statements to be evaluated while the contained lines are executed. If a expression evaluates to *False* while inside the body of the `If`-part, the following `ElseIf` or `Else` statement will be reached.

13.19 Command: Script



Script Code

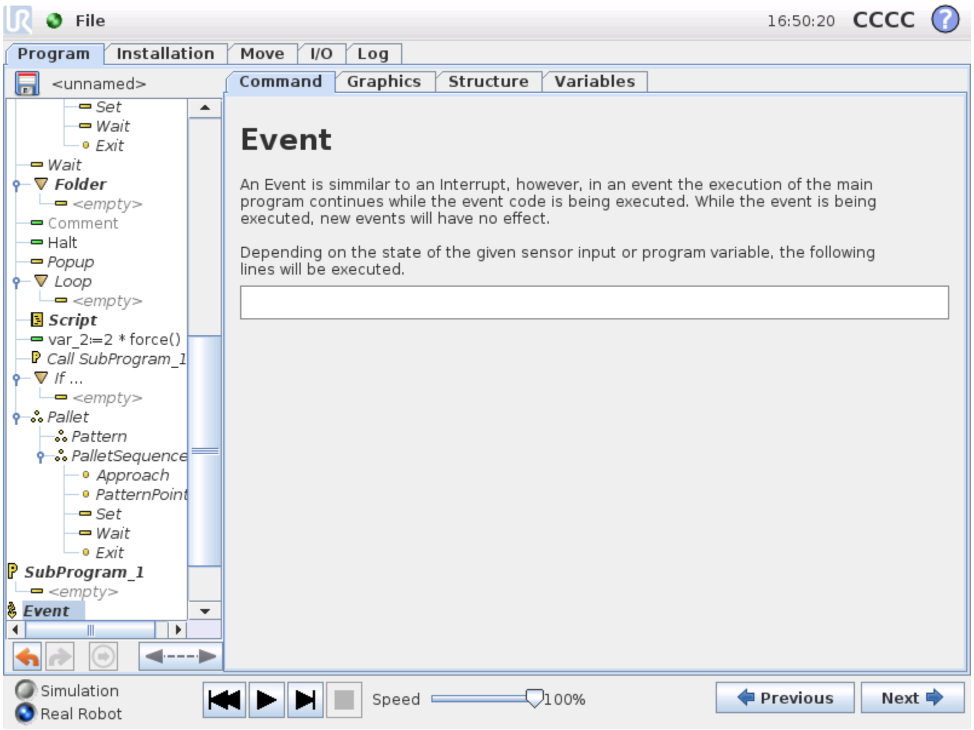
Below you can enter text that will be executed as script code by the URController.

File
16:50:06
CCCC
Program
Installation
Move
I/O
Log
<unnamed>
Command
Graphics
Structure
Variables
Line
Pallet
Pattern
PalletSequence
Approach
PatternPoint
Set
Wait
Exit
Destack
StartPos
Direction
FromPos
ToPos
PickSequence
StackPos
Set
Wait
Exit
Wait
Folder
<empty>
Comment
Halt
Popup
Loop
<empty>
Script
f(x)
Simulation
Real Robot
Speed
100%
Previous
Next

This command gives access to the underlying real time script language that is executed by the robot controller. It is intended for advanced users only and instructions on how to use it can be found in the Script Manual on the support website (`http://support.universal-robots.com/`). Note that only UR distributors and OEM customers have access to the website.

If the "File" option in the top left corner is choosen, it is possible to create and edit script programs files. This way, long and complex script programs can be used together with the operator-friendly programming of PolyScope.

13.20 Command: Event


File
16:50:20
CCCC
Program
Installation
Move
I/O
Log
<unnamed>
Command
Graphics
Structure
Variables
Set
Wait
Exit
Wait
Folder
<empty>
Comment
Halt
Popup
Loop
<empty>
Script
var_2:=2 * force()
Call SubProgram_1
If ...
<empty>
Pallet
Pattern
PalletSequence
Approach
PatternPoint
Set
Wait
Exit
SubProgram_1
<empty>
Event
Simulation
Real Robot
Speed
100%
Previous
Next

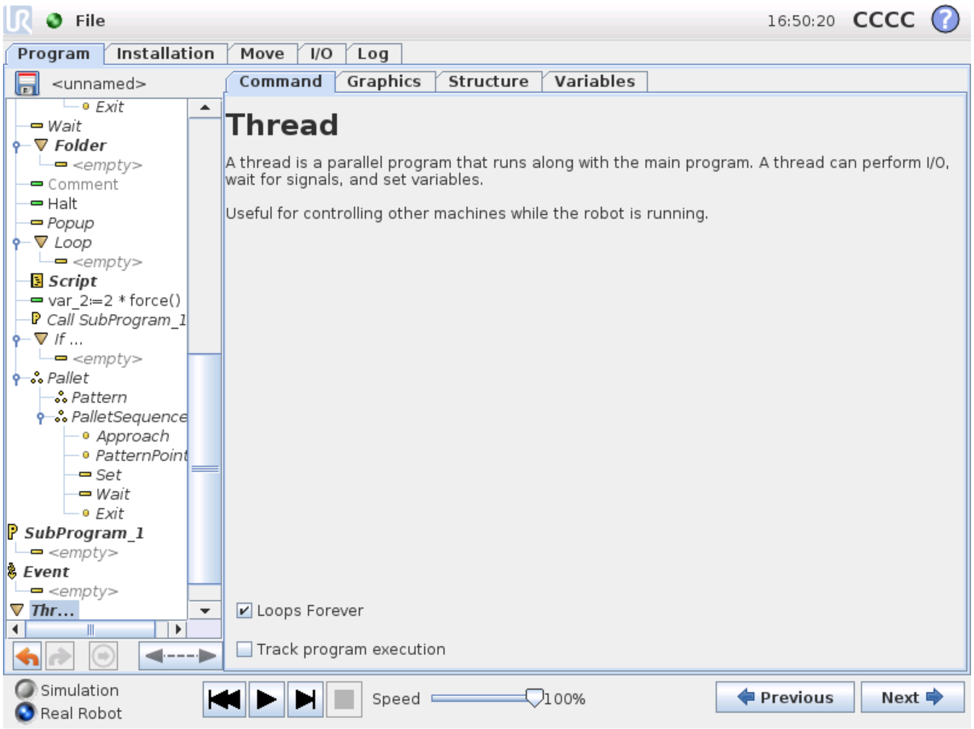
Event

An Event is simmilar to an Interrupt, however, in an event the execution of the main program continues while the event code is being executed. While the event is being executed, new events will have no effect.

Depending on the state of the given sensor input or program variable, the following lines will be executed.

An event can be used to monitor an input signal, and perform some action or set a variable when that input signal goes high. For example, in the event that an output signal goes high, the event program can wait for 200ms and then set it back to low again. This can make the main program code a lot simpler in the case on an external machine triggering on a rising flank rather than a high input level. Events are checked once every control cycle (8ms).

13.21 Command: Thread


File
16:50:20 CCCC
Program
Installation
Move
I/O
Log
<unnamed>
Command
Graphics
Structure
Variables
Exit
Wait
Folder
<empty>
Comment
Halt
Popup
Loop
<empty>
Script
var_2:=2 * force()
Call SubProgram_1
If ...
<empty>
Pallet
Pattern
PalletSequence
Approach
PatternPoint
Set
Wait
Exit
SubProgram_1
<empty>
Event
<empty>
Thr...
Simulation
Real Robot
Speed
100%
Previous
Next

Thread

A thread is a parallel program that runs along with the main program. A thread can perform I/O, wait for signals, and set variables.

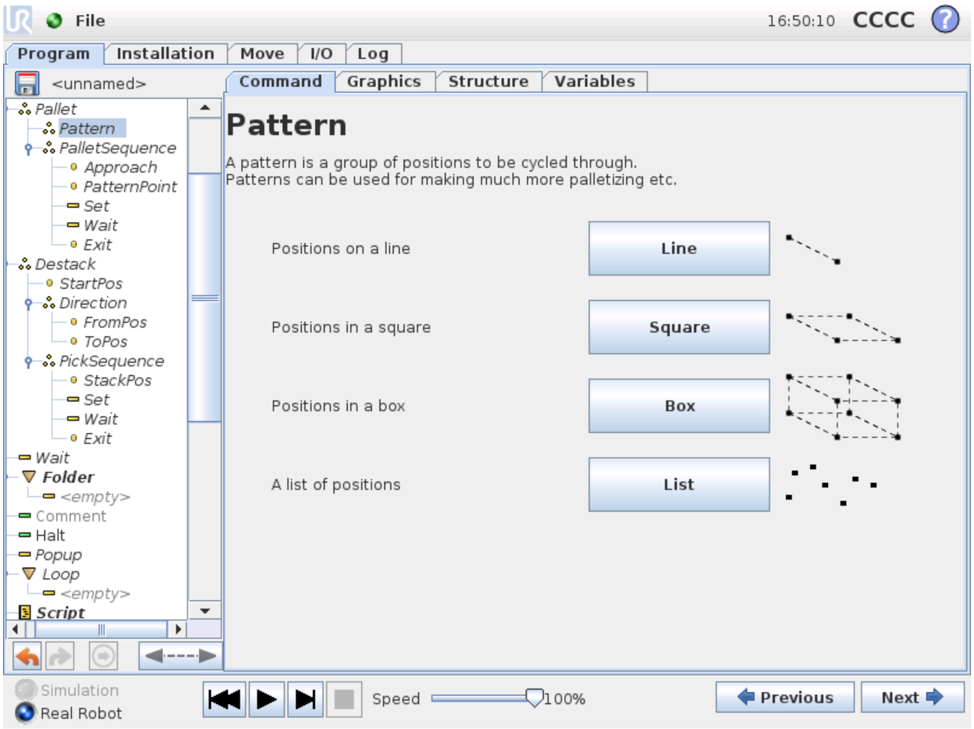
Useful for controlling other machines while the robot is running.

☑ Loops Forever

☐ Track program execution

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.

13.22 Command: Pattern


File
16:50:10 CCCC
Program
Installation
Move
I/O
Log
<unnamed>
Command
Graphics
Structure
Variables
Pallet
Pattern
PalletSequence
Approach
PatternPoint
Set
Wait
Exit
Destack
StartPos
Direction
FromPos
ToPos
PickSequence
StackPos
Set
Wait
Exit
Wait
Folder
<empty>
Comment
Halt
Popup
Loop
<empty>
Script
Line
Square
Box
List
Simulation
Real Robot
Speed
100%
Previous
Next

Pattern

A pattern is a group of positions to be cycled through.
Patterns can be used for making much more palletizing etc.

Positions on a line

Positions in a square

Positions in a box

A list of positions

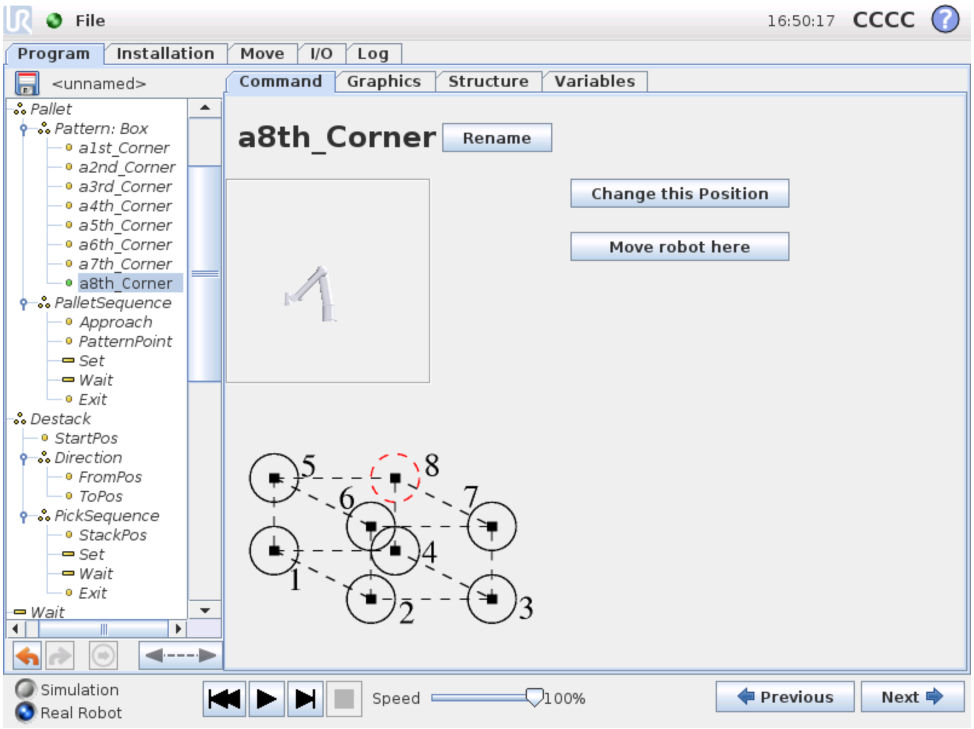
The Pattern command can be used to cycle through positions in the robot program. The pattern command corresponds to one position at each execution.

A pattern can be given as one of four types. The first three, “Line”, “Square” or “Box” can be used for positions in a regular pattern. The regular patterns are defined by a number of characteristic points, where the points define the edges of the pattern. For “Line” this is the two end points, for “Square” this is three of the four corner points, where as for “Box” this is four of the eight corner points. The programmer enters the number of positions along each of the edges of the pattern. The robot controller then calculates the individual pattern positions by proportionally adding the edge vectors together.

If the positions to be traversed do not fall in a regular pattern, the “List” option can be chosen, where a list of all the positions is provided by the programmer. This way any kind of arrangement of the positions can be realized.

Defining the Pattern

When the “Box” pattern is selected, the screen changes to what is shown below.


File
16:50:17
CCCC
Program
Installation
Move
I/O
Log
<unnamed>
Command
Graphics
Structure
Variables
Pallet
Pattern: Box
a1st_Corner
a2nd_Corner
a3rd_Corner
a4th_Corner
a5th_Corner
a6th_Corner
a7th_Corner
a8th_Corner
PalletSequence
Approach
PatternPoint
Set
Wait
Exit
Destack
StartPos
Direction
FromPos
ToPos
PickSequence
StackPos
Set
Wait
Exit
Wait
a8th_Corner
Rename
Change this Position
Move robot here
5
8
6
7
4
1
2
3
Simulation
Real Robot
Speed
100%
Previous
Next

A “Box” pattern uses three vectors to define the side of the box. These three vectors are given as four points, where the first vector goes from point one to point two, the second vector goes from point two to point three, and the third vector goes from point three to point four. Each vector is divided by the interval count numbers. A specific position in the pattern is calculated by simply adding the interval vectors proportionally.

The “Line” and “Square” patterns work similarly.

A counter variable is used while traversing the positions of the pattern. The name of the variable can be seen on the `Pattern` command screen. The variable cycles through the numbers from 0 to $X * Y * Z - 1$, the number of points in the pattern. This variable can be manipulated using assignments, and can be used in expressions.

13.23 Command: Force

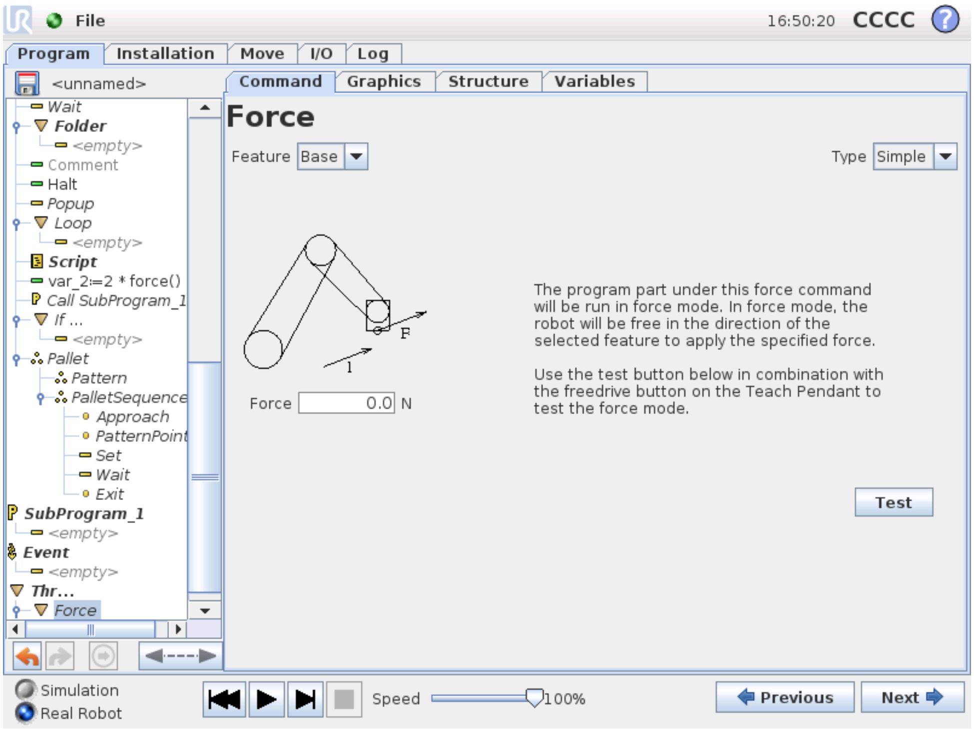
Force mode allows for compliance and forces in selectable axis in the robot's workspace. All robot arm movements under a Force command will be in Force mode. When the robot arm is moving in force mode, it is possible to select one or more axes in which the robot arm is compliant. Along/around compliant axes the robot arm will comply with the environment, which means it will automatically adjust its position in order to achieve the desired force. It is also possible to make the robot arm itself apply a force to its environment, e.g. a workpiece.

Force mode is suited for applications where the actual tcp position along a predefined axis is not important, but in stead a desired force along that axis is required. For example if the robot TCP should roll against a curved surface, or when pushing or pulling a workpiece. Force mode also supports applying certain torques around predefined axes. Note that if no obstacles are met in an axis where a non-zero force is set, the robot arm will try to accelerate along/about that axis.

Although an axis has been selected to be compliant, the robot program will still try to move the robot along/around that axis. However, the force control assures that the robot arm will still approach the specified force.

WARNING:

If the force function is used incorrectly, it can produce a force of more than 150N. The programmed force shall be taken into consideration during risk assessment.


File
16:50:20 CCCC
Program
Installation
Move
I/O
Log
<unnamed>
Command
Graphics
Structure
Variables
Wait
Folder
<empty>
Comment
Halt
Popup
Loop
<empty>
Script
var_2:=2 * force()
Call SubProgram_1
If ...
<empty>
Pallet
Pattern
PalletSequence
Approach
PatternPoint
Set
Wait
Exit
SubProgram_1
<empty>
Event
<empty>
Thr...
Force
F
1
Test
Simulation
Real Robot
Speed
100%
Previous
Next

Force

Feature Base

Type Simple

Force 0.0 N

The program part under this force command will be run in force mode. In force mode, the robot will be free in the direction of the selected feature to apply the specified force.

Use the test button below in combination with the freedrive button on the Teach Pendant to test the 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 12.12.

Force mode type

The are four different types of force mode each determining the way in which the selected feature will be 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 origo of the selected feature. The distance between the robot TCP and the origo of the selected feature is required to be at least 10 mm. Note that the task frame will change 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 usefull when deburring along a complex path, where a force is needed perpendicular to the TCP

motion. Note, 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 on 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 (13.28), when the robot is operating in force mode.

Force value selection

A force can be set for both compliant and non-compliant axes, but the effects are different.

- **Compliant**: The robot arm will adjust its position to achieve the selected force.
- **Non-compliant**: The robot arm will follow its trajectory set by the program while accounting for an external force of the value set here.

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

Limits selection

For all axes a limit can be set, but these have different meaning corresponding to the axes being complian or non-compliant.

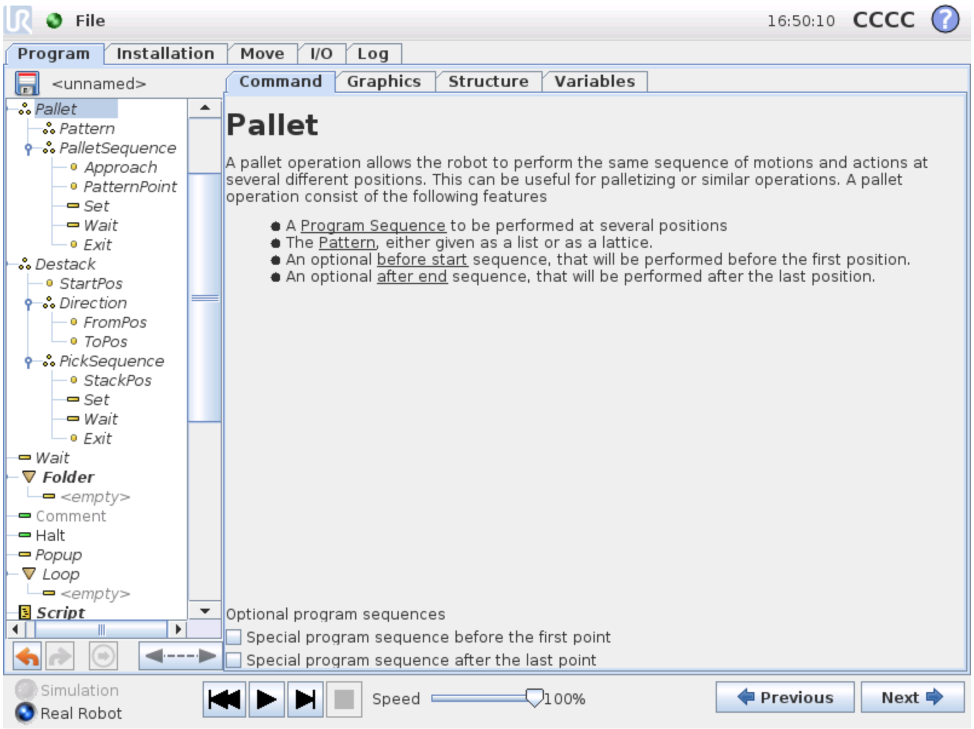
- **Compliant**: The limit is the maximum speed the TCP is allowed to attain along/about the axis. Units are [mm/s] and [deg/s].
- **Non-compliant**: The limit is the maximum deviation from the program trajectory which is allowed before the robot protective stops. Units are [mm] and [deg].

Test force settings

The on/off button, labeled “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. Note: The Freedrive button will only be effectual when a valid feature has been selected for the Force command.

13.24 Command: Pallet


File
16:50:10 CCCC
Program
Installation
Move
I/O
Log
<unnamed>
Command
Graphics
Structure
Variables
Pallet
Pattern
PalletSequence
Approach
PatternPoint
Set
Wait
Exit
Destack
StartPos
Direction
FromPos
ToPos
PickSequence
StackPos
Set
Wait
Exit
Wait
Folder
<empty>
Comment
Halt
Popup
Loop
<empty>
Script
Simulation
Real Robot
Speed
100%
Previous
Next

Pallet

A pallet operation allows the robot to perform the same sequence of motions and actions at several different positions. This can be useful for palletizing or similar operations. A pallet operation consist of the following features

- A Program Sequence to be performed at several positions
- The Pattern, either given as a list or as a lattice.
- An optional before start sequence, that will be performed before the first position.
- An optional after end sequence, that will be performed after the last position.

Optional program sequences

☐ Special program sequence before the first point

☐ Special program sequence after the last point

A pallet operation can perform a sequence of motions in a set of places given as a pattern, as described in 13.22. At each of the positions in the pattern, the sequence of motions will be run relative to the pattern position.

Programming a Pallet Operation

The steps to go through are as follows;

1. Define the pattern.
2. Make a "PalletSequence" for picking up/placing at each single point. The sequence describes what should be done at each pattern position.
3. Use the selector on the sequence command screen to define which of the waypoints in the sequence should correspond to the pattern positions.

Pallet Sequence/Anchorable Sequence

In an Pallet Sequence node, the motions of the robot arm are relative to the pallet position. The behavior of a sequence is such that the robot arm will be at the position specified by the pattern at the `Anchor Position/Pattern Point`. The remaining positions will all be moved to make this fit.

Do not use the `Move` command inside a sequence, as it will not be relative to the anchor position.

"BeforeStart"

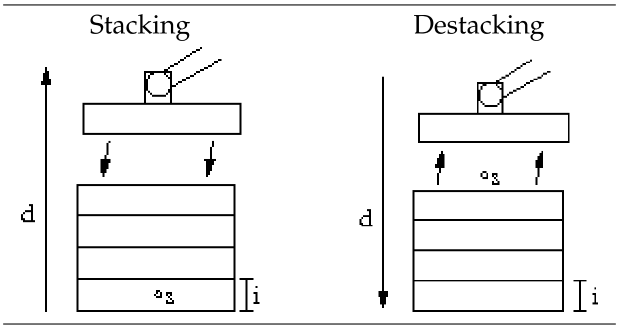
The optional `BeforeStart` sequence is run just before the operation starts. This can be used to wait for ready signals.

“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 pallet.

13.25 Command: Seek

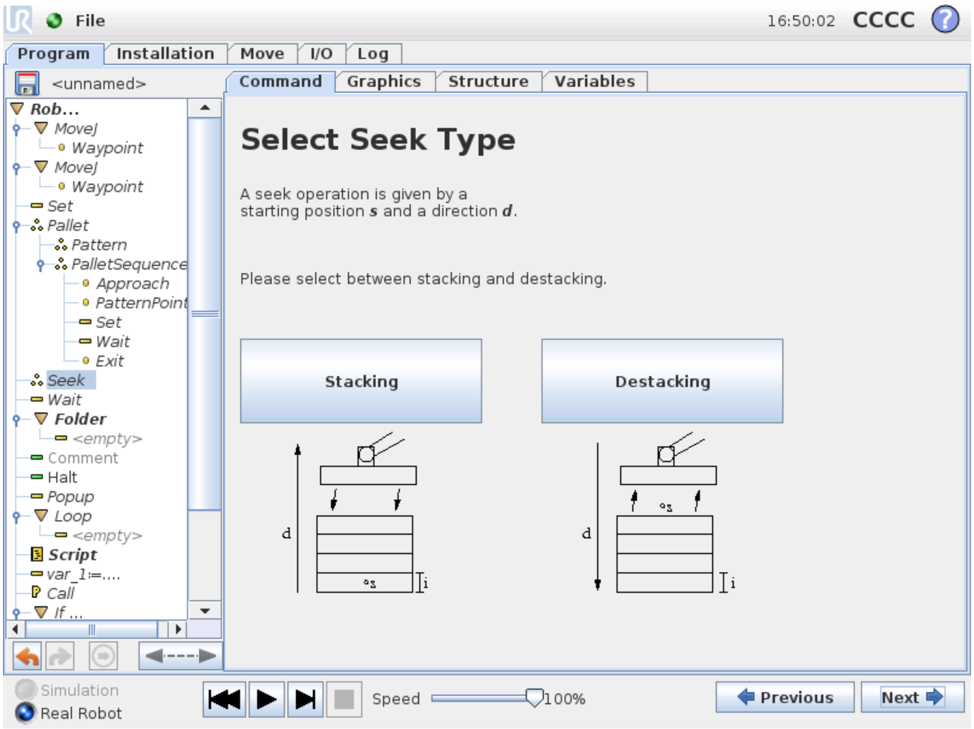
A seek function uses a sensor to determine when the correct position is reached to grab or drop an item. The sensor can be a push button switch, a pressure sensor or a capacitive sensor. This function is made for working on stacks of items with varying item thickness, or where the exact positions of the items are not known or too hard to program.



When programming a seek operation for working on a stack, one must define *s* the starting point, *d* the stack direction and *i* the thickness of the items in the stack.

On top of this, one must define the condition for when the next stack position is reached, and a special program sequence that will be performed at each of the stack positions. Also speed and accelerations need to be given for the movement involved in the stack operation.

Stacking



File
16:50:02 CCCC
Program
Installation
Move
I/O
Log
<unnamed>
Command
Graphics
Structure
Variables
Rob…
MoveJ
Waypoint
MoveJ
Waypoint
Set
Pallet
Pattern
PalletSequence
Approach
PatternPoint
Set
Wait
Exit
Seek
Wait
Folder
<empty>
Comment
Halt
Popup
Loop
<empty>
Script
var_1:=....
Call
If …
Stacking
Destacking
d
s
i
d
s
i
Simulation
Real Robot
Speed
100%
Previous
Next

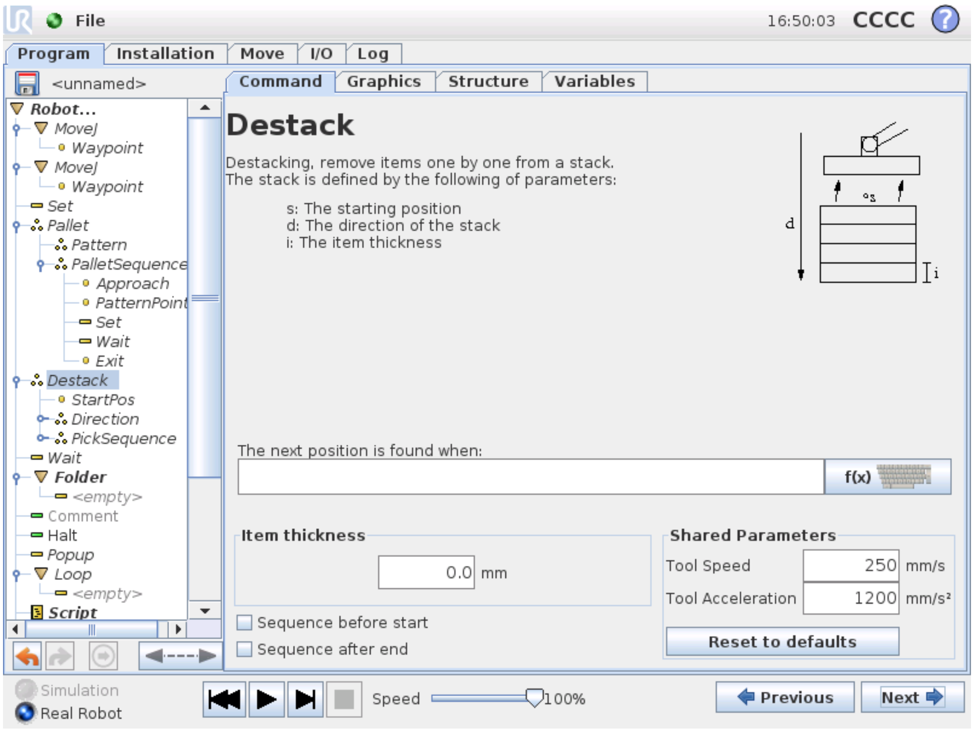
Select Seek Type

A seek operation is given by a starting position *s* and a direction *d*.

Please select between stacking and destacking.

When stacking, the robot arm moves to the starting position, and then moves *opposite* the direction to search for the next stack position. When found, the robot remembers the position and performs the special sequence. The next time round, the robot starts the search from the remembered position incremented by the item thickness along the direction. The stacking is finished when the stack hight is more than some defined number, or when a sensor gives a signal.

Destacking


File
16:50:03 CCCC
Program
Installation
Move
I/O
Log
<unnamed>
Command
Graphics
Structure
Variables
Robot...
MoveJ
Waypoint
MoveJ
Waypoint
Set
Pallet
Pattern
PalletSequence
Approach
PatternPoint
Set
Wait
Exit
Destack
StartPos
Direction
PickSequence
Wait
Folder
<empty>
Comment
Halt
Popup
Loop
<empty>
Script
d
s
i
f(x)
Reset to defaults
Simulation
Real Robot
Speed
100%
Previous
Next

Destack

Destacking, remove items one by one from a stack.
The stack is defined by the following of parameters:

s: The starting position
d: The direction of the stack
i: The item thickness

The next position is found when:

Item thickness: 0.0 mm

Shared Parameters

Tool Speed: 250 mm/s
Tool Acceleration: 1200 mm/s²

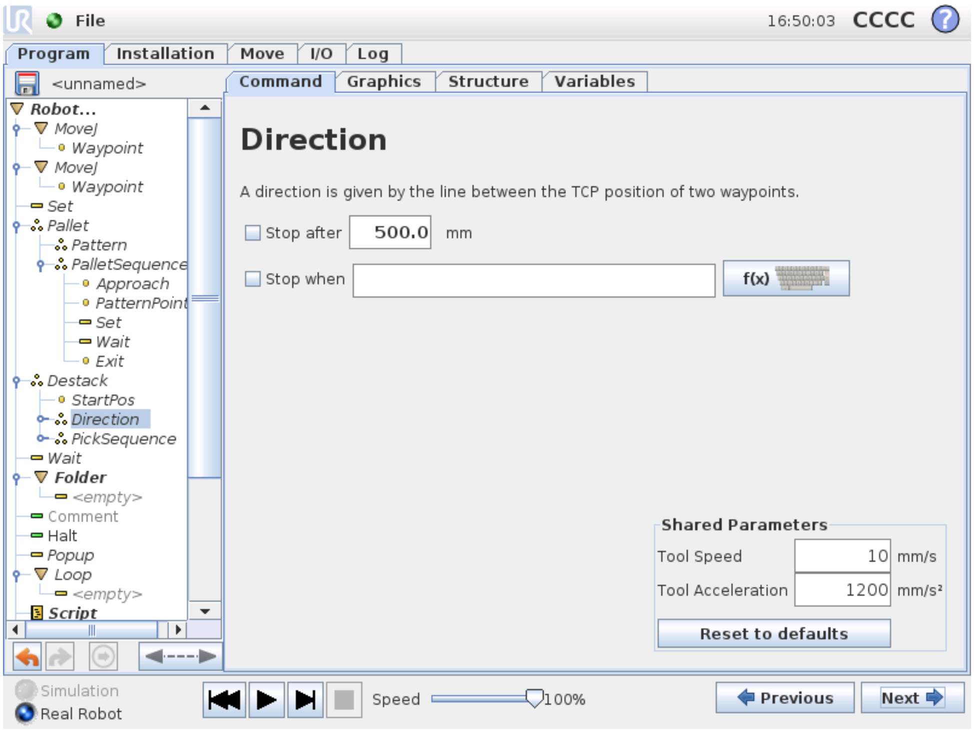
☐ Sequence before start
☐ Sequence after end

When destacking, the robot arm moves from the starting position 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 becomes satisfied, the robot remembers the position and performs the special sequence. The next time round, the robot starts the search from the remembered position, incremented by the item 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



File
16:50:03 CCCC
Program
Installation
Move
I/O
Log
<unnamed>
Command
Graphics
Structure
Variables
Robot...
MoveJ
Waypoint
MoveJ
Waypoint
Set
Pallet
Pattern
PalletSequence
Approach
PatternPoint
Set
Wait
Exit
Destack
StartPos
Direction
PickSequence
Wait
Folder
<empty>
Comment
Halt
Popup
Loop
<empty>
Script
f(x)
Reset to defaults
Simulation
Real Robot
Speed
100%
Previous
Next

Direction

A direction is given by the line between the TCP position of two waypoints.

☐ Stop after 500.0 mm

☐ Stop when

Shared Parameters

Tool Speed 10 mm/s

Tool Acceleration 1200 mm/s²

The direction is given by two positions, and is calculated as the position difference from the first positions TCP to the second positions TCP. Note: A 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

Like for the Pallet operation (13.24), a special program sequence is performed at each stack position.

13.26 Command: Start/Stop Conveyor Tracking

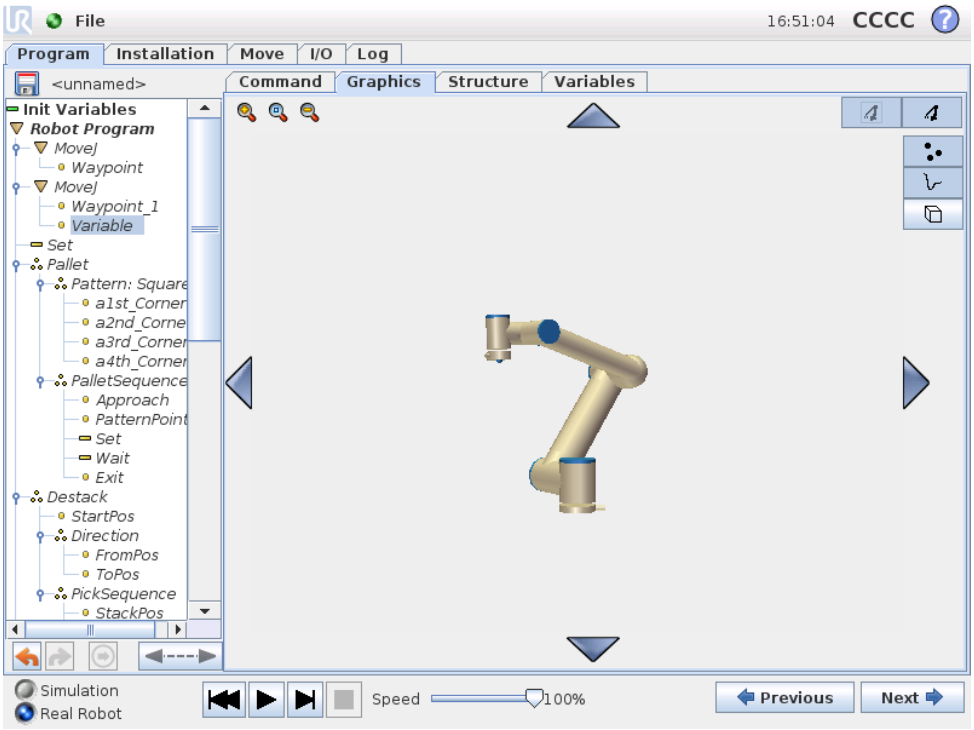
When using a conveyor, the robot can be configured to track the movement of the conveyor. Two program nodes are available for tracking a conveyor, `Start`

`Conveyor Tracking` and `Stop Conveyor Tracking`. When the Conveyor Tracking defined in the installation is configured correctly, a linear or circular conveyor can be tracked. The nodes can be added from the Wizard Program nodes under the Structure tab. When the program is executing between a `Start Conveyor Tracking` and a `Stop Conveyor Tracking` node, the robot adjusts it's movements to follow the conveyor.

13.27 Command: Suppress

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.

13.28 Graphics Tab


File
16:51:04
CCCC
Program
Installation
Move
I/O
Log
<unnamed>
Command
Graphics
Structure
Variables
Init Variables
Robot Program
MoveJ
Waypoint
MoveJ
Waypoint_1
Variable
Set
Pallet
Pattern: Square
a1st_Corner
a2nd_Corne
a3rd_Corner
a4th_Corner
PalletSequence
Approach
PatternPoint
Set
Wait
Exit
Destack
StartPos
Direction
FromPos
ToPos
PickSequence
StackPos
Simulation
Real Robot
Speed
100%
Previous
Next

Graphical representation of the current robot program. The path of the TCP is shown in the 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 15.11), a 3D representation of the proximate boundary limit is shown. Note that when the robot is running a program, the visualization of boundary limits will be disabled.

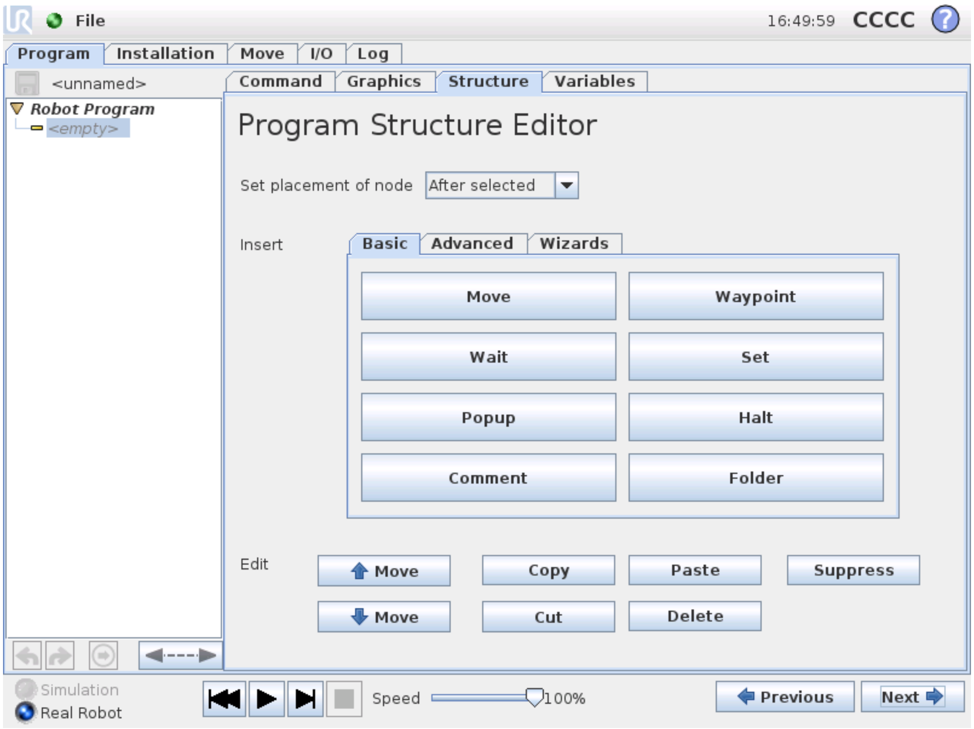
Safety planes are visualized in yellow and black with a small arrow representing the plane normal, which indicates the side of the plane on which the robot TCP is allowed to be positioned. Trigger planes are displayed in blue and green and a small arrow pointing to the side of the plane, where the *Normal* mode limits (see 15.5) 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 the 3D view. The bottom button switches on/off the visualization of proximate boundary limits.

The motion segments shown depend on the selected program node. If a `Move` node is selected, the displayed path is the motion defined by that move. If a `Waypoint` node is selected, the display shows the following ∼ 10 steps of movement.

13.29 Structure Tab



Program Structure Editor

Set placement of node: After selected

File
16:49:59
CCCC
Program
Installation
Move
I/O
Log
<unnamed>
Robot Program
<empty>
Command
Graphics
Structure
Variables
Insert
Basic
Advanced
Wizards
Move
Waypoint
Wait
Set
Popup
Halt
Comment
Folder
Edit
Move
Copy
Paste
Suppress
Move
Cut
Delete
Simulation
Real Robot
Speed
100%
Previous
Next

The program structure tab gives an opportunity for inserting, moving, copying and removing the various types of commands.

To insert new commands, perform the following steps:

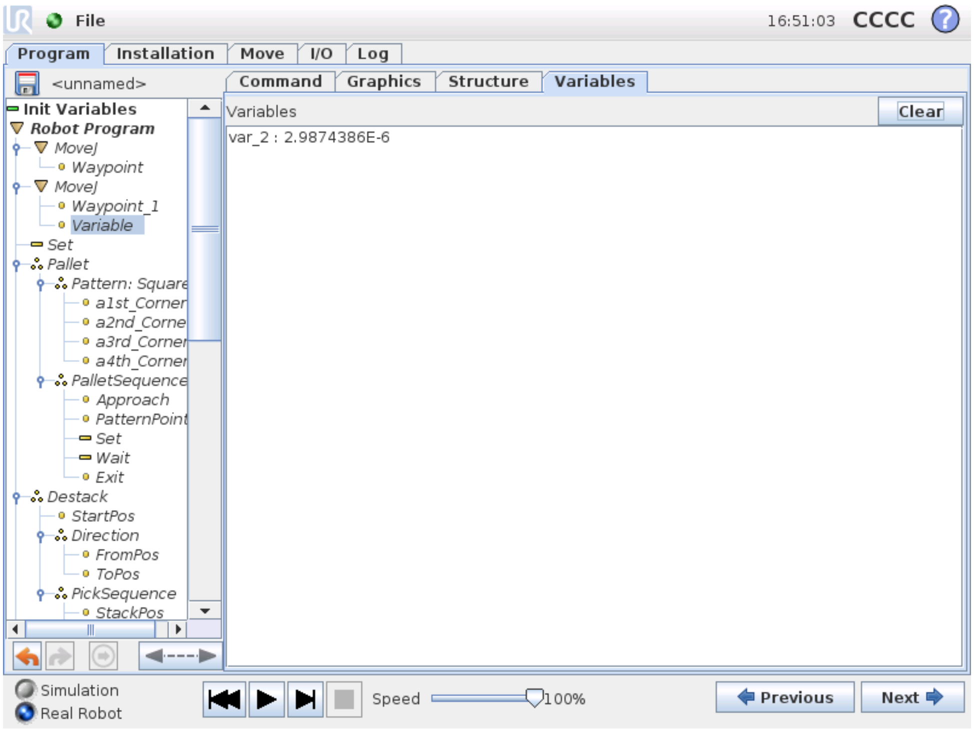
1) Select an existing program command.
2) Select whether the new command should be inserted above or below the selected command.

3) Press the button for the command type you wish to insert. For adjusting the details for the new command, go to the `Command` tab.

Commands can be moved/cloned/deleted using the buttons in the edit frame. If a command has sub-commands (a triangle next to the command) all sub-commands are also moved/cloned/deleted.

Not all commands fit at all places in a program. `Waypoints` must be under a Move command (not necessarily directly under). `ElseIf` and `Else` commands are required to be after an `If`. In general, moving `ElseIf` commands around can be messy. Variables must be assigned values before being used.

13.30 Variables Tab

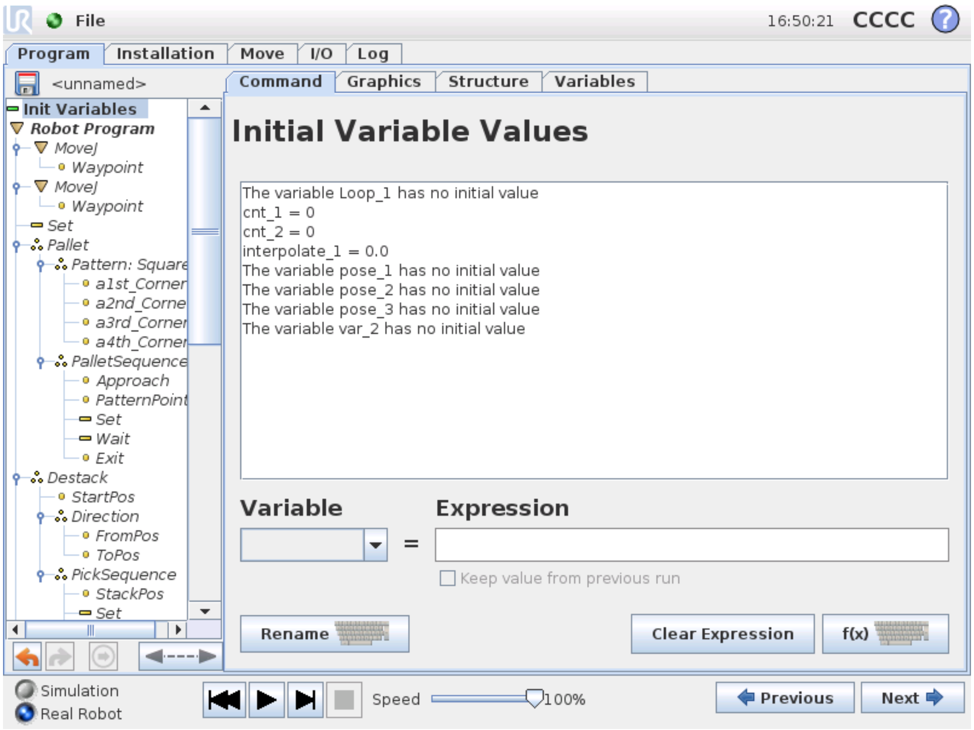

File
16:51:03
CCCC
Program
Installation
Move
I/O
Log
<unnamed>
Command
Graphics
Structure
Variables
Clear
Init Variables
Robot Program
MoveJ
Waypoint
MoveJ
Waypoint_1
Variable
Set
Pallet
Pattern: Square
a1st_Corner
a2nd_Corne
a3rd_Corner
a4th_Corner
PalletSequence
Approach
PatternPoint
Set
Wait
Exit
Destack
StartPos
Direction
FromPos
ToPos
PickSequence
StackPos
Simulation
Real Robot
Speed
100%
Previous
Next

Variables

```
var_2 : 2.9874386E-6
```

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. The variables are ordered alphabetically by their names. The variable names on this screen are shown with at most 50 characters, and the values of the variables are shown with at most 500 characters.

13.31 Command: Variables Initialization


File
16:50:21
CCCC
Program
Installation
Move
I/O
Log
<unnamed>
Init Variables
Robot Program
MoveJ
Waypoint
MoveJ
Waypoint
Set
Pallet
Pattern: Square
a1st_Corner
a2nd_Corne
a3rd_Corner
a4th_Corner
PalletSequence
Approach
PatternPoint
Set
Wait
Exit
Destack
StartPos
Direction
FromPos
ToPos
PickSequence
StackPos
Set
Command
Graphics
Structure
Variables
Initial Variable Values
Variable
Expression
=
☐ Keep value from previous run
Rename
Clear Expression
f(x)
Simulation
Real Robot
Speed
100%
Previous
Next

The variable Loop_1 has no initial value
cnt_1 = 0
cnt_2 = 0
interpolate_1 = 0.0
The variable pose_1 has no initial value
The variable pose_2 has no initial value
The variable pose_3 has no initial value
The variable var_2 has no initial value

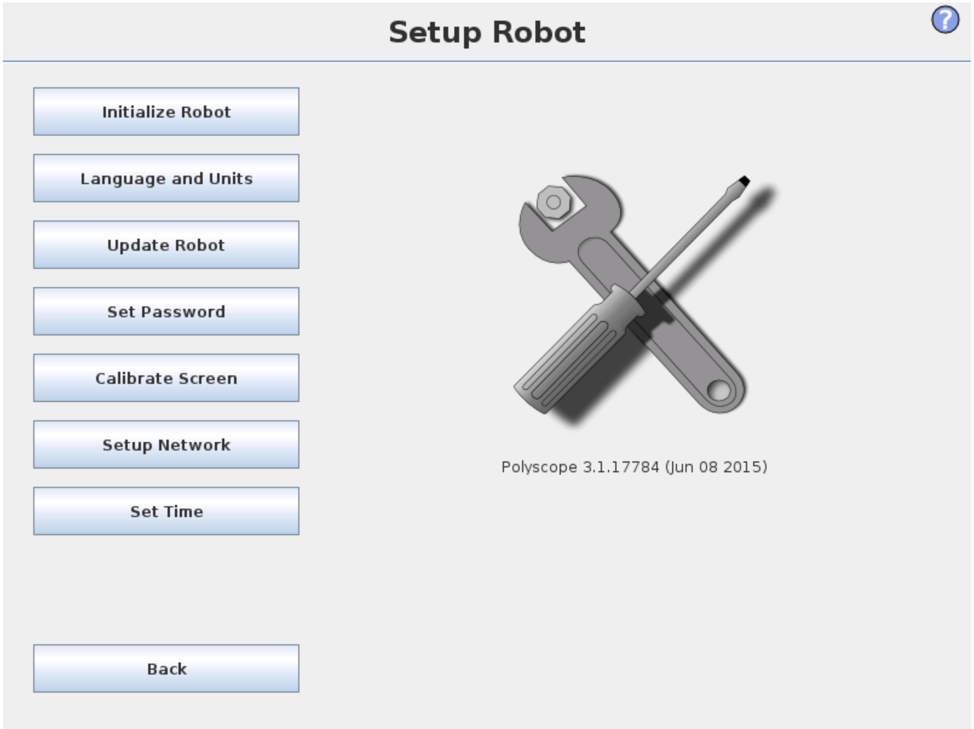
This screen allows setting variable values before the program (and any threads) start executing.

Select a variable from the list of variables by clicking on it, or by using the variable selector box. For a selected variable, an expression can be entered that will be used to set the variable value at program start.

If the “Prefers to keep value from last run” checkbox is selected, the variable will be initialized to the value found on the `Variables` tab, described in 13.30. This permits variables to maintain their values between program executions. The variable will get 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).

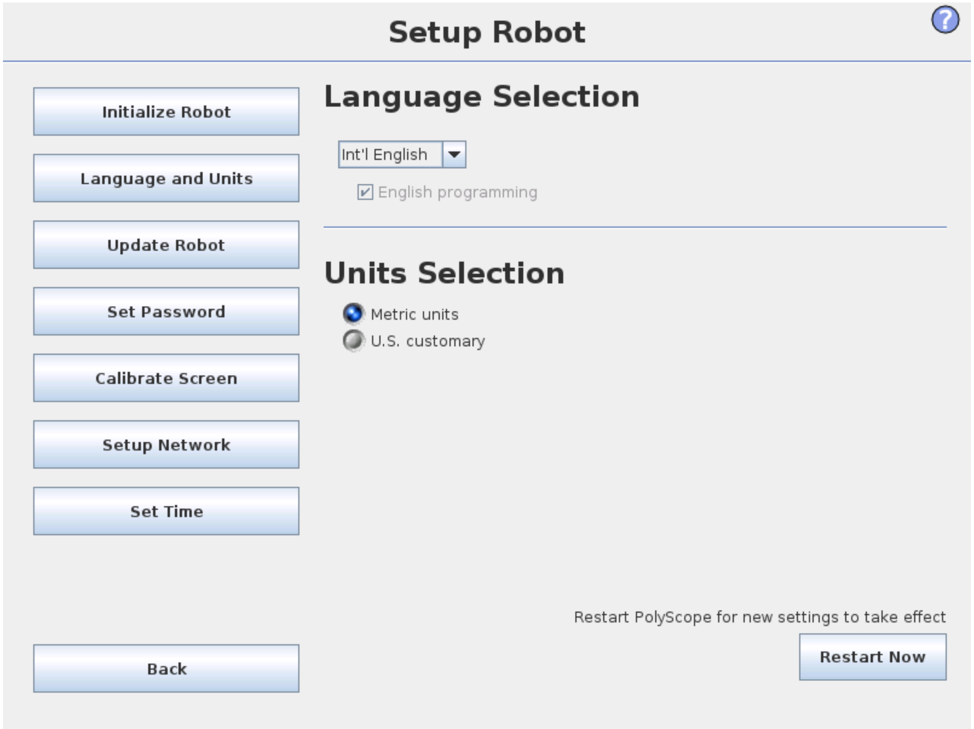
14 Setup Screen


Setup Robot
Initialize Robot
Language and Units
Update Robot
Set Password
Calibrate Screen
Setup Network
Set Time
Back

Polyscope 3.1.17784 (Jun 08 2015)

- **Initialize Robot** Goes to the initialization screen, see 10.4.
- **Language and Units** Configure the language and units of measurements for the user interface, see 14.1.
- **Update Robot** Upgrades the robot software to a newer version, see 14.2.
- **Set Password** Provides the facility to lock the programming part of the robot to people without a password, see 14.3.
- **Calibrate Screen** Calibrates the “touch” of the touch screen, see 14.4.
- **Setup Network** Opens the interface for setting up the Ethernet network for the robot control box, see 14.5.
- **Set Time** Set the time and date for the system and configure the display formats for the clock, see 14.6.
- **Back** Returns to the Welcome Screen.

14.1 Language and Units


Setup Robot
Initialize Robot
Language and Units
Update Robot
Set Password
Calibrate Screen
Setup Network
Set Time
Back
Restart Now

Language Selection

Int'l English

☑ English programming

Units Selection

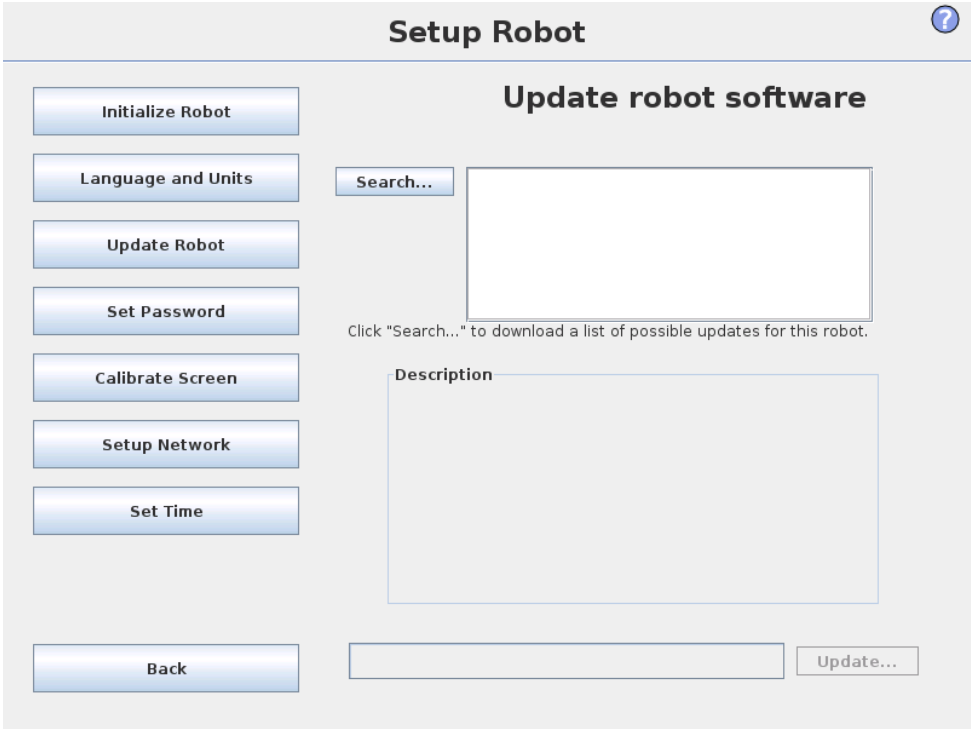
● Metric units

○ U.S. customary

Restart PolyScope for new settings to take effect

Language and units used in PolyScope can be selected on this screen. The selected language will be used for the text visible on the various screens of PolyScope as well as in the embedded help. Tick off "English programming" to have the names of commands within robot programs written in English. PolyScope needs to be restarted for changes to take effect.

14.2 Update Robot

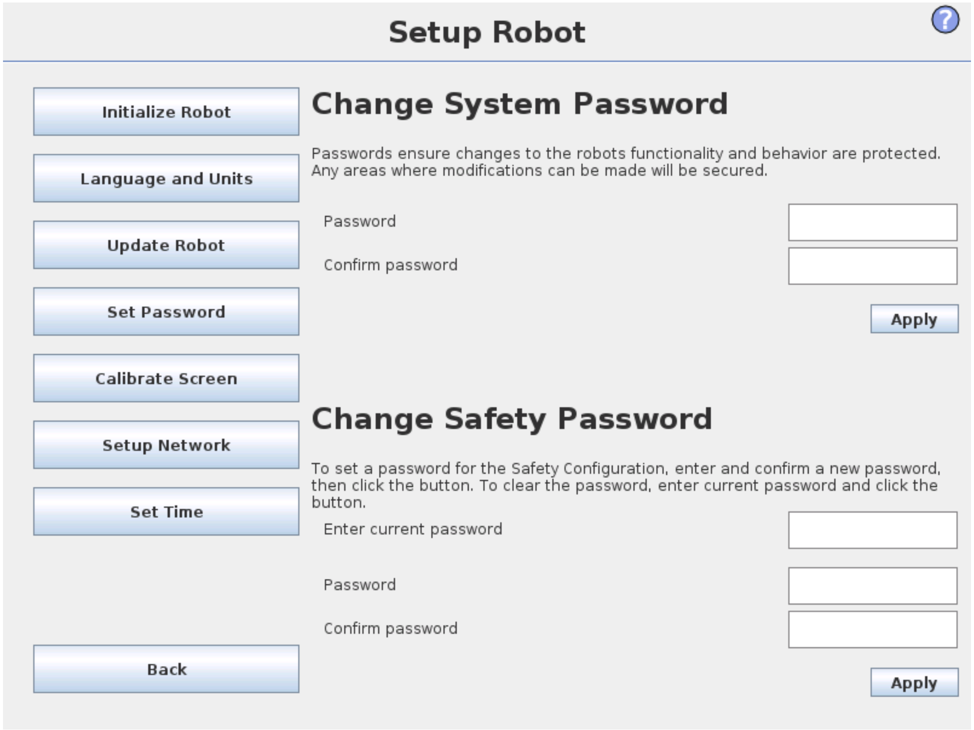

Setup Robot
?
Initialize Robot
Language and Units
Update Robot
Set Password
Calibrate Screen
Setup Network
Set Time
Back
Update robot software
Search...
Description
Update...

Click "Search..." to download a list of possible updates for this robot.

Software updates can be installed from USB flash memory. Insert an USB memory stick and click **Search** to list its contents. To perform an update, select a file, click **Update**, and follow the on-screen instructions.

WARNING:
Always check your programs after a software upgrade. The upgrade might change trajectories in your program. The updated software specifications can be found by pushing the "?" button located at the top right corner of the GUI. Hardware specifications remain the same and can be found in the original manual.

14.3 Set Password


Setup Robot
Initialize Robot
Language and Units
Update Robot
Set Password
Calibrate Screen
Setup Network
Set Time
Back
Apply
Apply

Change System Password

Passwords ensure changes to the robots functionality and behavior are protected. Any areas where modifications can be made will be secured.

Password

Confirm password

Change Safety Password

To set a password for the Safety Configuration, enter and confirm a new password, then click the button. To clear the password, enter current password and click the button.

Enter current password

Password

Confirm password

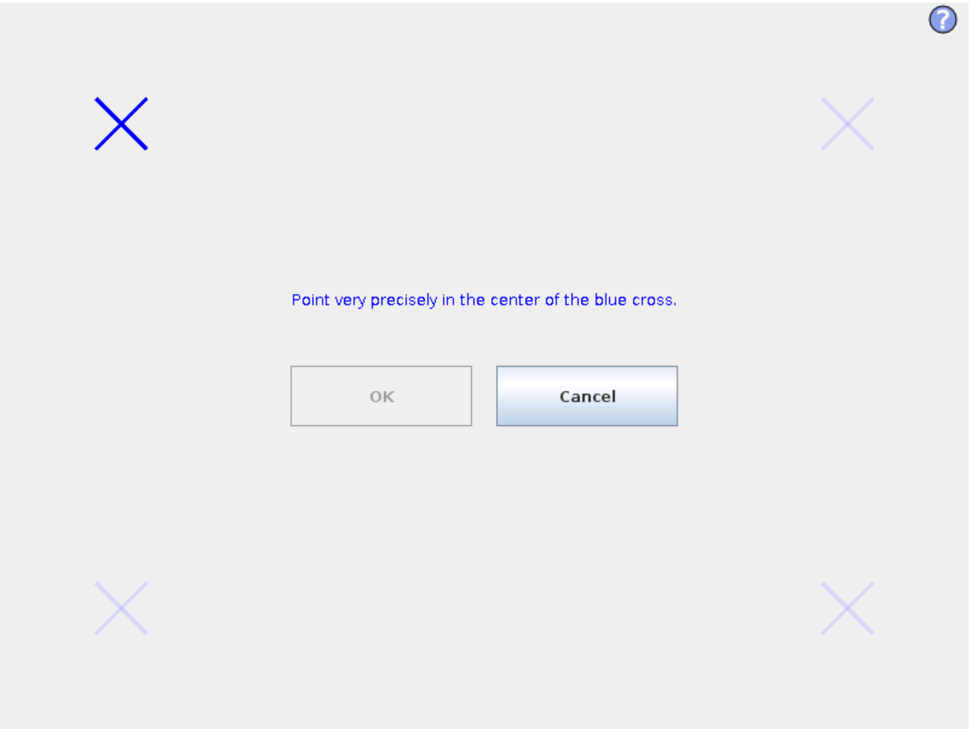
Two passwords are supported. The first is an *optional* System password which prevents unauthorized modification of the setup of the robot. When the System password is set, programs can be loaded and executed without the password, but the user must enter the correct password in order to create or change programs.

The second is a *required* Safety password which must be entered correctly in order to modify the safety configuration.

NOTE:
In order to change the safety configuration, the Safety password must be set.

WARNING:
Add a System password to prevent non-authorized personnel from changing the robot installation.

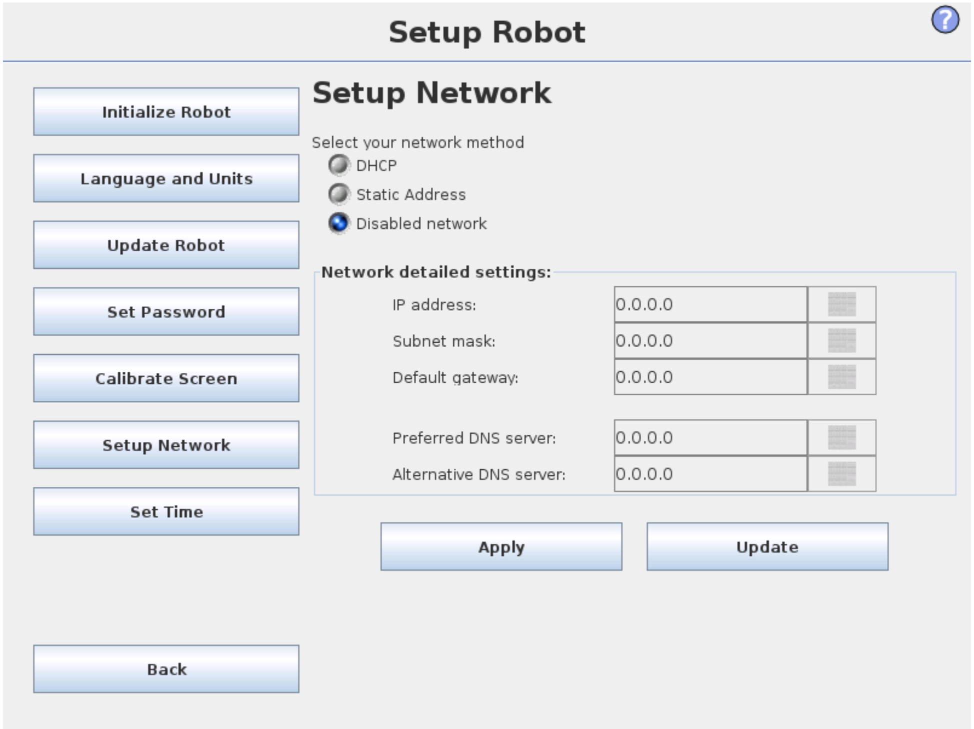
14.4 Calibrate Screen



Point very precisely in the center of the blue cross.

Calibrating the touch screen. Follow the on-screen instructions to calibrate the touch screen. Preferably use a pointed non-metallic object, such as a closed pen. Patience and care help achieve a better result.

14.5 Setup Network



Setup Robot

Setup Network

Select your network method
○ DHCP
○ Static Address
● Disabled network

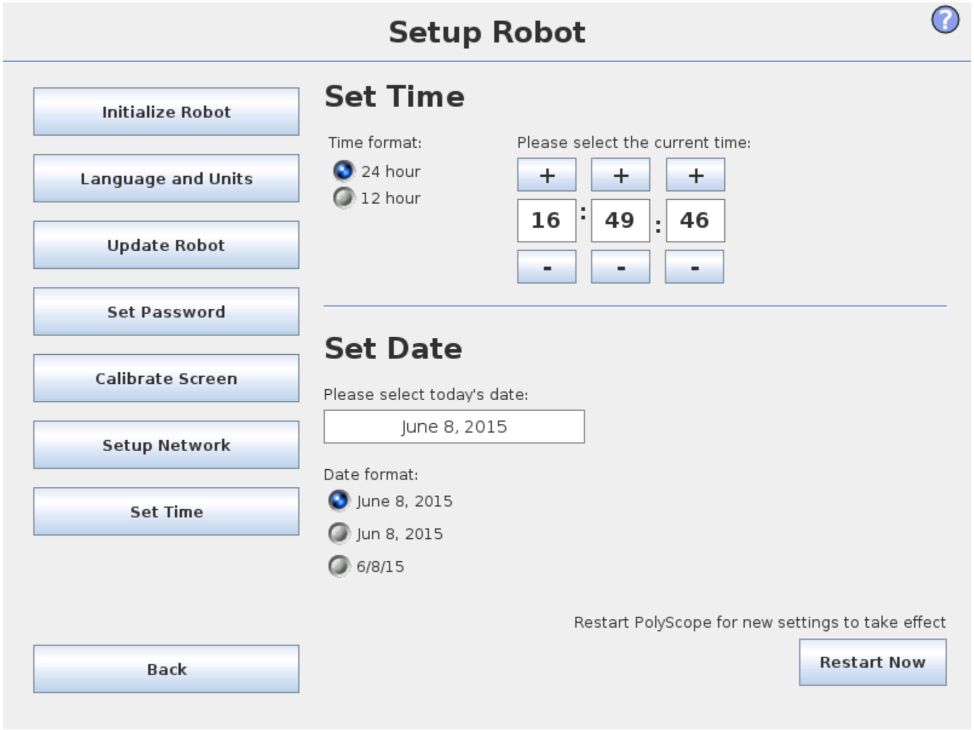
Network detailed settings:

IP address: 0.0.0.0
Subnet mask: 0.0.0.0
Default gateway: 0.0.0.0
Preferred DNS server: 0.0.0.0
Alternative DNS server: 0.0.0.0

Initialize Robot
Language and Units
Update Robot
Set Password
Calibrate Screen
Setup Network
Set Time
Back
Apply
Update

Panel for setting up the Ethernet network. An Ethernet connection is not necessary for the basic robot functions, and is disabled by default.

14.6 Set Time


Setup Robot
Initialize Robot
Language and Units
Update Robot
Set Password
Calibrate Screen
Setup Network
Set Time
Back
+
-
Restart Now

Set Time

Time format:

● 24 hour

○ 12 hour

Please select the current time:

16 : 49 : 46

Set Date

Please select today's date:

June 8, 2015

Date format:

● June 8, 2015

○ Jun 8, 2015

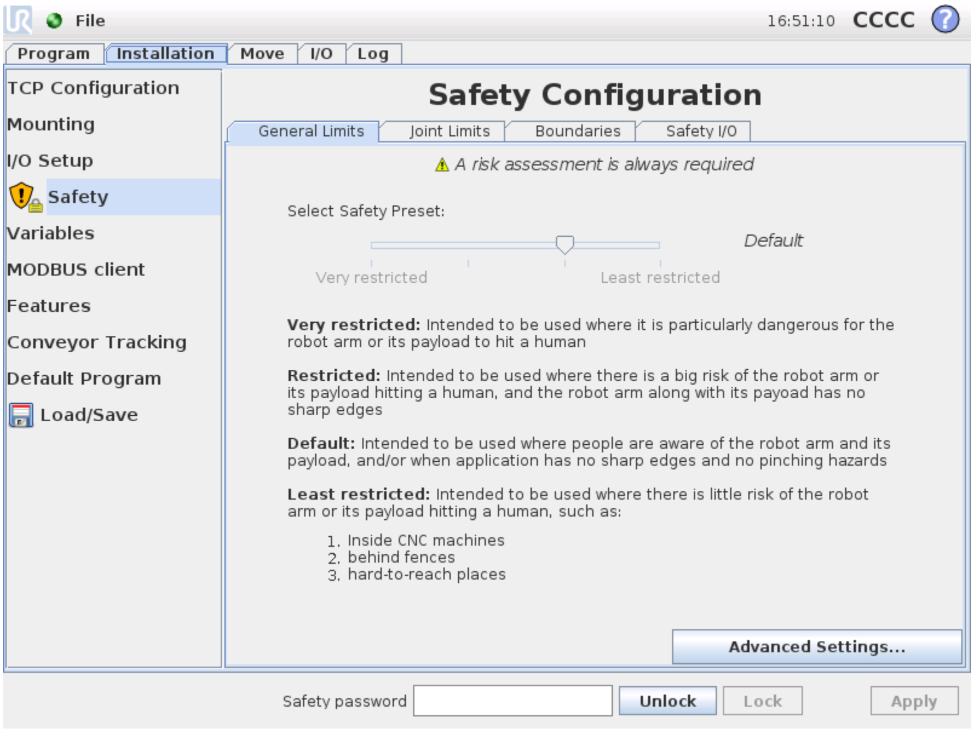
○ 6/8/15

Restart PolyScope for new settings to take effect

Set the time and date for the system and configure the display formats for the clock. The clock is displayed at the top of the *Run Program* and *Program Robot* screens. Tapping on it will show the date briefly. The GUI needs to be restarted for changes to take effect.

15 Safety Configuration

The robot is equipped with an advanced safety system. Depending on the particular characteristics of its workspace, the settings for the safety system must be configured to guarantee the safety of all personnel and equipment around the robot. For details on the safety system, see the Hardware Installation Manual. The `Safety Configuration` screen can be accessed from the `Welcome` screen (see 10.3) by pressing the `Program Robot` button, selecting the `Installation` tab and tapping `Safety`. The safety configuration is password protected, see 15.7.


File
16:51:10
CCCC
Program
Installation
Move
I/O
Log
TCP Configuration
Mounting
I/O Setup
Safety
Variables
MODBUS client
Features
Conveyor Tracking
Default Program
Load/Save
Safety Configuration
General Limits
Joint Limits
Boundaries
Safety I/O
Advanced Settings...
Unlock
Lock
Apply

A risk assessment is always required

Select Safety Preset:

Very restricted — Least restricted: *Default*

Very restricted: Intended to be used where it is particularly dangerous for the robot arm or its payload to hit a human

Restricted: Intended to be used where there is a big risk of the robot arm or its payload hitting a human, and the robot arm along with its payoad has no sharp edges

Default: Intended to be used where people are aware of the robot arm and its payload, and/or when application has no sharp edges and no pinching hazards

Least restricted: Intended to be used where there is little risk of the robot arm or its payload hitting a human, such as:

1. Inside CNC machines
2. behind fences
3. hard-to-reach places

Safety password

WARNING:

1. A risk assessment is always required.
2. All safety settings accessible on this screen and its subtabs are required to be set according to the risk assessment.
3. The integrator is required to ensure that all changes to the safety settings are done in agreement with the risk assessment.

The safety settings consist of a number of limit values used to constrain the movements of the robot arm, and of safety function settings for the configurable inputs and outputs. They are defined in the following subtabs of the safety screen:

- The `General Limits` subtab defines the maximum *force*, *power*, *speed* and *momentum* of the robot arm. When the risk of hitting a human or colliding with a part of its environment is particularly high, these settings need to be set to low values. If the risk is low, higher general limits enable the robot to move faster and exert more force on its environment. For further details, see 15.9.
- The `Joint Limits` subtab consists of *joint speed* and *joint position* limits. The *joint speed* limits define the maximum angular velocity of individual joints and serve to further limit the speed of the robot arm. The *joint position* limits define the allowed position range of individual joints (in joint space). For further details, see 15.10.
- The `Boundaries` subtab defines safety planes (in Cartesian space) and a tool orientation boundary for the robot TCP. The safety planes can be configured either as hard limits for the position of the robot TCP, or triggers for activating the *Reduced* mode safety limits (see 15.5)). The tool orientation boundary puts a hard limit on the orientation of the robot TCP. For further details, see 15.11.
- The `Safety I/O` subtab defines safety functions for configurable inputs and outputs (see 12.2). For example, *Emergency Stop* can be configured as an input. For further details, see 15.12.

15.1 Changing the Safety Configuration

NOTE:
The recommended procedure for changing the safety configuration is as follows:

1. Make a risk assessment.
2. Adjust safety settings to the appropriate level (refer to relevant directives and standards from our manual on how to set the safety limits).
3. Test the setting on the robot.
4. Put the 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 checksum in the top right corner of the PolyScope (see 15.4 in the PolyScope Manual)."

15.2 Safety Synchronization and Errors

The state of the applied Safety configuration in comparison to what robot installation the GUI has loaded, is depicted by the shield icon next to the text `Safety` on the left side of the screen. These icons provide a quick indicator to the current state. They are defined below:

- *Configuration Synchronized*: Shows the GUI installation is identical to the currently applied Safety configuration. No changes have been made.
- *Configuration Altered*: Shows the GUI installation is different from the currently applied Safety configuration.

When editing the Safety configuration, the shield icon will inform you whether or not the current settings have been applied.

If any of the text fields in the `Safety` tab contain any invalid input, the Safety configuration is in an error state. This is indicated in several ways:

- A red error icon is displayed next to the text `Safety` on the left side of the screen.
- The subtab(s) with errors are marked with a red error icon at the top.
- Text fields containing errors are marked with a red background.

When errors exist and attempting to navigate away from the `Installation` tab, a dialog appears with the following options:

1. Resolve the issue(s) so that all errors have been removed. This will be visible when the red error icon is no longer displayed next to the text `Safety` on the left side of the screen.
2. Revert back to the previously applied Safety configuration. This will disregard all changes and allow you to continue to the desired destination.

If no errors exist and attempting to navigate away, a different dialog appears with the following options:

1. Apply changes and restart the system. This will apply the Safety configuration modifications to the system and restart. Note: This does not imply that any changes have been saved; shutdown of the robot at this point will lose all changes to the robot installation including the Safety configuration.
2. Revert back to the previously applied Safety configuration. This will disregard all changes and allow you to continue to the desired selected destination.

15.3 Tolerances

In the *Safety Configuration*, physical limits are set. The input fields for these limits are excluding the tolerances: where applicable tolerances are displayed next to the field. The *Safety System* receives the values from the input fields, and detects any violation of these values. The *Robot Arm* attempts to prevent any violations of the safety system and gives a protective stop by stopping the program execution when the limit minus the tolerance is reached. Note, that this means that a program might not be able to perform motions very close to a limit, e.g. the robot may not be able to obtain the exact maximum speed specified by a joint speed limit or the TCP speed limit.

WARNING:
A risk assessment is always required using the limit values without tolerances.

WARNING:
Tolerances are specific to the version of the software. Updating the software may change the tolerances. Consult the release notes for changes between versions.

15.4 Safety Checksum

The text in the top right corner of the screen gives a shorthand representation of the safety configuration currently used by the robot. When the text changes, this indicates that the current safety configuration has changed as well. Clicking on the checksum displays the details about the currently active safety configuration.

15.5 Safety Modes

Under normal conditions (i.e. when no protective stop is in effect), the safety system operates in one of the following *safety modes*, each with an associated set of safety limits:

Normal mode: The safety mode that is active by default;

Reduced mode: Active when the robot TCP is positioned beyond a *Trigger Reduced mode* plane (see 15.11), or when triggered using a configurable input (see 15.12).

Recovery mode: When the robot arm is in violation of one of the other modes (i.e. *Normal* or *Reduced* mode) and a category 0 stop has occurred, the robot arm will start up in *Recovery* mode. This mode allows the robot arm to be manually adjusted until all violations have been resolved. It is not possible to run programs for the robot in this mode.

WARNING:
Note that limits for *joint position*, *TCP position* and *TCP orientation* are disabled in *Recovery* mode, so take caution when moving the robot arm back within the limits.

The subtabs of the `Safety Configuration` screen enable the user to define separate sets of safety limits for *Normal* and *Reduced* mode. For the tool and joints, *Reduced* mode limits regarding speed and momentum are required to be more restrictive than their *Normal* mode counterparts.

When a safety limit from the active limit set is violated, the robot arm performs a category 0 stop. 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 limited by a fixed limit set that is not customizable by the user. For details about *Recovery* mode limits, see the Hardware Installation Manual.

15.6 Freedrive Mode

When in *Freedrive* mode (see 12.1.5) and the movement of the robot arm comes close to certain limits, the user will feel a repelling force. This force is generated for limits on the position, orientation and speed of the robot TCP and the position and speed of the joints.

The purpose of this repelling force is to inform the user that the current position or speed is close to a limit and to prevent the robot from violating that limit. However, if enough force is applied by the user to the robot arm, the limit can be violated. The magnitude of the force increases as the robot arm comes closer to the limit.

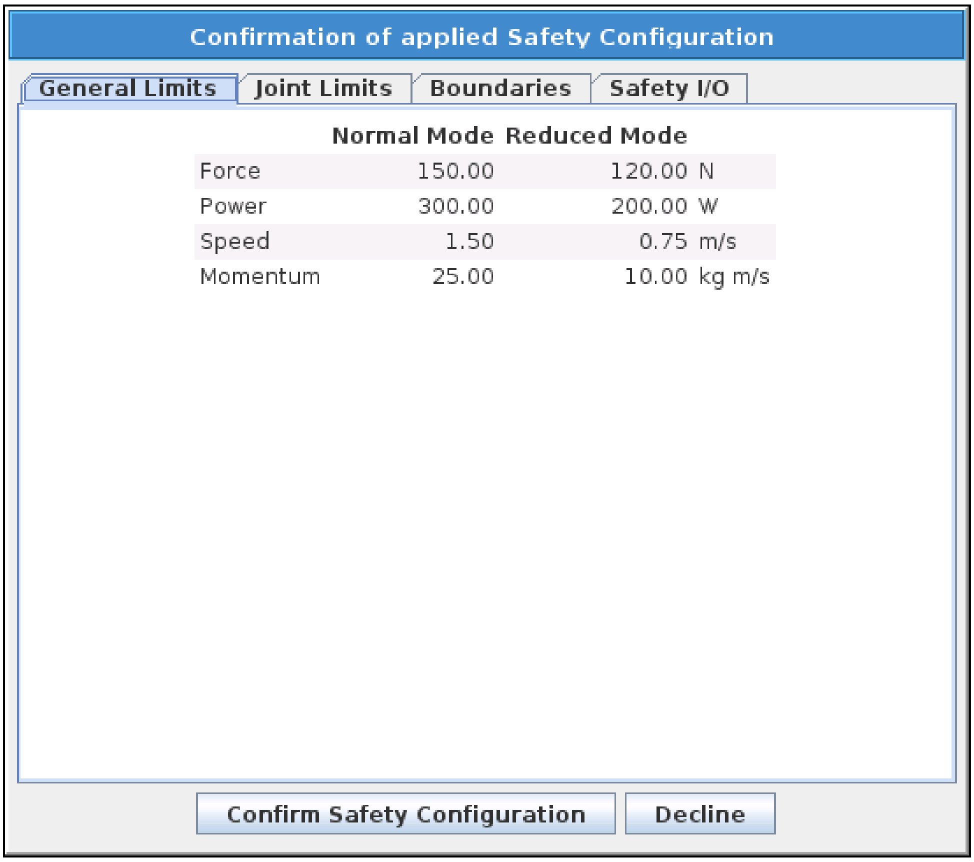
15.7 Password Lock

All settings on this screen are locked until the correct Safety password (see 14.3) is entered in the white text field at the bottom of the screen and the `Unlock` button is pressed. The screen can be locked again by clicking the `Lock` button. The `Safety` tab is automatically locked when navigating away from the Safety Configuration screen. When the settings are locked, a lock icon is visible next to the text `Safety` on the left side of the screen. An unlock icon is shown when the settings are unlocked.

NOTE:
Note that the robot arm is powered off when the Safety Configuration screen is unlocked.

15.8 Apply

When unlocking the safety configuration, the robot arm will be powered off while changes are being made. The robot arm cannot be powered on until the changes have been applied or reverted, and a manual power on is performed from the initialization screen. Any changes to the safety configuration must be applied or reverted, before navigating away from the Installation tab. These changes are *not* in effect until after the `Apply` button is pressed and confirmation is performed. Confirmation requires visual inspection of the changes given to the robot arm. For safety reasons, the information shown is given in SI Units. An example of the confirmation dialog is shown in figure 15.8.


Confirmation of applied Safety Configuration
General Limits
Joint Limits
Boundaries
Safety I/O
Confirm Safety Configuration
Decline

	Normal Mode	Reduced Mode	
Force	150.00	120.00	N
Power	300.00	200.00	W
Speed	1.50	0.75	m/s
Momentum	25.00	10.00	kg m/s

Furthermore, on confirmation the changes are automatically saved as part of the current robot installation. See 12.5 for further information on saving the robot installation.

15.9 General Limits

The general safety limits serve to limit the linear speed of the robot TCP as well as the force it may exert on the environment. They are composed of the following values:

Force: A limit for the maximum force that the robot TCP exerts on the environment.

Power: A limit for the maximum mechanical work produced by the robot on the environment, considering that the payload is part of the robot and not of the environment.

Speed: A limit for the maximum linear speed of the robot TCP.

Momentum: A limit for the maximum momentum of the robot arm.

There are two means available for configuring the general safety limits within the installation; *Basic Settings* and *Advanced Settings* which are described more fully below.

Defining the general safety limits only defines the limits for the tool, and not the overall limits of the robot arm. This means that although a speed limit is specified, it does *not* guarantee that other parts of the robot arm will obey this same limitation.

When in *Freedrive* mode (see 12.1.5), and the current speed of the robot TCP is close to the *Speed* limit, the user will feel a repelling force which increases in magnitude the closer the speed comes to the limit. The force is generated when the current speed is within approximately 250 mm/s of the limit.

Basic Settings The initial general limits subpanel, shown as the default screen, features a slider with the following predefined sets of values for the general limits in both *Normal* and *Reduced* modes:

Very Restricted: Intended to be used where it is particularly dangerous for the robot arm or its payload to hit a human.

Restricted: Intended to be used where there is a high risk of the robot arm or its payload hitting a human, and the robot arm along with its payload has no sharp edges.

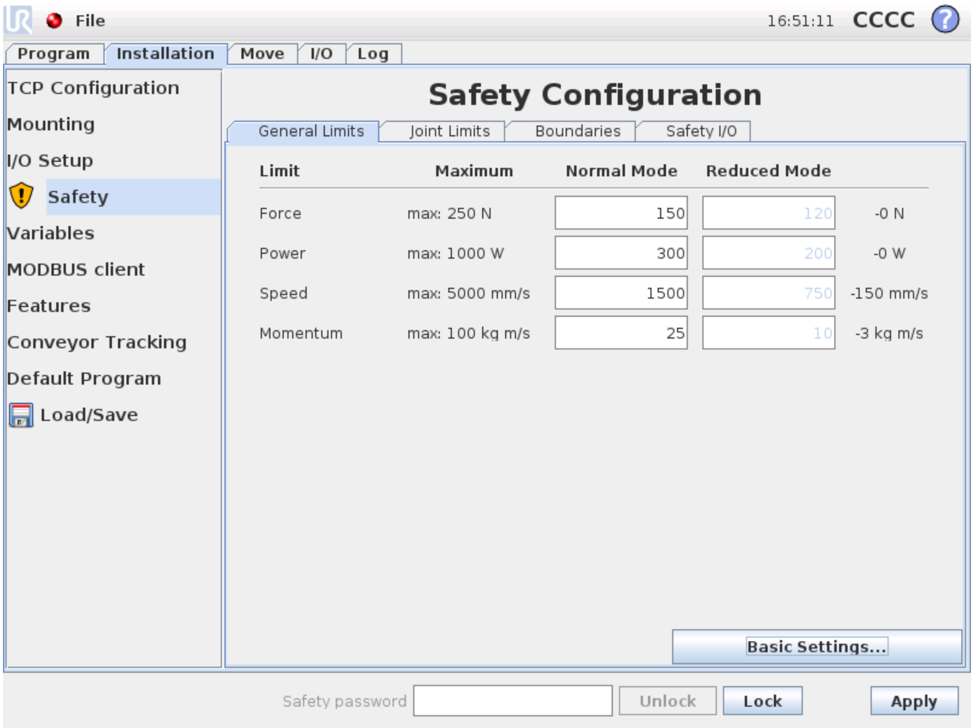
Default: Intended to be used where people are aware of the robot arm and its payload, and/or when application has no sharp edges and no pinching hazards.

Least Restricted: Intended to be used where there is little risk of the robot arm or its payload hitting a human, such as inside CNC machines, behind fences or in hard-to-reach places.

These modes are merely suggestions and a proper risk assessment is always required.

Switching to Advanced Settings Should *none* of the predefined sets of values be satisfactory, the `Advanced Settings...` button can be pressed to enter the advanced general limits screen.

Advanced Settings


File
16:51:11
CCCC
Program
Installation
Move
I/O
Log
TCP Configuration
Mounting
I/O Setup
Safety
Variables
MODBUS client
Features
Conveyor Tracking
Default Program
Load/Save
General Limits
Joint Limits
Boundaries
Safety I/O
Basic Settings...
Safety password
Unlock
Lock
Apply

Safety Configuration

Limit	Maximum	Normal Mode	Reduced Mode	
Force	max: 250 N	150	120	-0 N
Power	max: 1000 W	300	200	-0 W
Speed	max: 5000 mm/s	1500	750	-150 mm/s
Momentum	max: 100 kg m/s	25	10	-3 kg m/s

Here, each of the general limits, described in 15.9, can be modified independently of the others. This is done by tapping the corresponding text field and entering the new value. The highest accepted value for each of the limits is listed in the column titled `Maximum`. The force limit can be set to a value between 100 N and 250 N, and the power limit can be set to a value between 80 W and 1000 W.

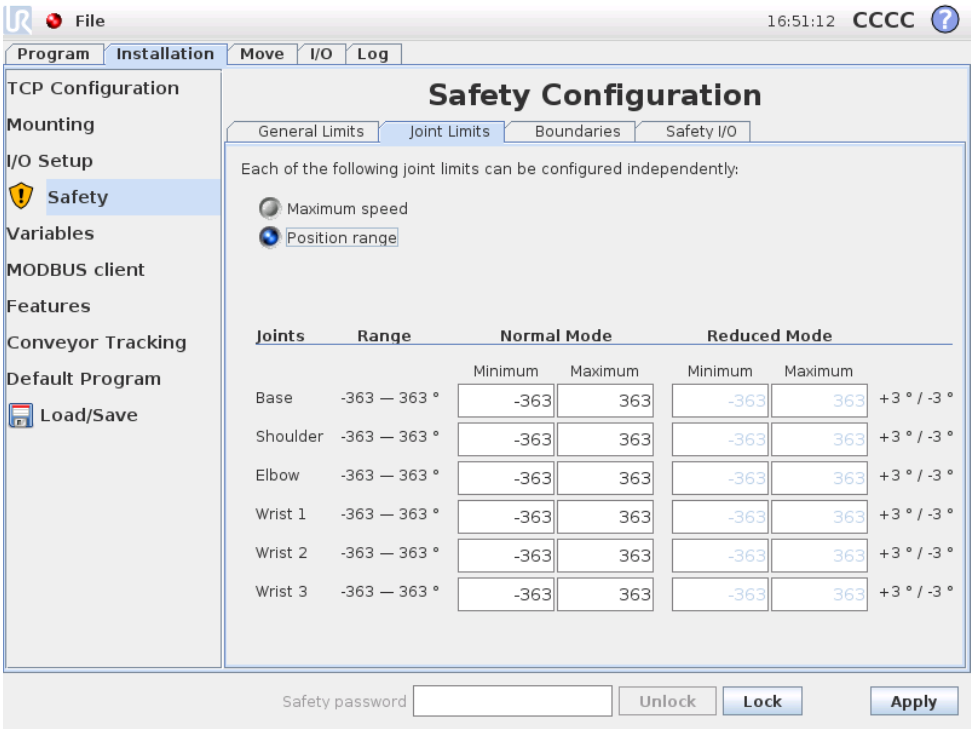
Note that the fields for limits in *Reduced* mode are disabled when neither a safety plane nor a configurable input is set to trigger it (see 15.11 and 15.12 for more details). Furthermore, the *Speed* and *Momentum* limits in *Reduced* mode must not be higher than their *Normal* mode counterparts.

The tolerance and unit for each limit are listed at the end of the row that corresponds to it. When a program is running, the speed of the robot arm is automatically adjusted in order to not exceed any of the entered values minus the tolerance (see 15.3). Note that the minus sign displayed with the tolerance value is only there to indicate that the tolerance is subtracted from the actual entered value. The safety system performs a category 0 stop, should the robot arm exceed the limit (without tolerance).

WARNING:
The speed limit is imposed only on the robot TCP, so other parts of the robot arm may move faster than the defined value.

Switching to Basic Settings Pressing the `Basic Settings...` button switches back to the basic general limits screen and all general limits are reset to their *Default* preset. Should this cause any customized values to be lost, a popup dialog is shown to confirm the action.

15.10 Joint Limits


File
16:51:12 CCCC
Program
Installation
Move
I/O
Log
TCP Configuration
Mounting
I/O Setup
Safety
Variables
MODBUS client
Features
Conveyor Tracking
Default Program
Load/Save
Safety Configuration
General Limits
Joint Limits
Boundaries
Safety I/O
Safety password
Unlock
Lock
Apply

Each of the following joint limits can be configured independently:

○ Maximum speed
● Position range

Joints	Range	Normal Mode		Reduced Mode		
		Minimum	Maximum	Minimum	Maximum	
Base	-363 — 363 °	-363	363	-363	363	+3 ° / -3 °
Shoulder	-363 — 363 °	-363	363	-363	363	+3 ° / -3 °
Elbow	-363 — 363 °	-363	363	-363	363	+3 ° / -3 °
Wrist 1	-363 — 363 °	-363	363	-363	363	+3 ° / -3 °
Wrist 2	-363 — 363 °	-363	363	-363	363	+3 ° / -3 °
Wrist 3	-363 — 363 °	-363	363	-363	363	+3 ° / -3 °

Joint limits restrict the movement of individual joints in joint space, i.e. they do not refer to Cartesian space but rather to the internal (rotational) position of the joints and their rotational speed. The radio buttons in the upper portion of the subpanel make it possible to independently set up `Maximum Speed` and `Position Range` for the joints.

When in *Freedrive* mode (see 12.1.5), and the current position or speed of a joint is close to the limit, the user will feel a repelling force which increases in magnitude as the joint approaches the limit. The force is generated when joint speed is within approximately 20 °/s of the speed limit or joint position is within approximately 8 ° of the position limit.

Maximum Speed This option defines the maximum angular velocity for each joint. This is done by tapping the corresponding text field and entering the new value. The highest accepted value is listed in the column titled `Maximum`. None of the values can be set below the tolerance value.

Note that the fields for limits in *Reduced* mode are disabled when neither a safety plane nor a configurable input is set to trigger it (see 15.11 and 15.12 for more details). Furthermore, the limits for *Reduced* mode must not be higher than their *Normal* mode counterparts.

The tolerance and unit for each limit are listed at the end of the row that corresponds to it. When a program is running, the speed of the robot arm is automatically adjusted in order to not exceed any of the entered values minus the tolerance (see 15.3). Note that the minus sign displayed with each tolerance value is only there to indicate that the tolerance is subtracted from the actual entered value. Nevertheless, should the angular velocity of some joint exceed the entered value (without tolerance), the safety system performs a category 0 stop.

Position Range This screen defines the position range for each joint. This is done by tapping the corresponding text fields and entering new values for the lower and upper joint position boundary. The entered interval must fall within the values listed in the column titled `Range` and the lower boundary cannot exceed the upper boundary.

Note that the fields for limits in *Reduced* mode are disabled when neither a safety plane nor a configurable input is set to trigger it (see 15.11 and 15.12 for more details).

The tolerances and unit for each limit are listed at the end of the row that corresponds to it. The first tolerance value applies to the minimum value and the second applies to the maximum value. Program execution is aborted when the position of a joint is about to exceed the range resulting from adding the first tolerance to the entered minimum value and subtracting the second tolerance from the entered maximum value, if it continues moving along the predicted trajectory. Note that the minus sign displayed with the tolerance value is only there to indicate that the tolerance is subtracted from the actual entered value. Nevertheless, should the joint position exceed the entered range, the safety system performs a category 0 stop.

15.11 Boundaries

In this tab you can configure boundary limits consisting of safety planes and a limit on the maximum allowed deviation of the robot tool orientation. It is also possible to define planes that trigger a transition into *Reduced* mode.

Safety planes can be used to restrict the allowed workspace of the robot by enforcing that the robot TCP stay on the correct side of the defined planes and not pass through them. Up to eight safety planes can be configured. The constraint on the orientation of tool can be utilized to ensure that the robot tool orientation does not deviate more than a certain specified amount from a desired orientation.

WARNING:
Defining safety planes only limits the TCP and not the overall limit for the robot arm. This means that although a safety plane is specified, it does *not* guarantee that other parts of the robot arm will obey this restriction.

The configuration of each boundary limit is based on one of the features defined in the current robot installation (see 12.12).

NOTE:
It is highly recommended, that you create all features needed for the configuration of all the desired boundary limits and assign them appropriate names before editing the safety configuration. Note that since the robot arm is powered off once the *Safety* tab has been unlocked, the `Tool` feature (containing the current position and orientation of the robot TCP) as well as *Freedrive* mode (see 12.1.5) will not be available.

When in *Freedrive* mode (see 12.1.5), and the current position of the robot TCP is close to a safety plane, or the deviation of the orientation of the robot tool from the desired orientation is close to the specified maximum deviation, the user will feel a repelling force which increases in magnitude as the TCP approaches the limit. The force is generated when the TCP is within approximately 5 cm of a safety plane, or the deviation of the orientation of the tool is approximately 3 ° from the specified maximum deviation.

When a plane is defined as a *Trigger Reduced mode* plane and the TCP goes beyond this boundary, the safety system transitions into *Reduced* mode which applies the *Reduced* mode safety settings. Trigger planes follow the same rules as regular safety planes except they allow the robot arm to pass through them.

15.11.1 Selecting a boundary to configure

The `Safety Boundaries` panel on the left side of the tab is used to select a boundary limit to configure.

To set up a safety plane, click on one of the top eight entries listed in the panel. If the selected safety plane has already been configured, the corresponding 3D representation of the plane is highlighted in the `3D View` (see 15.11.2) to the right of this panel. The safety plane can be set up in the `Safety Plane Properties` section (see 15.11.3) at the bottom of the tab.

Click the `Tool Boundary` entry to configure the orientation boundary limit for the robot tool. The configuration of the limit can be specified in the `Tool Boundary Properties` section (see 15.11.4) at the bottom of the tab.

Click the 👁 / 👁 button to toggle the 3D visualization of the boundary limit on/off. If a boundary limit is active, the *safety mode* (see 15.11.3 and 15.11.4) is indicated by one of the following icons 🟩 / ● / 🟩 / ⇥.

15.11.2 3D visualization

The `3D View` displays the configured safety planes and the orientation boundary limit for the robot tool together with the current position of the robot arm. All configured boundary entries where the visibility toggle is selected (i.e. showing 👁 icon) in the `Safety Boundaries` section are displayed together with the current selected boundary limit.

The (active) safety planes are shown 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. A small arrow illustrates the side of the plane that does *not* trigger the transition into *Reduced* mode. If a safety plane has been selected in the panel on the left side of the tab, the corresponding 3D representation is highlighted.

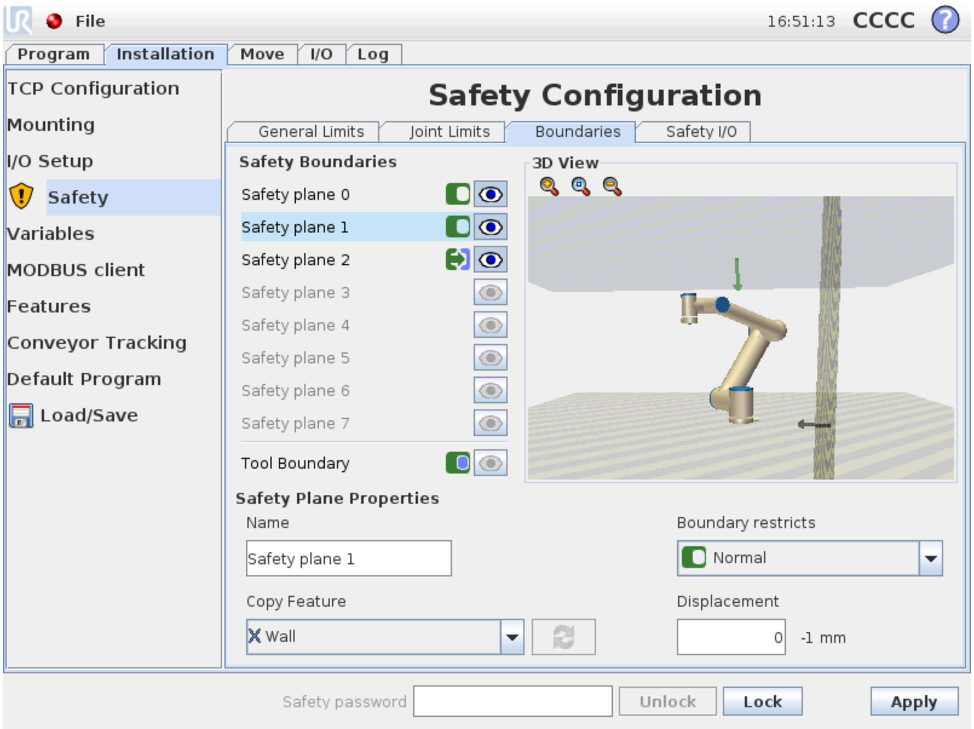
The tool orientation boundary limit is visualized with a spherical cone together with a vector indicating the current orientation of the robot tool. The inside of the cone represents the allowed area for the tool orientation (vector).

When a plane or the tool orientation boundary limit is configured but not active, the visualization is gray.

Push the magnifying glass icons to zoom in/out or drag a finger across to change the view.

15.11.3 Safety plane configuration

The `Safety Plane Properties` section at the bottom of the tab defines the configuration of the selected safety plane in the `Safety Boundaries` panel in the upper left portion of the tab.


File
16:51:13
CCCC
Program
Installation
Move
I/O
Log
TCP Configuration
Mounting
I/O Setup
Safety
Variables
MODBUS client
Features
Conveyor Tracking
Default Program
Load/Save
Safety Configuration
General Limits
Joint Limits
Boundaries
Safety I/O
Safety Boundaries
Safety plane 0
Safety plane 1
Safety plane 2
Safety plane 3
Safety plane 4
Safety plane 5
Safety plane 6
Safety plane 7
Tool Boundary
3D View
Safety Plane Properties
Safety password
Unlock
Lock
Apply

Name: Safety plane 1

Boundary restricts: Normal

Copy Feature: Wall

Displacement: 0 -1 mm

Name The `Name` text field allows the user to assign a name to the selected safety plane. Change the name by tapping the text field and entering a new name.

Copy Feature The position and normal of the safety plane is specified using a feature (see 12.12) from the current robot installation. Use the drop-down box in the lower left portion of the `Safety Plane Properties` section to select a feature. Only the point and plane type features are available. Choosing the `<Undefined>` item clears the configuration of the plane.

The z-axis of the selected feature will point to the disallowed area and the plane normal will point in the opposite direction, except when the `Base` feature is selected, in which case the plane normal will point in the same direction. If the plane is configured as a *Trigger Reduced mode* plane (see 15.11.3), the plane normal indicates the side of the plane that does *not* trigger transition into *Reduced* mode.

It should be noted that when the safety plane has been configured by selecting a feature, the position information is only *copied* to the safety plane; the plane is *not* linked to that feature. This means that if there are changes to the position or orientation of a feature which has been used to configure a safety plane, the safety plane is not automatically updated. If the feature has changed, this is indicated by a ⚠ icon positioned over the feature selector. Click the 🔄 button next to the selector to update the safety plane with the current position and orientation of the feature. The ⚠ icon is also displayed if the selected feature has been deleted from the installation.

Safety mode The drop down menu on the right hand side of the `Safety Plane Properties` panel is used to choose the *safety mode* for the safety plane, with the following modes available:

	`Disabled`	The safety plane is *never active*.
🟩	`Normal`	When the safety system is in *Normal* mode, a `Normal` mode plane is *active* and it acts as a *strict limit* on the position of the robot TCP.
●	`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 of the robot TCP.
🟩	`Normal & Reduced`	When the safety system is either in *Normal* or *Reduced* mode, a `Normal & Reduced` mode plane is *active* and it acts as a *strict limit* on the position of the robot TCP.
➡	`Trigger Reduced mode`	When the safety system is either in *Normal* or *Reduced* mode, a *Trigger Reduced mode* plane is *active* and it causes the safety system to switch to *Reduced* mode for as long as the robot TCP is positioned beyond it.

The selected *safety mode* is indicated by an icon in the corresponding entry in the `Safety Boundaries` panel. If the *safety mode* is set to `Disabled`, no icon is shown.

Displacement When a feature has been selected in the drop down box in the lower left portion of the `Safety Plane Properties` panel, the safety plane can be translated by tapping the `Displacement` text field in the lower right portion of this panel and entering a value. Entering in a positive value increases the allowed workspace of the robot by moving the plane in the opposite direction of the plane normal, while entering a negative value decreases the allowed area by moving the plane in the direction of the plane normal.

The tolerance and unit for the displacement of the boundary plane are shown to the right of the text field.

Effect of *strict limit* planes Program execution is aborted when the TCP position is about to cross an active, strict limit safety plane minus the tolerance (see 15.3), if it continues moving along the predicted trajectory. Note that the minus sign displayed with the tolerance value is only there to indicate that the tolerance is subtracted from the actual entered value. The safety system will perform a category 0 stop, should the TCP position exceed the specified limit safety plane (without tolerance).

Effect of *Trigger Reduced mode* planes When no protective stop is in effect and the safety system is not in the special *Recovery* mode (see 15.5), it operates either in *Normal* or *Reduced* mode and the movements of the robot arm are limited by the respective limit set.

By default, the safety system is in *Normal* mode. It transitions into *Reduced* mode whenever one of the following situations occurs:

a) The robot TCP is positioned beyond some *Trigger Reduced mode* plane, i.e. it is located on the side of the plane that is *opposite to* the direction of the small arrow in the visualization of the plane.

b) The `Reduced Mode` safety input function is configured and the input signals are low (see 15.12 for more details).

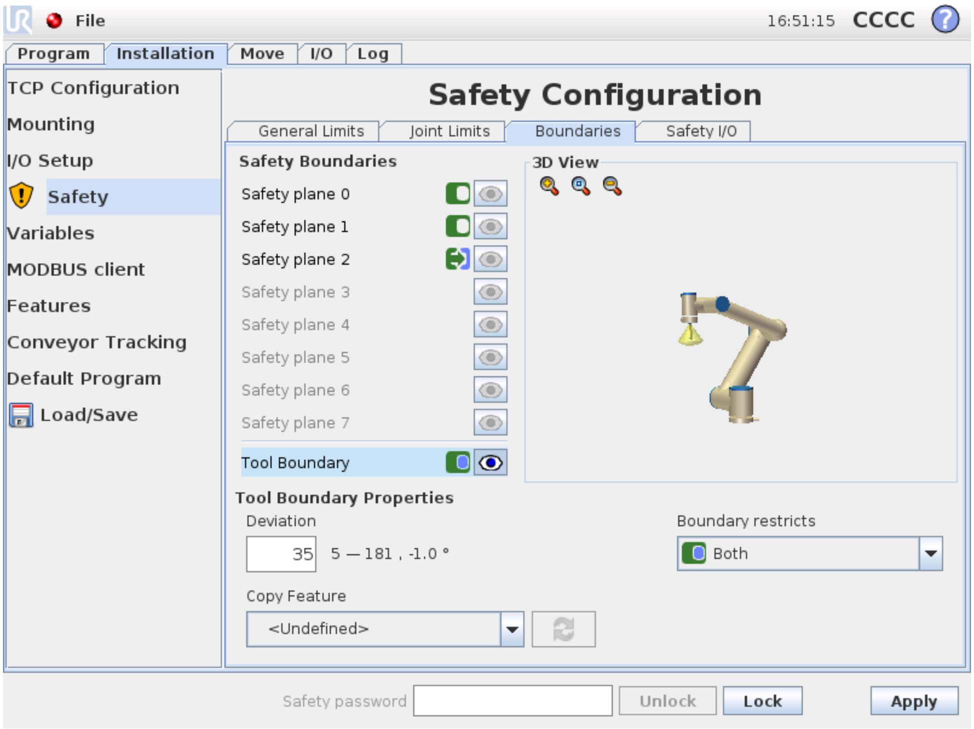
When none of the above is the case any longer, the safety system transitions back to *Normal* mode.

When the transition from *Normal* to *Reduced* mode is caused by passing through a *Trigger Reduced mode* plane, a transition from the *Normal* mode limit set to the *Reduced* mode limit set occurs. As soon as the robot TCP is positioned 20 mm or closer to the *Trigger Reduced mode* plane (but still on the *Normal* mode side), the more permissive of the *Normal* and *Reduced* mode limits is applied for each limit value. Once the robot TCP passes through the *Trigger Reduced mode* plane, the *Normal* mode limit set is no longer active and the *Reduced* mode limit set is enforced.

When a transition from *Reduced* to *Normal* mode is caused by passing through a *Trigger Reduced mode* plane, a transition from the *Reduced* mode limit set to the *Normal* mode limit set occurs. As soon as the robot TCP passes through the *Trigger Reduced mode* plane, the more permissive of the *Normal* and *Reduced* mode limits is applied for each limit value. Once the robot TCP is positioned 20 mm or further from the *Trigger Reduced mode* plane (on the *Normal* mode side), the *Reduced* mode limit set is no longer active and the *Normal* mode limit set is enforced.

If the predicted trajectory takes the robot TCP through a *Trigger Reduced mode* plane, the robot arm will start decelerating even before passing through the plane if it is about to exceed joint speed, tool speed or momentum limit in the new limit set. Note that since these limits are required to be more restrictive in the *Reduced* mode limit set, such premature deceleration can occur only when transitioning from *Normal* to *Reduced* mode.

15.11.4 Tool Boundary configuration


File
16:51:15
CCCC
Program
Installation
Move
I/O
Log
TCP Configuration
Mounting
I/O Setup
Safety
Variables
MODBUS client
Features
Conveyor Tracking
Default Program
Load/Save
Safety Configuration
General Limits
Joint Limits
Boundaries
Safety I/O
Safety Boundaries
Safety plane 0
Safety plane 1
Safety plane 2
Safety plane 3
Safety plane 4
Safety plane 5
Safety plane 6
Safety plane 7
Tool Boundary
3D View
Tool Boundary Properties
Safety password
Unlock
Lock
Apply

Deviation: 35 5 — 181 , -1.0 °

Boundary restricts: Both

Copy Feature: <Undefined>

The `Tool Boundary Properties` panel at the bottom of the tab defines a limit on the orientation of robot tool composed of a desired tool orientation and a value for the maximum allowed deviation from this orientation.

Deviation The `Deviation` text field shows the value for the maximum allowed deviation of the orientation of the robot tool from the desired orientation. Modify this value by tapping the text field and entering the new value.

The accepted value range together with the tolerance and unit of the deviation are listed next to the text field.

Copy Feature The desired orientation of the robot tool is specified using a feature (see 12.12) from the current robot installation. The z-axis of the selected feature will be used as the desired tool orientation vector for this limit.

Use the drop down box in the lower left portion of the `Tool Boundary Properties` panel to select a feature. Only the point and plane type features are available. Choosing the `<Undefined>` item clears the configuration of the plane.

It should be noted that when the limit has been configured by selecting a feature, the orientation information is only *copied* to the limit; the limit is *not* linked to that feature. This means that if there are changes to the position and orientation of a feature, which has been used to configure the limit, the limit is not automatically updated. If the feature has changed, this is indicated by a ⚠ icon positioned over the feature selector. Click the 🔄 button next to the selector to update the limit with the current orientation of the feature. The ⚠ icon is also displayed if the selected feature has been deleted from the installation.

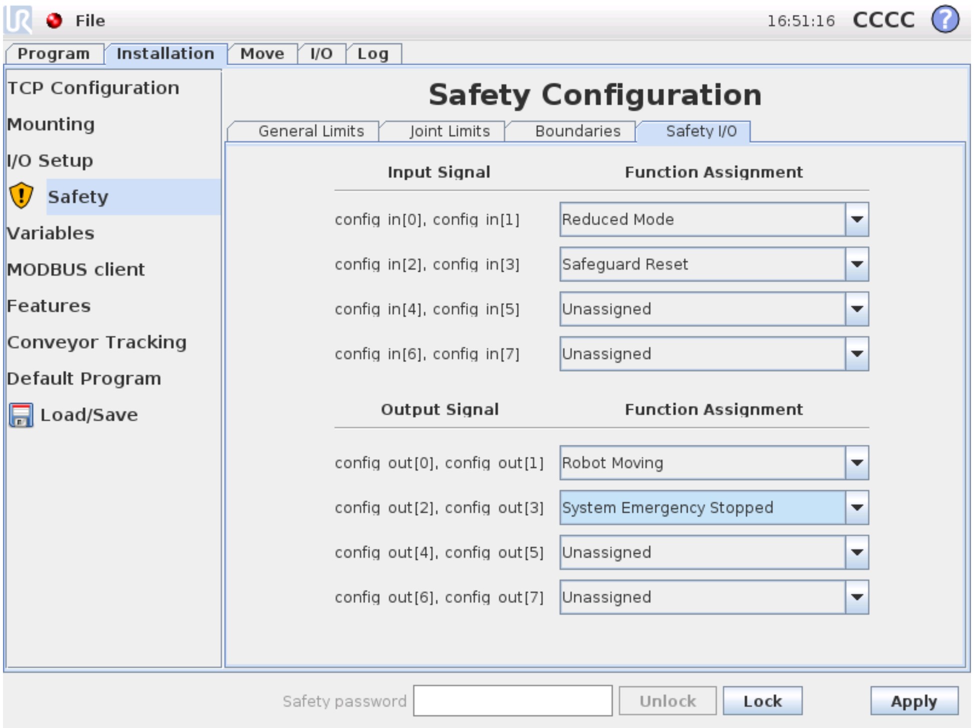
Safety mode The drop down menu on the right hand side of the `Tool Boundary Properties` panel is used to choose the *safety mode* for the tool orientation boundary. The available options are:

	`Disabled`	The tool boundary limit is never active.
🟩	`Normal`	When the safety system is in *Normal* mode, the tool boundary limit is active.
🔵	`Reduced`	When the safety system is in *Reduced* mode, the tool boundary limit is active.
🟩🔵	`Normal & Reduced`	When the safety system is either in *Normal* or *Reduced* mode, the tool boundary limit is active.

The selected *safety mode* is indicated by an icon in the corresponding entry in the `Safety Boundaries` panel. If the *safety mode* is set to `Disabled`, no icon is shown.

Effect Program execution is aborted when the deviation of the tool orientation is about to exceed the entered maximum deviation minus the tolerance (see 15.3), if it continues moving along the predicted trajectory. Note that the minus sign displayed with the tolerance value is only there to indicate that the tolerance is subtracted from the actual entered value. The safety system will perform a category 0 stop, should the deviation of the tool orientation exceed the limit (without tolerance).

15.12 Safety I/O


File
16:51:16
CCCC
Program
Installation
Move
I/O
Log
TCP Configuration
Mounting
I/O Setup
Safety
Variables
MODBUS client
Features
Conveyor Tracking
Default Program
Load/Save
Safety Configuration
General Limits
Joint Limits
Boundaries
Safety I/O
Safety password
Unlock
Lock
Apply

Input Signal	Function Assignment
config in[0], config in[1]	Reduced Mode
config in[2], config in[3]	Safeguard Reset
config in[4], config in[5]	Unassigned
config in[6], config in[7]	Unassigned

Output Signal	Function Assignment
config out[0], config out[1]	Robot Moving
config out[2], config out[3]	System Emergency Stopped
config out[4], config out[5]	Unassigned
config out[6], config out[7]	Unassigned

This screen defines the *Safety functions* for configurable inputs and outputs (I/Os). The I/Os are divided between the inputs and outputs, and are paired up so that each function is providing a Category 3 and PLd I/O.

Each *Safety function* can only control one pair of I/Os. Trying to select the same safety function a second time removes it from the first pair of I/Os previously defined. There are 3 *Safety functions* for input signals, and 4 for output signals.

Input Signals For input signals, the following *Safety functions* can be selected:

- `Emergency Stop`: When selected, this allows the option of having an alternative `Emergency Stop` button in inclusion of the one that is on the Teach Pendant. This will provide the same functionality that the `Emergency Stop` button provides on the Teach Pendant when a device complying with ISO 13850:2006 is attached.
- `Reduced Mode`: All safety limits have two modes in which they can be applied: *Normal* mode, which specifies the default safety configuration, and *Reduced* mode (see 15.5 for more details). When this input safety function is selected, a low signal given to the inputs causes the safety system to transition to *Reduced* mode. If necessary, the robot arm then decelerates to satisfy the *Reduced* mode limit set. Should the robot arm still violate any of the *Reduced* mode limits, it performs a category 0 stop. The transition back to *Normal* mode happens in the same manner. Note that safety planes can also cause a transition to *Reduced* mode (see 15.11.3 for more details).

- `Safeguard Reset`: If `Safeguard Stop` is wired in the safety I/Os, then `Safeguard Reset` is used to ensure the Safeguard Stopped state continues until a reset is triggered. The robot arm will not move when in Safeguard Stopped state.

WARNING:
By default, the `Safeguard Reset` input function is configured for input pins 0 and 1. Disabling it altogether implies that the robot arm ceases to be Safeguard Stopped as soon as the `Safeguard Stop` input becomes high. In other words, without a `Safeguard Reset` input, the `Safeguard Stop` inputs SI0 and SI1 (see the Hardware Installation Manual) fully determine whether the Safeguard Stopped state is active or not.

Output Signals For the output signals the following *Safety functions* can be applied. All signals return to low when the state which triggered the high signal has ended:

- `System Emergency Stop`: Low signal is given when the safety system has been triggered into an `Emergency Stopped` state. It is in a high signal state otherwise.
- `Robot Moving`: A low signal is given whenever the robot arm is in a mobile state. When the robot arm is in a fixed position, a high signal is given.
- `Robot Not Stopping`: When the robot arm has been requested to stop, some time will pass from the request until the arm stops. During this time the signal will be high. When the robot arm is moving and has not been requested to stop, or when the robot arm is in a stopped position, the signal will be low.
- `Reduced Mode`: Sends a low signal 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.

Glossary

Category 0 stop: 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 EN ISO13850:2008 or IEC60204-1:2006.

Category 1 stop: 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 after one second or as soon as the robot stands still. For more information, see EN ISO13850:2008 or IEC60204-1:2006.

Category 2 stop: A controlled stop with power left available to the robot. The robot can use up to one second to stop all motion. The safety-related control system monitors that the robot stays at the stop position. For more information, see IEC 60204-1:2006.

Diagnostic coverage (DC): is a measure of the effectiveness of the diagnostics implemented to achieve the rated performance level. For more information, see EN ISO13849-1:2008.

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.

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 EN ISO13849-1:2008.

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.

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 EN ISO13849-1:2008.

Index