UNIVERSAL ROBOTS

User Manual

Version 1.6

Robot:
UR5 with CB2
US Version

Contents

Chapter 1

Safety

1.1 Introduction

This chapter gives a short introduction to the statutory requirements and important information about the risk assessment, followed by a section concerning emergency stop and emergency movement of the robot arm. All mounting instructions in section 2.5 shall be followed. Technical specifications of the electrical safety interface, including performance level and safety categories, are found in section 3.2.
Special attention shall be paid to the texts marked with warning symbols. The following symbols are used throughout the manual and can be found 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.

This chapter is intended for integrators with a technical understanding of mechanics and electronics. The guidance provided in the manual assumes that the integrator is based in United States of America (US) and that the robot is to be installed within the US. Contact your supplier if guidance for other countries is needed.
The robot and the documentation are designed for industrial applications. Do not use the robot for medical or terror purposes.

1.2 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:
1. Make sure to install the robot and all electrical equipment according to the specifications and warnings found in the sections "mounting instructions" and "electrical interface".

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 (e.g. guardrail, rope or safety screen) has been set up around the robot operating area to protect both the operator and bystandards.

4. Do not enter the safety range of the robot or touch the robot when the system is in operation. Before approaching the robot make sure the robot arm is powered off.

5. Do not wear loose clothing or jewelry when working with the robot. Make sure long hair is tied back when working with the robot.

WARNING:

6. Never use the robot if it is damaged.

7. If the software prompts a fatal error, immediately activate emergency stop, write down the conditions that lead to the error, find the corresponding error codes on the log screen, and contact your supplier.

8. Do not connect any safety equipment to normal I/O. Use the emergency stop and safeguard stop interfaces only.

9. Make sure to use the correct installation sittings (e.g. Robot mounting angle, weight in TCP, TCP offset). Save and load the installations file along with the program.

10. The teach 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 people have their heads and faces kept outside the reach of the robot.

11. The robot force limitation does not give protection against momentum. Any collision will release high portions of kinetic energy which are significantly higher at high speeds and with high payloads. (Kinetic Energy = Mass * 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. The module manual is usually attached as an appendix to this user 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.

1.3 Statutory requirements

The robot is a component in a robot installation. It cannot be considered a complete machine because it requires an installation including a tool. When a specific robot installation is constructed it is very important to make a risk assessment of the complete robot installation. Guidance on risk assessment is given in the next sub-chapter 1.4.
The integrator shall ensure that all personnel are protected against all serious hazards. OSHA (Occupational Safety & Health Administration) is an agency of the United States Department of Labor and OSHA makes rules and standards for machine safety. Two OSHA standards relevant for safety of robot installations are listed below:

1. OSHA 29 CFR 1910.333, Selection and Use of Work Practices
2. OSHA 29 CFR 1910.147, The Control of Hazardous Energy (Lockout/Tagout)

All electrical installations shall be constructed to prevent electrical shock and the complete robot installation shall have a lockout/tagout function. Lockout/tagout refers to when all power to the machinery in the robot installation can be disconnected and that the disconnecting switch can be locked in the off position; preventing other people from re-powering the system. The integrator is responsible for installing the lockout/tagout function. For more information go to http://www.osha.gov

Other safety standards exist besides the OSHA standards:

1. ANSI/RIA 15.06-2012 "American national standard - Safety requirements"

2. ISO 10218-1:2011 "Robots and robotic devices - Safety requirements for industrial robots - Part 1: Robot"
3. ISO 10218-2:2011 "Robots and robotic devices - Safety requirements for industrial robots - Part 2: Robot systems and integration"

Note that these standards describe what is normally done to sufficiently reduce hazards. For hazardous and complex robot installations it is recommended to follow the relevant guidelines of R15.06-2012.
The integrator is always responsible for the following:

1. Making a risk assessment of the complete robot installation.
2. Installing a lockout/tagout function for the complete robot installation.
3. Writing a user manual for the customer and/or operators.
4. Providing a warning for any foreseeable (imaginable) dangers.
5. Installing the robot in accordance with the recommendations of the manufacturer.
6. Knowing and applying all relevant national and regional requirements.

1.4 Risk assessment

One of the most important things that an integrator needs to do is to conduct a risk assessment. A risk assessment is explained as below.

1. A risk assessment is:
   The overall process comprising a risk analysis and a risk evaluation.
2. A risk analysis is:
   The combination of the specification of the limits of the machine, hazards identification and defining likely severity of harm and probability of it occurrence. This includes foreseeable misuse.
3. A risk evaluation is:
   The judgment, on the basis of risk analysis, of whether the risk reduction objectives have been achieved.

More guidance on risk assessment is found in the international standard ISO 12100:2010 and at http://www.osha.gov. The risk assessment shall be documented for future reference.
Universal Robots has identified the potential significant hazards listed below as hazards which shall be considered by the integrator. Note that other significant hazards might be present in a specific robot installation.

1. Fingers caught between robot mounting flange and base (joint 0).
2. Fingers caught between the robot arm and robot wrist (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. Items falling out of tool. E.g. due to a poor grip or power interruption.
8. Electrical shock or fire due to malfunction of power supplies if the mains connection is not protected by a main fuse, a residual current device and a proper connection to earth.
9. Mistakes due to different emergency stop bottons for different machines. Use common emergency stop function as described in section 3.2.1.

However, the UR5 is a very safe robot due to the following reasons:

1. Control system conforms to ISO 13849-1 performance level **d**.
2. The control system of the robot is redundant so that all dangerous failures forces the robot to enter a safe condition.
3. High level software generates a protective stop if the robot hits something. This stop force limit is lower than $150N$.
4. Low level software limits the torque generated by the joints, permitting only a small deviation from the expected torque.
5. The software prevents program execution when the robot is mounted differently than specified in the setup.
6. The weight of the robot is less than $18kg$.
7. The robot shape is smooth, to reduce pressure (N/m^2) per force (N).
8. It is possible to move the joints of an unpowered robot. See section 1.6

The fact that the robot is very safe opens the possibility of either saving the safety guards or using safety guards with a low performance level, depending on the robot installation. As a help for the integrator when he is conducting the risk assessment, the robot has been third party certified by the Danish Technological Institute (TI). TI is a Notified Body under the Machinery Directive in EU. This means that TI has the highest authority within EU to certifying machines. This third party certification can also be used when conducting risk assessment for installations within US.
The TI certification concludes that the UR robots cannot exceed a force of more than $150N$. The certification is performed in accordance with the international ISO 10218 standards and thereby also in accordance with the R15.06-2012. Find the certificate in the chapter A.

1.5 Emergency stop

To immediately abort all running programs and stop all robot movement, press the EMERGENCY STOP button.
The emergency stop function is a separate circuit only intended for emergency situations. The function is not intended to be a part of a normal routine.
Emergency stop shall be tested after installation and after any service. Periodic

tests of emergency stop and other safety functions shall be scheduled according to regional and national recommendations.
Emergency stop shall be shared between all machines at a work station such that a push on a random emergency stop button stops all machines. Extra emergency stop buttons shall be placed at the work station according to the risk assessment.
All emergency stop buttons shall be marked with the text "Emergency Stop" or "E-STOP". All individuals in the facility shall be instructed on how to activate emergency stop.
Emergency stop and other safety equipment shall be connected redundantly. All relays shall be monitored and the system shall enter a safe condition if a relay fails. The circuitry shall confirm to specifications in section 3.2.

DANGER:
1. Make sure to install a sufficient number of emergency stop buttons and to place them carefully. Failure to locate an emegency stop button in the event of an emergency could result in serious injury or death.

1.6 Emergency movement of the robot arm

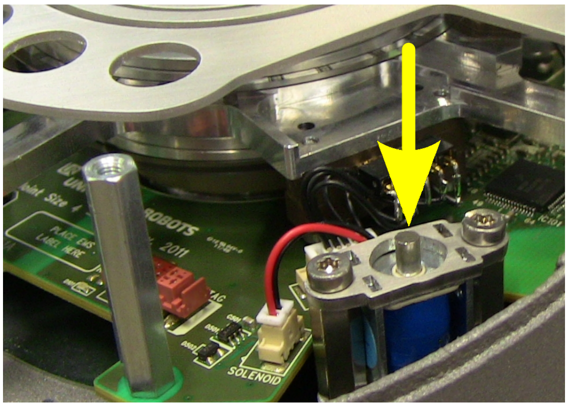
In the unlikely event of an emergency situation where one or more robot joints needs to be moved and robot power is either not possible or unwanted, there are three different ways to force movements of the robot joints without powering the motors of the joints:

1. Active backdriving: If possible, power on the robot by pushing the "ON" button on the initializing screen. Instead of pushing the "brake release" button to power up the joint motors, push the teach button on the backside of the teach pendant. A special backdrive mode is entered and the robot will loosen its brakes automatically while the robot is hand guided. Releasing the teach button re-locks the brakes.

2. Forced backdriving: Force a joint to move by pulling hard in the robot arm. Each joint break has a friction clutch which enables movement during high forced torque. Forced backdriving is intended for urgent emergencies only and might damage the joint gears and other parts.

3. Manual brake release: Remove the joint cover by removing the few M3 screws that fix it. Release the break by pushing the plunger on the small electromagnet as shown in the picture below.

WARNING:
1. Beware of gravity and heavy payloads. The robot can collapse with full weight. Do not turn any joints more than necessary.

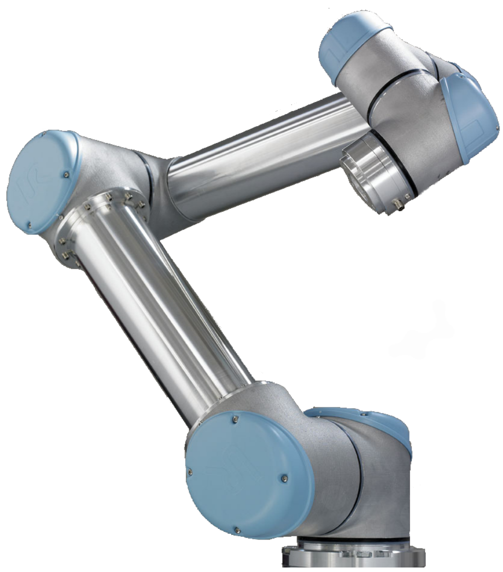

SOLENOID

Chapter 2

Getting started

2.1 Introduction

Congratulations on the purchase of your new Universal Robot, UR5.

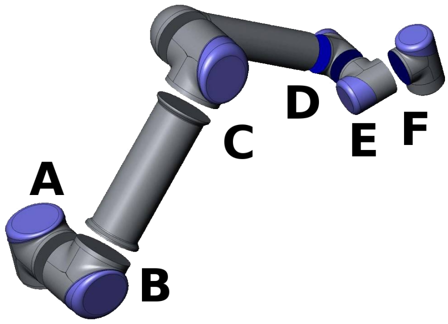


The robot is a machine that can be programmed to move a tool, and communicate with other machines using electrical signals. Using our patented programming interface, PolyScope, it is easy to program the robot to move the tool along a desired trajectory. PolyScope is described in section 4.1.

The reader of this manual is expected to be technically minded, to be familiar with the basic general concepts of programming, be able to connect a wire to a screw terminal, and be able to drill holes in a metal plate. No special knowledge about robots in general or Universal Robots in particular is required.

The rest of this chapter is an appetizer for getting started with the robot.

2.1.1 The Robot



The robot itself is an arm composed of extruded aluminum tubes and joints. The joints are named A:*Base*, B:*Shoulder*, C:*Elbow* and D,E,F:*Wrist 1,2,3*. 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 robot, and of course limited by the reach of the robot ($850mm$ from the center of the base).

2.1.2 Programs

A program is a list of commands telling the robot what to do. The user interface *PolyScope*, described later in this manual, 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 such an important part of a robot program, a way of teaching the robot how to move is essential. In PolyScope, the motions of the tool are given using a series of *waypoints*. Each waypoint is a point in the robot's workspace.

Waypoints

A waypoint is a point in the workspace of the robot. A waypoint can be given by moving the robot to a certain position, or can be calculated by software. The robot performs a task by moving through a sequence of waypoints. Various options regarding how the robot moves between the waypoints can be given in the program.

Defining Waypoints, Moving the Robot. The easiest way to define a waypoint is to move the robot to the desired position. This can be done in two ways: 1) By simply pulling the robot, while pressing the 'Teach' button on the screen (see 4.3.1). 2) By using the touch screen to drive the tool linearly or to drive each joint individually.

Blends. Per default the robot stops at each waypoint. By giving the robot freedom to decide how to move near the waypoint, it is possible to drive through the desired path faster without stopping. This freedom is given by setting a *blend radius* for the waypoint, which means that once the robot comes within a certain distance of the waypoint, the robot can decide to deviate from the path. A blend radius of 5-10 cm usually gives good results.

Features

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.

2.1.3 Risk assessment

A complete robot installation (Robot system / robot application / robot integration) is a machine. Since the robot cannot operate without being installed and without having a specific tool, it is not considered as a complete machine. A risk assessment shall be performed during each installation of the robot, see chapter 1.1.

2.2 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 section 2.5.

2.3 Turning On and Off

How to turn the different parts of the robot system on and off is described in the following subsections.

2.3.1 Turning on the Controller Box

The controller box is turned on by pressing the power button, at the front side of the teach pendant. When the controller box is turned on, a lot of text will appear on the screen. After about 20 seconds, the Universal Robot's Logo will appear, with the text 'Loading'. After around 40 seconds, a few buttons appear on the screen and a popup will force the user to go to the initialization screen.

2.3.2 Turning on the Robot

The robot can be turned on if the controller box is turned on, and if all emergency stop buttons are not activated. Turning the robot on is done at the initialization screen, by touching the 'ON' button at the screen, and then pressing

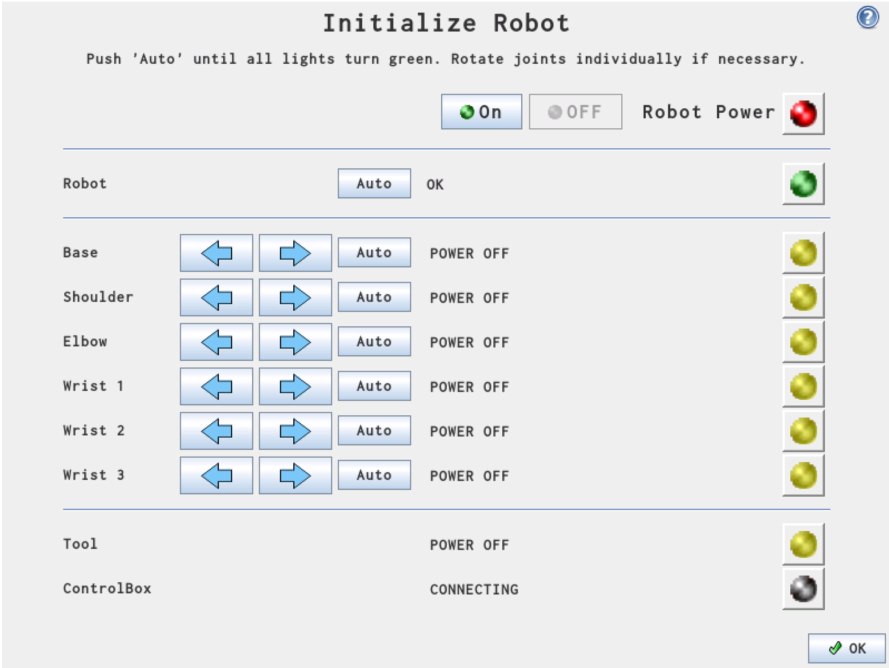


Figure 2.1: The initialization screen

'Start'. When a robot is started, a noise can be heard as the brakes unlock. After the robot has powereded up, it needs to be initialized before it can begin to perform work.

2.3.3 Initializing the Robot

After the robot is powered up, each of the robot's joints needs to find its exact position, by moving to a home position. Each large joint has around 20 home positions, evenly distributed over one joint revolution. The small joints have around 10. The Initialization screen, shown in figure 2.1, gives access to manual and semi-automatic driving of the robot's joints, to move them to a home position. The robot cannot automatically avoid collision with itself or the surrounds during this process. Therefore, caution should be exercised.

The *Auto* button near the top of the screen drives all joints until they are ready. When released and pressed again, all joints change drive direction. The *Manual* buttons permit manual driving of each joint.

A more detailed description of the initialization screen is found in section 4.1.2.

2.3.4 Shutting Down the Robot

The power to the robot can be turned off by touching the 'OFF' button at the initialization screen. Most users do not need to use this feature since the robot is automatically turned off when the controller box is shutting down.

2.3.5 Shutting Down the Controller Box

Shut down the system by pressing the green power button on the screen, or by using the 'Shut Down' button on the welcome screen.

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

2.4 Quick start, Step by Step

To quickly set up the robot, perform the following steps:

1. Unpack the robot and the controller box.
2. Mount the robot on a sturdy surface strong enough to withstand at least 10 times the full torque of the base joint and at least 5 times the weight of the robot arm. The surface shall be vibration free.
3. Place the controller box on its foot.
4. Plug on the robot cable between the robot and the controller box.
5. Plug in the mains plug of the controller box.
6. Press the Emergency Stop button on the front side of the teach pendant.
7. Press the power button on the teach pendant.
8. Wait a minute while the system is starting up, displaying text on the touch screen.
9. When the system is ready, a popup will be shown on the touch screen, stating that the emergency stop button is pressed.
10. Touch the `To Initialization Screen` button at the popup.
11. Unlock the emergency stop buttons. The robot state then changes from 'Emergency Stopped' to 'Robot Power Off'.
12. Step outside the reach (workspace) of the robot.
13. Touch the `On` button on the touch screen. Wait a few seconds.
14. Touch the `Start` button on the touch screen. The robot now makes a noise and moves a little while unlocking the breaks.
15. Touch the blue arrows and move the joints around until every "light" at the right side of the screen turns green. Be careful not to drive the robot into itself or anything else.
16. All joints are now `OK`. Touch the exit button, bringing you the `Welcome` screen.
17. Touch the `PROGRAM Robot` button and select `Empty Program`.
18. Touch the `Next` button (bottom right) so that the `<empty>` line is selected in the tree structure on the left side of the screen.
19. Go to the `Structure` tab.
20. Touch the `Move` button.
21. Go to the `Command` tab.
22. Press the `Next` button, to go to the `Waypoint` settings.
23. Press the `Set this waypoint` button next to the `"?"` picture.
24. On the `Move` screen, move the robot by pressing the various blue arrows, or move the robot by holding the `Teach` button while pulling the robot arm.

25. Press `OK`.
26. Press `Add waypoint before`.
27. Press the `Set this waypoint` button next to the `"?"` picture.
28. On the `Move` screen, move the robot by pressing the various blue arrows, or move the robot by holding the `Teach` button while pulling the robot arm.
29. Press `OK`.
30. 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'.
31. Congratulations! You have now produced your first robot program that moves the robot between the two given positions.

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

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

3. Only stretch your arm inside the reach (workspace) of the robot. Do not place fingers where they can be caught.

4. 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.

2.5 Mounting Instructions

The robot consists essentially of six robot joints and two aluminum tubes, connecting the robot's *base* with the robot's *tool*. The robot is built so that the tool can be translated and rotated within the robot's workspace. The next subsections describes the basic things to know when mounting the different parts of the robot system.

2.5.1 The Workspace of the Robot

The workspace of the UR5 robot extends to 850 mm from the base joint. The workspace of the robot is shown in figure 2.2. 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 robot 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.

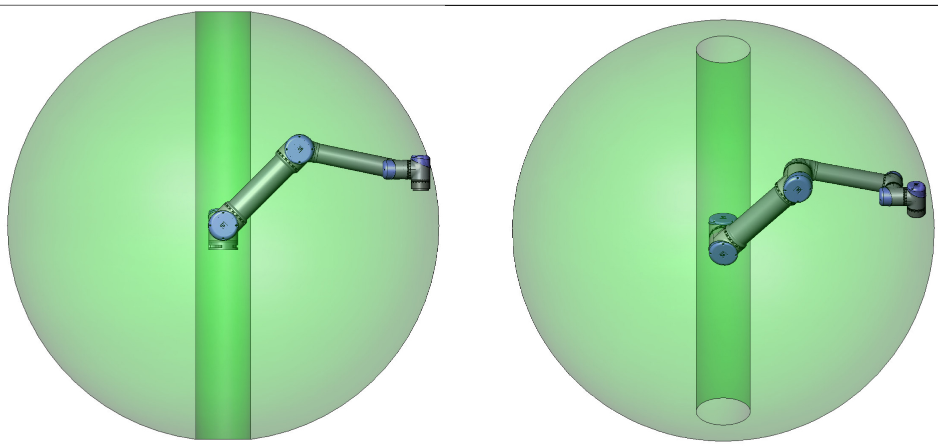


Figure 2.2: The workspace of the robot. The robot can work in an approximate sphere (Ø170cm) around the base, except for a cylindrical volume directly above and directly below the robot base.

2.5.2 Mounting the Robot

The robot is mounted using 4 M8 bolts, using the four $8.5mm$ holes on the robot's base. If very accurate repositioning of the robot is desired, two Ø8 holes are provided for use with a pin. Also an accurate base counterpart can be purchased as accessory. Figure 2.3 shows where to drill holes and mount the screws.
Mount the robot on a sturdy surface strong enough to withstand at least 10 times the full torque of the base joint and at least 5 times the waight of the robot arm. Furthermore the surface shall be vibration free.
If the robot is mounted on a linary 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:
1. Make sure the robot arm is properly and securely bolted in place. The mounting surface shall be sturdy.

CAUTION:
1. 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.

2.5.3 Mounting the Tool

The robot tool flange has four holes for attaching a tool to the robot. A drawing of the tool flange is shown in figure 2.4.

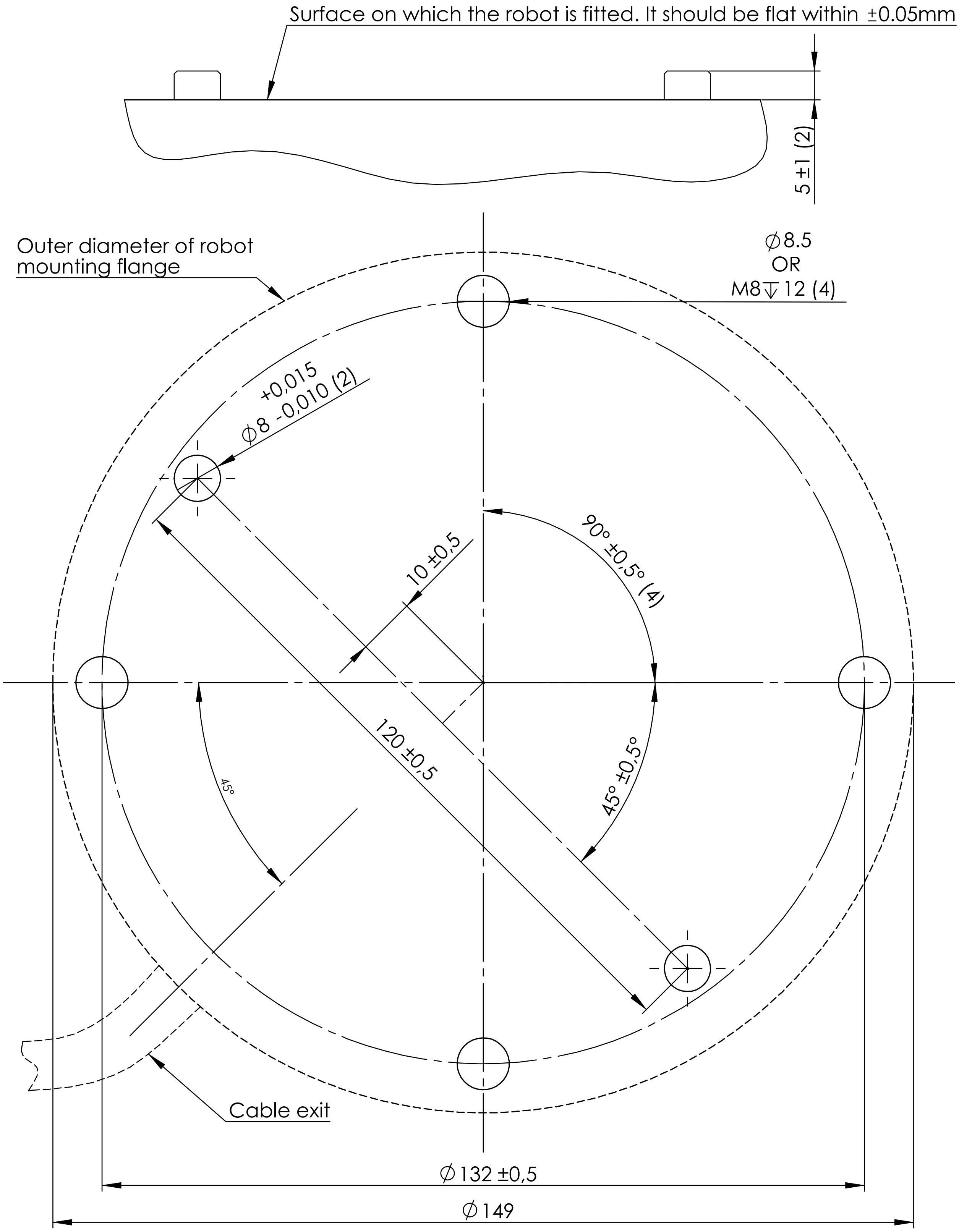


Figure 2.3: Holes for mounting the robot, scale 1:1. Use 4 M8 bolts. All measurements are in mm.

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.

2.5.4 Mounting the Controller Box

The controller box can be hung on a wall, or it can be placed on the ground. A clearance of 50mm on each side allows for sufficient airflow.

DANGER:
1. Make sure that the controller box and cables do not come into contact with liquids. A wet controller box could cause death.

2.5.5 Mounting the Screen

The screen can be hung on a wall or on the controller box. Extra fittings can be bought. Make sure that no one can trip over the cable.

2.5.6 Connecting Cables

Use only the original robot cable to connect the robot to the controller box. Ensure that the connectors are properly secured in place. All electrical connections to the tool or the interface inside the controller box shall conform to the specifications in section 3.1. Never connect or disconnect any cables when the power is enabled or if any connector is wet. All electrical equipment in the robot installation shall be connected to earth. Use the screw connection marked with earth symbol inside the controller box when potential equalization with other machines is required.
The mains supply shall be equiped with the following as a minimum:

1. A correct sized fuse.
2. A residual current device (RCD).
3. A correct connection to earth.

Mains input specifications are shown below. The shown power specifications are averages taken over one minute under normal conditions. Power consumption peaks during high accelerations at high speeds with high payloads.

Parameter	Min	Typ	Max	Unit
Input voltage	100	-	240	VAC
External fuse	8	-	16*	A
Input frequency	47	-	63	Hz
Stand-by power	-	-	0.5	W
Nominal operating power	90	150	325	W

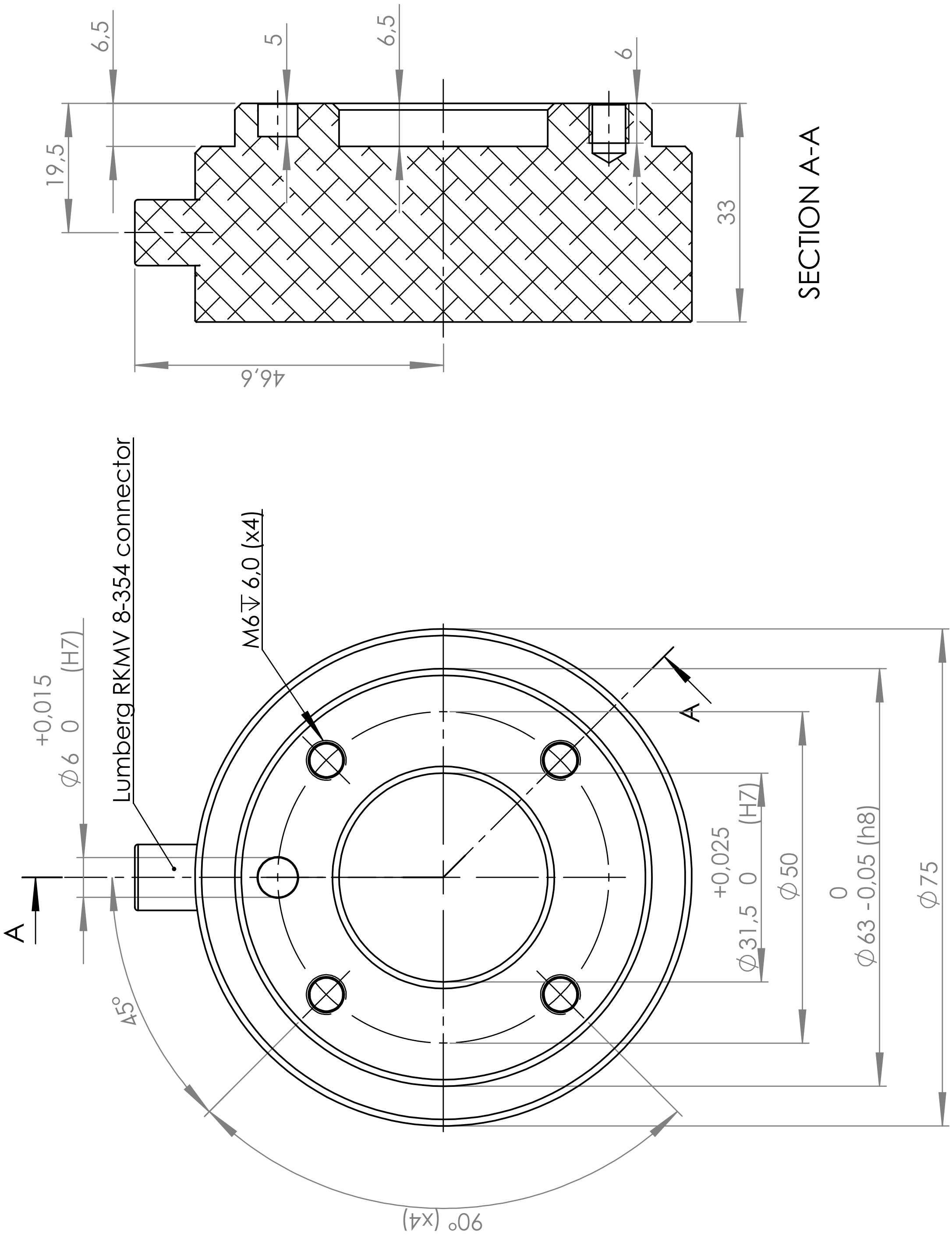


Figure 2.4: The tool output flange, ISO 9409-1-50-4-M6. This is where the tool is mounted at the tip of the robot. All measures are in mm.

*NOTE: The fuse shall not be higher than the lowest current rating of the specific country plug and socket.

DANGER:

1. 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.

2. 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.

3. Make sure that the input power to the controller box is protected with a RCD (Residual Current Device) and a correct fuse.

4. Make sure that all cables are connected correctly before the controller box is powered. Always use an original and correct power cord.

5. 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.

6. 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.

7. Never stick fingers behind the internal cover of the controller box. The cover protects against high voltages. High voltage can still be present inside the controller box, even though it is locked out.

8. 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 any holes. Make sure that all shavings are removed before reinstalling the plate. Remember to use correct gland sizes.

Chapter 3

Electrical Interface

3.1 Introduction

The robot is a machine that can be programmed to move a tool around in the robots workspace. Often, it is desired to coordinate robot motion with nearby machines or equipment on the tool. The most straightforward way to achieve this is often by using the electrical interface.

There are electrical input and output signals (I/Os) inside the controller box and at the robot tool flange. This chapter explains how to connect equipment to the I/Os. Some of the I/Os inside the controller box are dedicated to the robot safety functionality, and some are general purpose I/Os for connecting with other machines and equipment. The general purpose I/Os can be manipulated directly on the I/O tab in the user interface, see section 4.3.2, or by the robot programs.

In this chapter, all unspecified voltage and current data are in DC.

For additional I/O, Modbus units can be added via the extra Ethernet connector in the controller box.

DANGER:
1. Never connect a safety interface to a PLC which is not a safety PLC with the correct safety level. Failure to follow this warning could result in serious injury or death as the safety stop function could be overridden. It is important to keep safety interface signals seperated from the normal I/O interface signals.

2. Every minus connection (0V) is referred to as GND, and is connected to the shield of the robot and the controller box. All mentioned GND connections are only for powering and signaling. For PE (Protective Earth) use the M6 sized screw connections marked with earth symbols inside the controller box. The grounding conductor shall have at least the current rating of the highest current in the system. If FE (Functional Earth) is needed use one of the M3 screws close to the screw terminals.

3. Make sure that the mounting instructions are followed, see section 2.5.

4. 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.

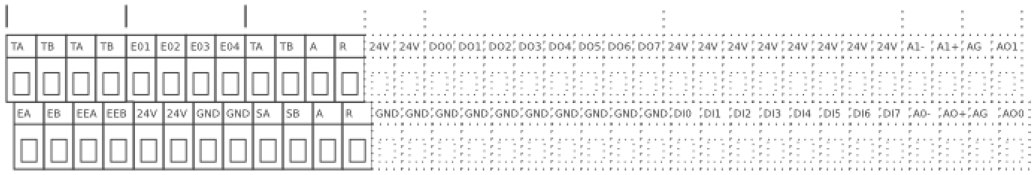
5. Make sure that all equipment not rated for water exposure remains dry. If water comes inside the product, lockout and tagout all power and then contact your supplier.

6. 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. According international IEC standards for EMC cables going from the controller box to other machinery and factory equipment may not be longer than 30m, unless extended tests are performed.

3.2 The Safety Interface



Inside the controller box there is a panel of screw terminals. The leftmost part, in black above, is the safety interface. The safety interface can be used to connect the robot to other machinery or protective equipment, to make sure the robots stops in certain situations.

The safety interface is comprised of two parts; the emergency stop interface and the safeguard stop interface, further described in the following sections. The table below summarizes their differences:

	Emergency Stop	Safeguard Stop
Robot stops moving	Yes	Yes
Initiations	Manual	Manual or automatic
Program execution	Stops	Pauses
Brakes	Active	Not active
Motor power	Off	Limited
Reset	Manual	Automatic or manual
Use frequency	Infrequent	Every cycle to infrequent
Requires re-initialization	Brake release only	No
EN/IEC 60204 and NFPA 79	Stop category 1	Stop category 2
Performance level	ISO 13849-1 PLd	ISO 13849-1 PLd

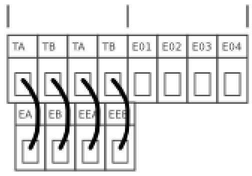
3.2.1 The Emergency Stop Interface

[TA]	Test Output A
[TB]	Test Output B
[EO1]	Emergency Stop Output Connection 1
[EO2]	Emergency Stop Output Connection 2
[EO3]	Emergency Stop Output Connection 3
[EO4]	Emergency Stop Output Connection 4
[EA]	Robot Emergency Stop Input A (Positive)
[EB]	Robot Emergency Stop Input B (Negative)
[EEA]	External Emergency Stop Input A (Positive)
[EEB]	External Emergency Stop B (Negative)
[24V]	+24V supply connection for safety devices
[GND]	0V supply connection for safety devices

The Emergency Stop interface has two inputs, the Robot Emergency Stop input and the External Emergency Stop input. Each input is doubled for redundancy due to the safety performance level **d**.

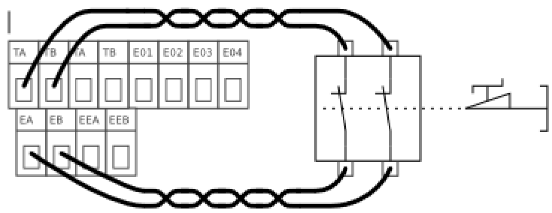
The Robot Emergency Stop interface will stop the robot, and will set the Emergency Stop output, intended for use by safety equipment near the robot. The External Emergency Stop will also stop the robot, but will not affect the Emergency Stop output, and is only intended for connecting to other machines.

The Simplest Emergency Stop Configuration



The simplest configuration is to use the internal emergency stop button as the only component to generate an emergency stop. This is done with the configuration shown above. This configuration is the default when the robot leaves the factory, and thereby the robot is ready to operate. However, the emergency configuration shall be changed if required by the risk assessment, see section 1.1.

Connecting an External Emergency Stop Button

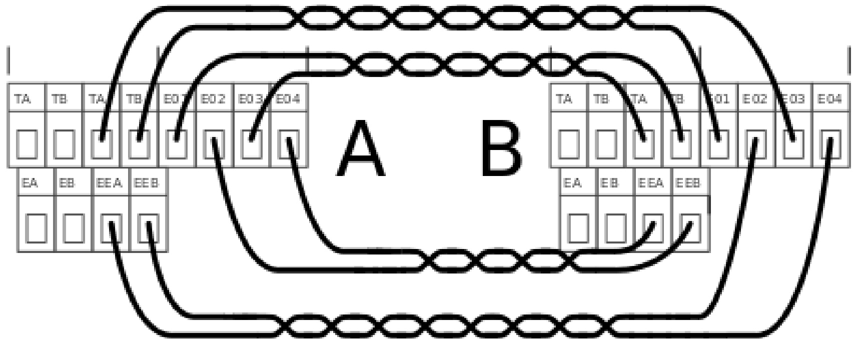


In almost every robot installation, the integrator is required, according to the risk assessment conducted, to connect one or more external emergency stop buttons. Doing so is simple and easy. An example of how to connect one extra button is shown above.

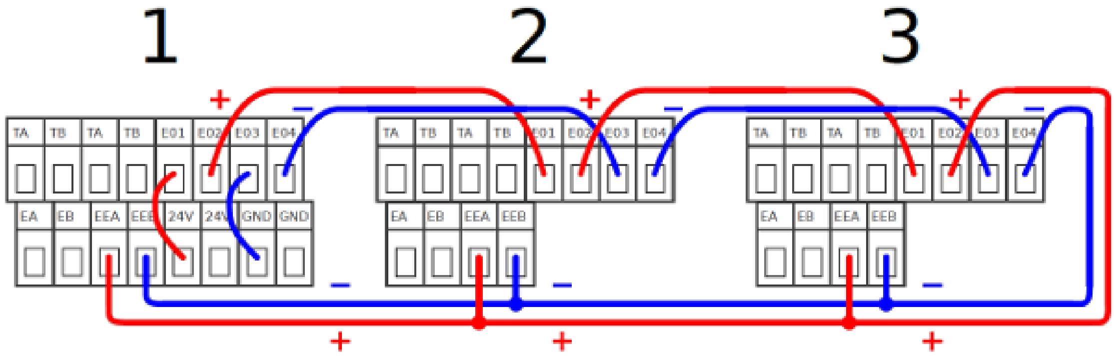
Connecting Emergency Stop to Other Machinery

When the robot is used together with other electro-mechanical machinery, it is often required to set up a common emergency stop circuit. This ensures that if a dangerous situation arises, the operator does not need to think about which buttons to use. It is also often preferable for every part of a sub-function in a product line to be synchronized, since a stop in only one part of the product line can lead to a dangerous situation.

An example with two UR robots emergency stopping each other is shown below.

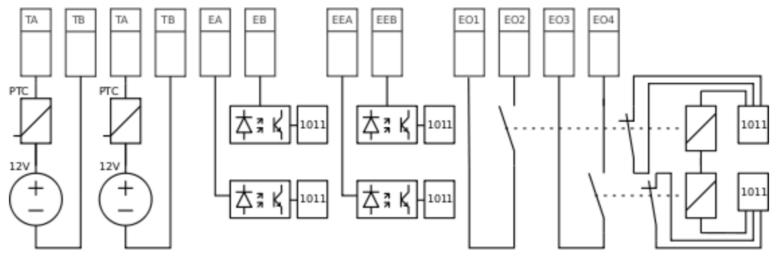

TA TB TA TB E01 E02 E03 E04
EA EB EEA EEB
A
B
TA TB TA TB E01 E02 E03 E04
EA EB EEA EEB

An example where multiple UR robots share their emergency stop function is shown below. Connect more robots as robot number 2 is connected.
This example uses 24V which works with many other machines. Make sure to comply with all electrical specifications when UR robots share emergency stop with other machinery.



Electric Specifications

A simplified internal schematic of circuitry is shown below. It is important to notice that any short circuit or lost connection will lead to a safe stop, as long as only one error appears at a time. Failure and abnormal behavior of relays and power supplies results in an error message in the robot log and prevents the robot from powering up.



Below: Specifications of the Emergency Stop Interface.

Parameter	Min	Typ	Max	Unit
[TA-TB] Voltage	10.5	12	12.5	V
[TA-TB] Current (Each output)	-	-	120	mA
[TA-TB] Current protection	-	400	-	mA
[EA-EB] [EEA-EEB] Input voltage	-30	-	30	V
[EA-EB] [EEA-EEB] Guaranteed OFF if	-30	-	7	V
[EA-EB] [EEA-EEB] Guaranteed ON if	10	-	30	V
[EA-EB] [EEA-EEB] Guaranteed OFF if	0	-	3	mA
[EA-EB] [EEA-EEB] ON Current (10-30V)	7	-	14	mA
[EO1-EO2] [EO3-EO4] Contact Current AC/DC	0.01	-	6	A
[EO1-EO2] [EO3-EO4] Contact Voltage DC	5	-	50	V
[EO1-EO2] [EO3-EO4] Contact Voltage AC	5	-	250	V

DANGER:
1. 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.

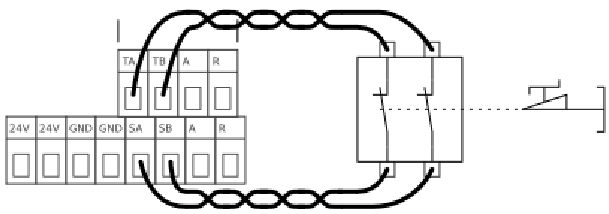
The two emergency stop inputs EA-EB and EEA-EEB are potential free inputs conforming to IEC 60664-1 and EN 60664-1, pollution degree 2, overvoltage category II.
The emergency stop outputs EO1-EO2-EO3-EO4 are relay contacts conforming to IEC 60664-1 and EN 60664-1, pollution degree 2, over-voltage category III.

3.2.2 The Safeguard Interface

[TA] [TB]	Test Output A Test Output B
[SA] [SB]	Safeguard Stop Input A (Positive) Safeguard Stop Input B (Negative)
[A] [R]	Automatic continue after safeguard stop Reset safeguard stop
[24V] [GND]	+24V supply connection for safety devices 0V supply connection for safety devices

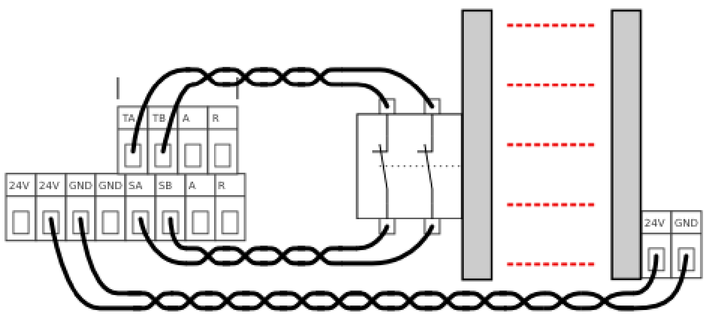
The Safeguard Interface is used to pause the robot movement in a safe way. The Safeguard Interface can be used for light guards, door switches, safety PLCs etc. Resuming from a safeguard stop can be automatic or can be controlled by a pushbutton, depending on the safeguard configuration. If the Safeguard Interface is not used then enable automatic reset functionality as described in section 3.2.3.

Connecting a door switch



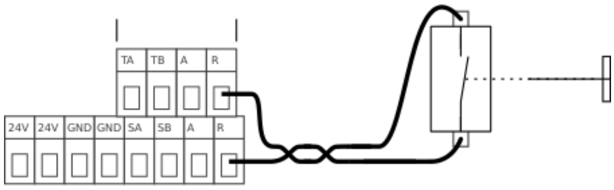
Connecting a door switch or something comparable is done as shown above. Remember to use a reset button configuration if the robot should not start automatically when the door is closed again.

Connecting a light guard



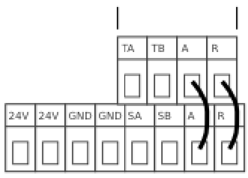
How to connect a light guard is shown above. It is also possible to use a category 1 (ISO 13849-1 and EN 954-1) light guard if the risk assessment allows it. When connecting a category 1 light guard use TA and SA and then connect TB and SB with a wire. Remember to use a reset button configuration so that the safeguard stop is latched.

Connecting a reset button



How to connect a reset button is shown above. It is not allowed to have a permanently pushed reset button. If the reset button is stuck a safeguard stop is generated and an error message will appear on the log screen.

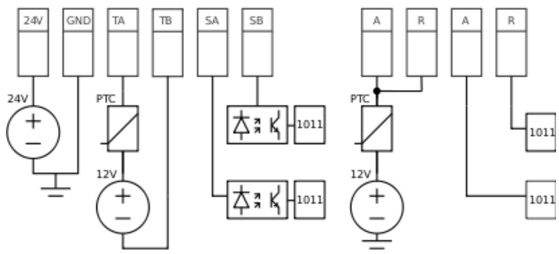
3.2.3 Automatic continue after safeguard stop



The safeguard interface can reset itself when a safeguard stop event is gone. How to enable automatic reset functionality is shown above. This is also the recommended configuration if the safeguard interface is not used. However, it is not recommended to use automatic reset if a reset button configuration is possible. Automatic reset is intended for special installations and installations with other machinery.

Electric Specifications

To understand the safeguard functionality, a simplified internal schematics of the circuitry is shown below. Any failure in the safety system will lead to a safe stop of the robot and an error message on the log screen.

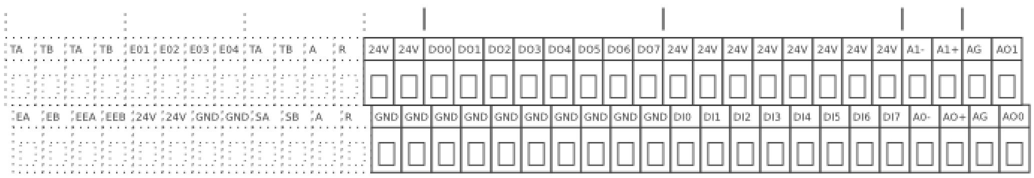


Parameter	Min	Typ	Max	Unit
24V Voltage tolerance	-15%	-	+20%	-
Current available from 24V supply	-	-	1.2*	A
Overload protection	-	1.4	-	A
[TA-TB] [A↑] [R↑] Voltage	10.5	12	12.5	V
[TA-TB] [A↑] [R↑] Current	-	-	120	mA
[TA-TB] [A↑] [R↑] Current protection	-	400	-	mA
[SA-SB] Input voltage	-30	-	30	V
[SA-SB] Guaranteed OFF if	-30	-	7	V
[SA-SB] Guaranteed ON if	10	-	30	V
[SA-SB] Guaranteed OFF if	0	-	3	mA
[SA-SB] ON Current (10-30V)	7	-	14	mA
[A↓] [R↓] Input voltage	-30	-	30	V
[A↓] [R↓] Input guaranteed OFF if	-30	-	7	V
[A↓] [R↓] Input guaranteed ON if	10	-	30	V
[A↓] [R↓] Guaranteed OFF if	0	-	5	mA
[A↓] [R↓] ON Current (10-30V)	6	-	10	mA

The safeguard stop input SA-SB is a potential free input conforming to IEC 60664-1 and EN 60664-1, pollution degree 2, over-voltage category II.

Note that the yellow 24V connections is sourced by the same internal 24V power supply as the 24V connections of the normal I/O, and that the maximum of 1.2 A is for both power sources together.

3.3 Controller I/O



Inside the controller box there is a panel of screw terminals with various I/O parts, as shown above. The rightmost part of this panel is general purpose I/O.

[24V]	+24V supply connection
[GND]	0V supply connection
[DOx]	Digital output number x
[DIx]	Digital input number x
[AOx]	Analog output number x plus
[AG]	Analog output GND
[Ax+]	Analog input number x plus
[Ax-]	Analog input number x minus

The I/O panel in the controller box has 8 digital and 2 analog inputs, 8 digital and 2 analog outputs, and a built in 24V power supply. Digital inputs and outputs are **pnp** technology and constructed in compliance with IEC 61131-2 and EN 61131-2. 24V and GND can be used as input for the I/O module or output as a 24V power supply. When the controller box is booting it checks if voltage is applied to the 24V connection from an external power supply, and if not, it automatically connects the internal 24V power supply.

Electrical specifications of the internal power supply

Parameter	Min	Typ	Max	Unit
Internal 24V voltage tolerance	-15%	-	+20%	-
Current from internal 24V supply	-	-	1.2∗	A
Overload protection	-	1.4	-	A
External power supply voltage	10	-	30	V

Note that the safeguard (yellow) 24V connections are sourced by the same internal 24V power supply as the 24V connections of the normal I/O, and that the maximum of 1.2 A is for both power sources together.

If the current load of the internal 24V power supply is exceeded, an error message is printed on the log screen. The power supply will automatically try to recover after a few seconds.

3.3.1 Digital Outputs

Parameter	Min	Typ	Max	Unit
Source current per output	0	-	2	A
Source current all outputs together	0	-	4	A
Voltage drop when ON	0	-	0.2	V
Leakage current when OFF 0	0	-	0.1	mA

The outputs can be used to drive equipment directly e.g. pneumatic relays or they can be used for communication with other PLC systems. The outputs are constructed in compliance with all three types of digital inputs defined in IEC 61131-2 and EN 61131-2, and with all requirements for digital outputs of the same standards.

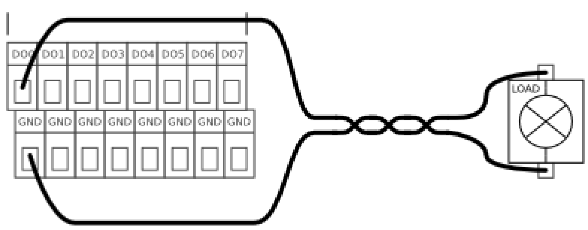
All digital outputs can be disabled automatically when a program is stopped, by using the check box "Always low at program stop" on the I/O Name screen (see section 4.3.8). In this mode, the output is always low when a program is not running.

CAUTION:
1. The digital outputs are not current limited and overriding the specified data can cause permanent damage to them. However, it is not possible to damage the outputs if the internal 24V power supply is used due to its current protection.

2. The controller box and the metal shields are connected to GND. Never send I/O current through the shields or earth connections; it might damage to the I/Os.

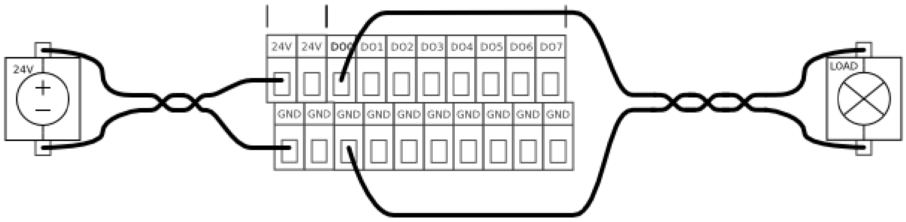
The next subsections show some simple examples of how the digital outputs could be used.

Load Controlled by Digital Output



This example illustrates how to turn on a load.

Load Controlled by Digital Output, External Power



If the available current from the internal power supply is not enough, simply use an external power supply, as shown above.

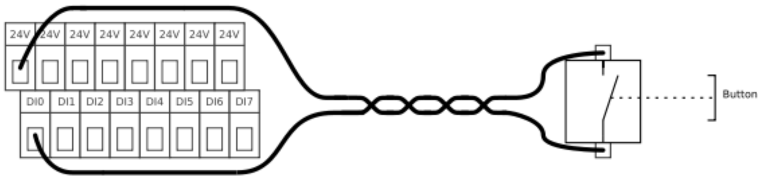
3.3.2 Digital Inputs

Parameter	Min	Typ	Max	Unit
Input voltage	-30	-	30	V
Input guaranteed OFF if	-30	-	7	V
Input guaranteed ON if	10	-	30	V
Guaranteed OFF if	0	-	5	mA
ON Current (10-30V)	6	-	10	mA

The digital inputs are implemented as **pnp** which means that they are active when voltage is applied to them. The inputs can be used to read buttons, sensors or for communication with other PLC systems. The inputs are compliant with all three types of digital inputs defined in IEC 61131-2 and EN 61131-2, which means that they will work together with all types of digital outputs defined in the same standards.

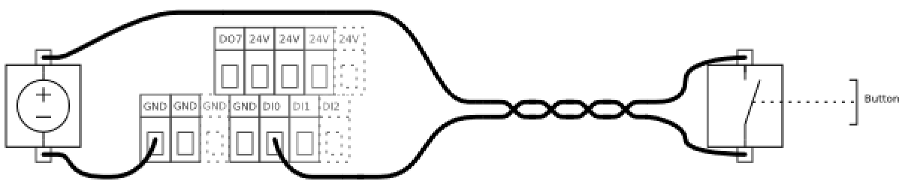
Technical specifications of the digital inputs are shown below.

Digital Input, Simple Button



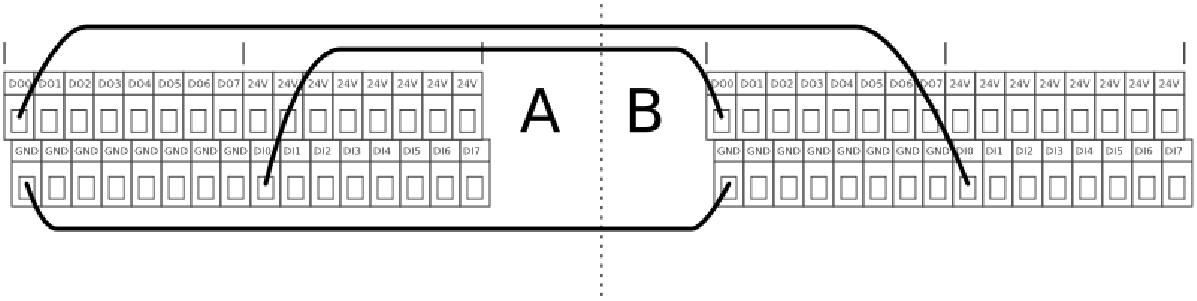
The above example shows how to connect a simple button or switch.

Digital Input, Simple Button, External Power



The above illustration shows how to connect a button using an external power source.

Signal Communication with other Machinery or PLCs



If communication with other machinery or PLCs is needed they must use **pnp** technology. Remember to create a common GND connection between the different interfaces. An example where two UR robots (A and B) are communicating with each other is illustrated above.

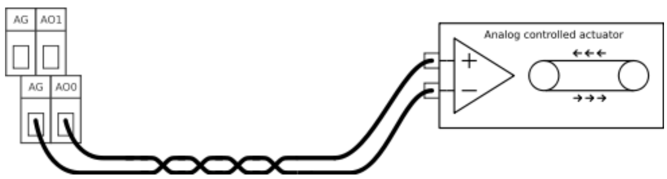
3.3.3 Analog Outputs

Parameter	Min	Typ	Max	Unit
Valid output voltage in current mode	0	-	10	V
Valid output current in voltage mode	-20	-	20	mA
Short-circuit current in voltage mode	-	40	-	mA
Output resistance in voltage mode	-	43	-	ohm

The analog outputs can be set for both current mode and voltage mode, in the range of 4-20mA and 0-10V respectively.

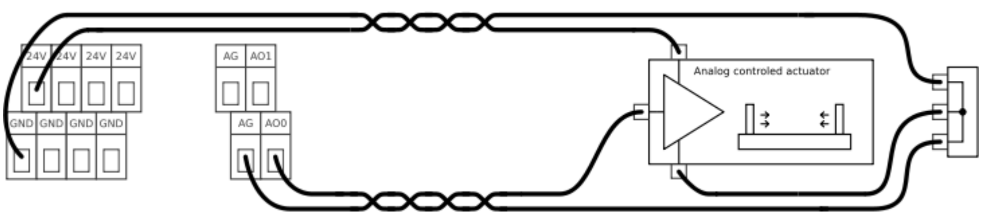
To illustrate clearly how easy it is to use analog outputs, some simple examples are shown.

Using the Analog Outputs



This is the normal and best way to use analog outputs. The illustration shows a setup where the robot controller controls an actuator like a conveyor belt. The best result is accomplished when using current mode, because it is more immune to disturbing signals.

Using the Analog Outputs, Non-Differential Signal



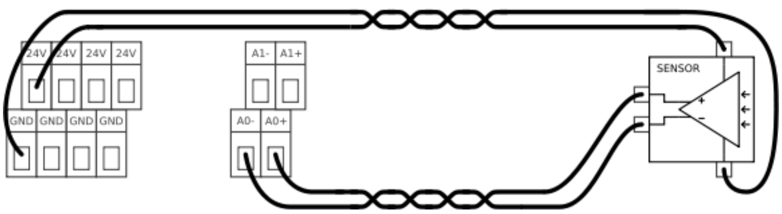
If the controlled equipment does not take a differential input, an alternative solution can be made as shown above. This solution is not very good in terms of noise, and can easily pick up disturbing signals from other machinery. Care must be taken when the wiring is done, and it must be kept in mind that disturbing signals induced into analog outputs may also be present on other analog I/O.

3.3.4 Analog Inputs

Parameter	Min	Typ	Max	Unit
Common mode input voltage	-33	-	33	V
Differential mode input voltage*	-33	-	33	V
Differential input resistance	-	220	-	kohm
Common mode input resistance	-	55	-	kohm
Common mode rejection ratio	75	-	-	dB

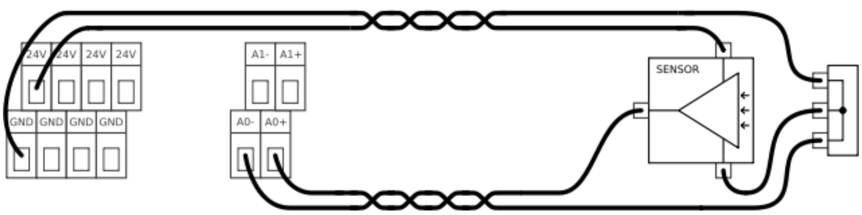
The analog inputs can be set to four different voltage ranges, which are implemented in different ways, and therefore can have different offset and gain errors. The specified differential mode input voltage is only valid with a common mode voltage of 0V. To make it clear how easy it is to use analog outputs, some simple examples are shown.

Using Analog Inputs, Differential Voltage Input



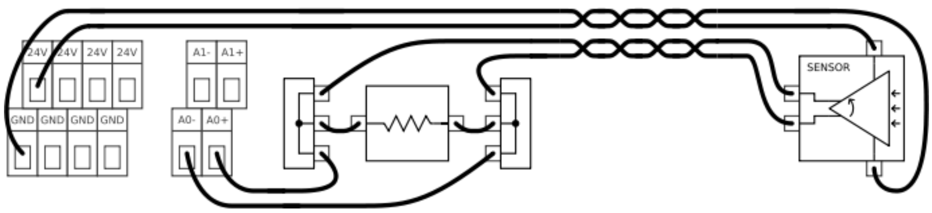
The simplest way to use analog inputs. The equipment shown, which could be a sensor, has a differential voltage output.

Using Analog Inputs, Non-differential Voltage Input



If it is not possible to achieve a differential signal from the equipment used, a solution could look something like the setup above. Unlike the non-differential analog output example in subsection 3.3.3, this solution would be almost as good as the differential solutions.

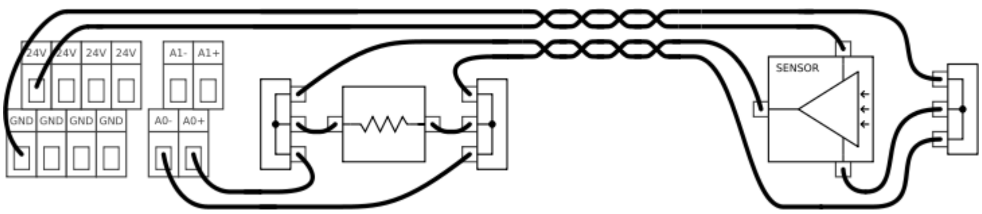
Using Analog Inputs, Differential Current Input



When longer cables are used, or if it is a very noisy environment, current based signals are preferred. Also, some equipment comes only with a current output. To use current as inputs, an external resistor is needed as shown above. The value of the resistor would normally be around 200 ohms, and the best result is accomplished when the resistor is close to the screw terminals of the controller box.

Note that the tolerance of the resistor and the ohmic change due to temperature must be added to the error specifications of the analog inputs.

Using Analog Inputs, Non-differential Current Input



If the output of the equipment is a non-differential current signal, a resistor must be used as shown above. The resistor should be around 200 ohms and the relationship between the voltage at the controller input and the output of the sensor is given by:

$$\text{Voltage} = \text{Current} \times \text{Resistance}$$

Note that the tolerance of the resistor and the ohmic change due to temperature must be added to the error specifications of the analog inputs.

3.4 Tool I/O

At the tool end of the robot there is a small connector with eight connections.

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)

This connector provides power and control signals for basic grippers and sensors, which may be present on a specific robot tool. This connector can be used to reduce wiring between the tool and the controller box. The connector is a standard Lumberg RSMEDG8, which mates with a cable named RKMV 8-354.

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

Internal Power Supply Specifications

Parameter	Min	Typ	Max	Unit
Supply voltage in 24V mode	TBD	24	TBD	V
Supply voltage in 12V mode	TBD	12	TBD	V
Supply current in both modes	-	-	600	mA
Short-circuit current protection	-	650	-	mA
Capacitive load	-	-	TBD	uF
Inductive load	-	-	TBD	uH

The available power supply can be set to either 0V, 12V or 24V at the I/O tab in the graphical user interface (see section 4.3.2).

WARNING:
1. Take care when using 12V. An error made by the programmer can cause a voltage change to 24V, which may damage the equipment and start a fire.

The internal control system will generate an error to the robot log if the current exceeds its limit. The different I/Os at the tool is described in the following three subsections.

3.4.1 Digital Outputs

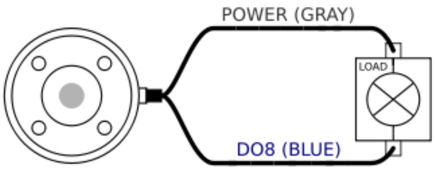
Parameter	Min	Typ	Max	Unit
Voltage when open	-0.5	-	26	V
Voltage when sinking 1A	-	0.05	0.20	V
Current when sinking	0	-	1	A
Current through GND	-	-	1	A
Switch time	-	1	-	ms

The digital outputs are implemented so that they can only sink to GND (0V) and not source current. 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 primary difference between the digital outputs inside the controller box and those in the tool is the reduced current due to the small connector.

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

To illustrate clearly how easy it is to use digital outputs, a simple example is shown.

Using Digital Outputs



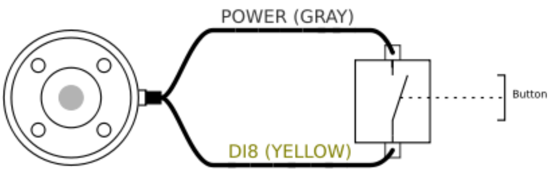
This example illustrates how to turn on a load, when using the internal 12V or 24V power supply. Remember that you have to define the output voltage at the I/O tab (see section 4.3.2). Keep in mind that there is voltage between the POWER connection and the shield/ground, even when the load is turned off.

3.4.2 Digital Inputs

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

The digital inputs are implemented with weak pull-down resistors. This means that a floating input will always read low. The digital inputs at the tool are implemented in the same way as the digital inputs inside the controller box.

Using Digital Inputs



The above example shows how to connect a simple button or switch.

3.4.3 Analog Inputs

The analog inputs at the tool are very different from those inside the controller box. The first thing to notice is that they are non-differential, which is a drawback compared to the analog inputs at the controller I/O. The second thing to notice is that the tool analog inputs have current mode functionality, which is an advantage compared with the controller I/O. The analog inputs can be set to different input ranges, which are implemented in different ways, and therefore can have different offset and gain errors.

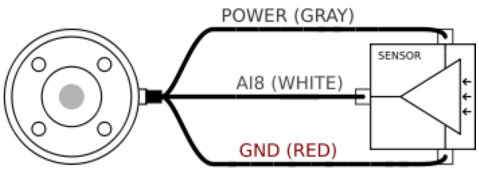
Parameter	Min	Typ	Max	Unit
Input voltage in voltage mode	-0.5	-	26	V
Input voltage in current mode	-0.5	-	5.0	V
Input current in current mode	-2.5	-	25	mA
Input resistance @ range 0V to 5V	-	29	-	kΩ
Input resistance @ range 0V to 10V	-	15	-	kΩ
Input resistance @ range 4mA to 20mA	-	200	-	Ω

An important thing to realize is that any current change in the common GND connection can result in a disturbing signal in the analog inputs, because there will be a voltage drop along the GND wires and inside connectors.

Note that a connection between the tool power supply and the analog inputs will permanently damage the I/O functionality, if the analog inputs are set in current mode.

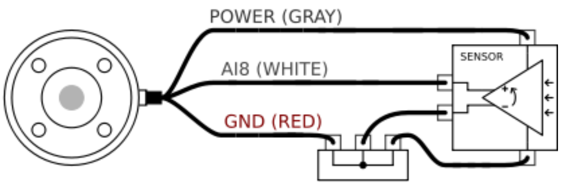
To make it clear how easy it is to use digital inputs, some simple examples are shown.

Using Analog Inputs, Non-differential



The simplest way to use analog inputs. 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 (see section 4.3.2). Remember to check that a sensor with voltage output can drive the internal resistance of the tool, or the measurement might be invalid.

Using Analog Inputs, Differential



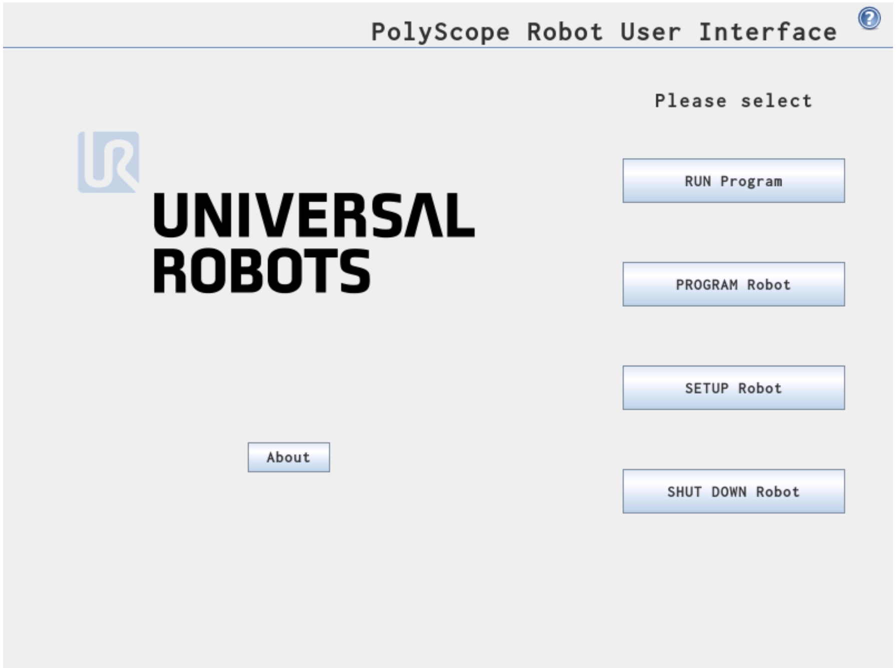
Using sensors with differential outputs is also straightforward. Simply connect the negative output part to GND (0V) with a terminal strip and it will work in the same way as a non-differential sensor.

Chapter 4

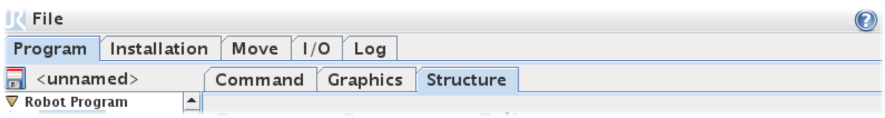
PolyScope Software

4.1 Introduction

PolyScope is the graphical user interface (GUI) which lets you operate the robot, run existing robot programs or easily create new ones. PolyScope runs on the touch sensitive screen attached to the control box. To calibrate the touch screen, read section 4.5.6.

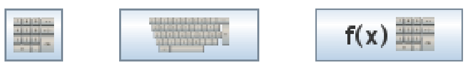


The picture above shows the Welcome Screen. The bluish areas of the screen are buttons that can be pressed by pressing a finger or the backside of a pen against the screen. PolyScope has a hierarchical structure of screens. In the programming environment, the screens are arranged in *tabs*, for easy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



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 can be moved. Similarly, by selecting the `I/O` tab, the current state of the electrical I/O can be monitored and changed.

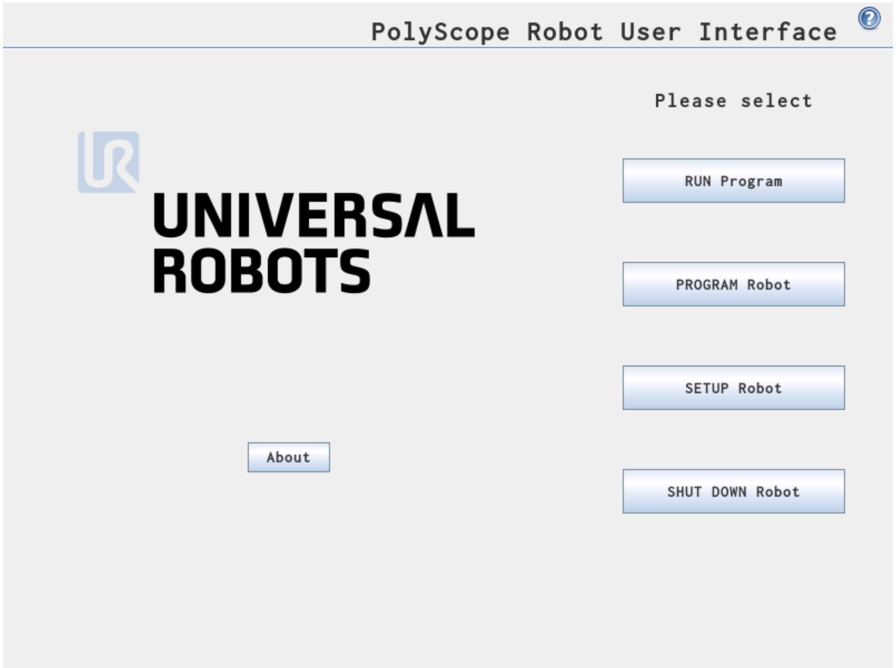
It is possible to connect a mouse and a keyboard to the controller box; however, this is not required. Whenever a text or number input is needed, an on-screen keypad or keyboard is provided.



The on-screen keypad, keyboard and expression editor can be reached using the buttons shown above.

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

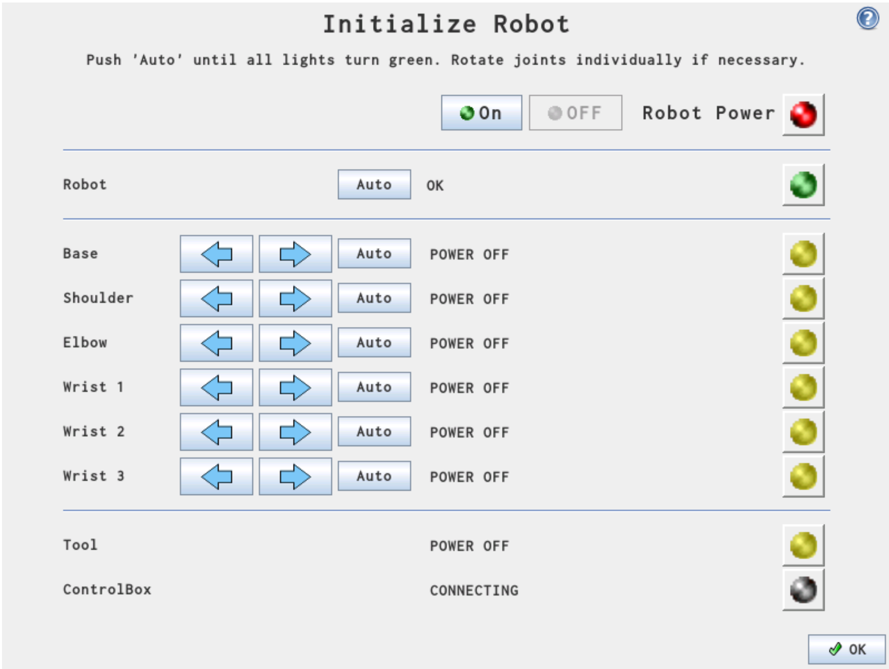
4.1.1 Welcome Screen



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

- **Run Program:** Choose a program to run. This is the simplest way to operate the robot, but requires a suitable program to have already been produced.
- **Program Robot:** Change a program, or create a new program.
- **Setup:** Set passwords, upgrade software via the Internet, request support, calibrate the touch screen, etc.
- **Shut Down Robot:** Shuts down the Controller PC and powers off the robot.

4.1.2 Initialization Screen



On this screen you control the initialization of the robot. When turned on, the robot needs to find the positions of each joint. To get the joint positions, the robot needs to move each joint.

WARNING:
1. The force protection during initialization can be higher than 150N. The maximum force is only reduced by torque limits in the joints. Stay outside the robot workspace when the robot arm is moving.

Status LEDs

The status LEDs give an indication of the joints' running state.

- A bright red LED tells that the robot is currently in a stopped state where the reasons can be several.
- A bright yellow LED indicates that the joint is running, but dosn't know its presents position and needs homing.
- Finally a green LED indicates that the joint is running correctly and is ready to execute.

All the LEDs have to be green in order for the robot to operate normally.

Manual motion (By hand)

When the joints are `Ready` and the "Teach" button on the back of the screen is pressed, the joint modes change to `Backdrive`. In this mode, the joints will release the brakes when motion is detected. This way, the robot can be moved out of a machine manually, before being started up. The brakes will reactive as soon as the button is released again.

Auto movement (Auto Buttons)

Normally it is always advisable to use the auto buttons to move the individual joints until they reach a known state. In order to operate the button, you have to press on the Auto button, and keep it pressed.
The auto buttons can be pressed individually for each joint, or for the whole robot.

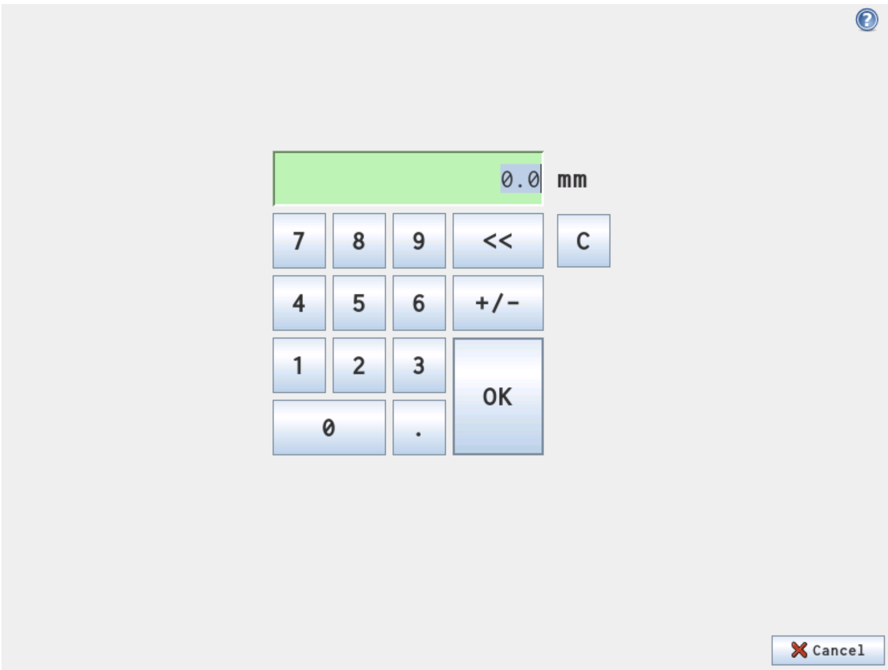
CAUTION:
1. Great care should be taken if the robot is touching an obstacle or table, since driving the robot into the obstacle might damage a joint gearbox.

Moving directly (Move Buttons)

In the case where a joint is in a position where there is a major risk that uncontrolled motion would cause damage to the robot or its surroundings, the operator can choose to home the robot manually for each joint. section 4.1.2.

4.2 On-screen Editors

4.2.1 On-screen Keypad

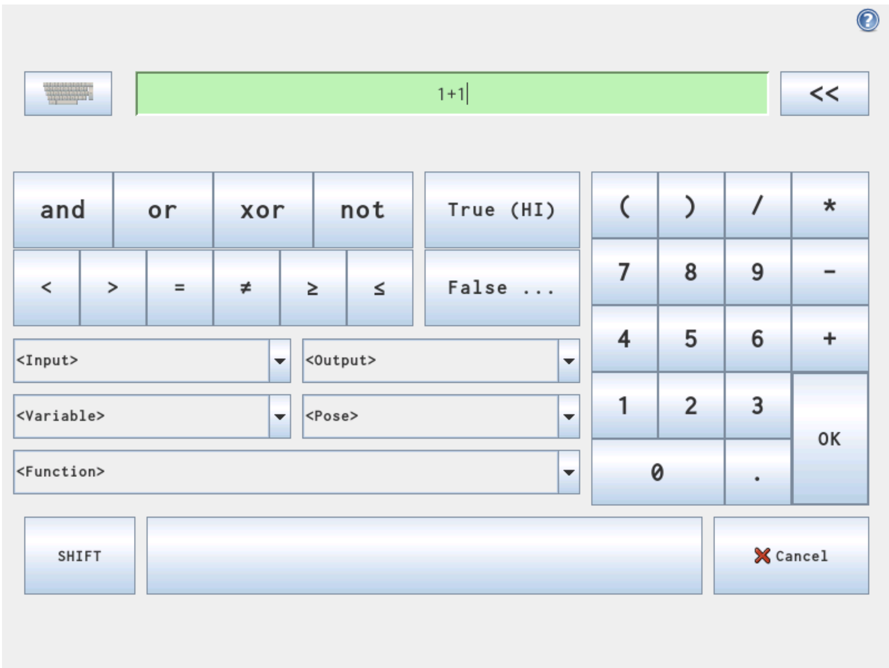


Simple number typing and editing facilities. In many cases, the unit of the typed value is displayed next to the number.

4.2.2 On-screen Keyboard

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

4.2.3 On-screen Expression Editor



While the expression itself is edited as text, the expression editor has a number of buttons and functions for inserting the special expression symbols, such as $*$ for multiplication and $\leq$ for less than or equal to. The keyboard symbol button in the top 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]=True and analog_in[0]<0.5
```

4.3 Robot Control

Use the "play", "Pause", "Stop" and "Step" buttons found at the bottom of the screen to start and stop robot programs. Also there is a speed slider which can slow down the speed of the program during verification of the robot program and movement.

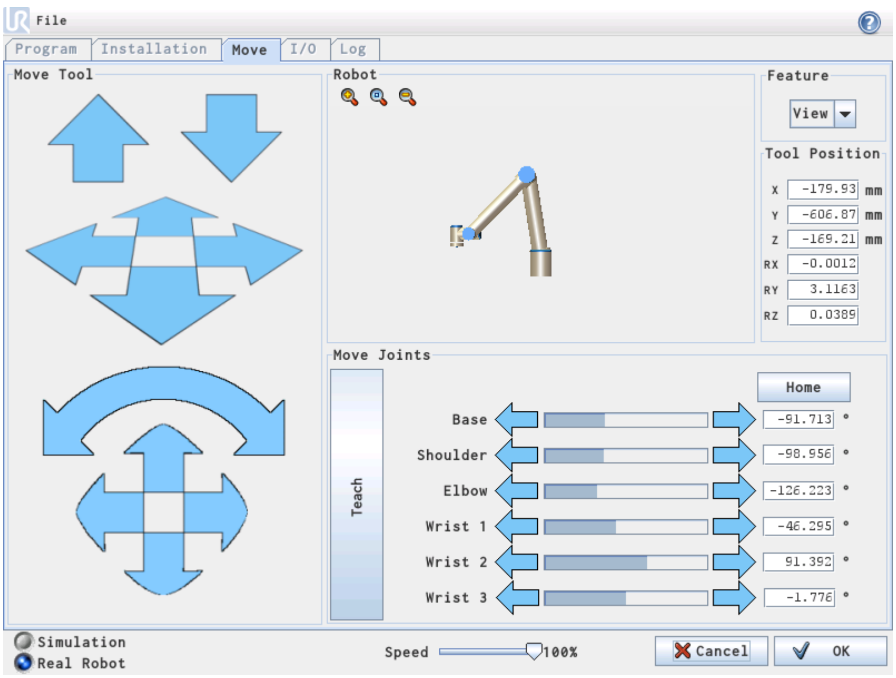
DANGER:
1. Make sure to stay outside the robot workspace when the "Play" button is pressed. The movement you programmed may be different than expected.

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.

4.3.1 Move Tab

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



Robot

The current position of the robot 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, select the "View" feature and rotate the viewing angle of the 3D drawing to match your view of the real robot.

Feature and tool position

At the top right part of the screen, the feature selector can be found. The features selector defines which feature to control the robot relative to, while below it, the boxes display the full coordinate value for the tool relative to the selected feature.

Values can be edited manually by clicking on the coordinate or the joint position.

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 TCP, drawn as a small blue ball.

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

Move Joints

Allows the individual joints to be controlled directly. Each joint can move from $-360°$ to $+360°$, which are the *joint limits* illustrated by the horizontal bar for each joint. If a joint reaches its joint limit, it cannot be driven any further away from $0°$.

Teach

While the 'Teach' button is held down, it is possible to physically grab the robot and pull it to where you want it to be. If the gravity setting (see 4.3.7) in the `Setup` tab is wrong, or the robot carries a heavy load, the robot might start moving (falling) when the 'Teach' button is pressed. In that case, just release the 'Teach' button again.

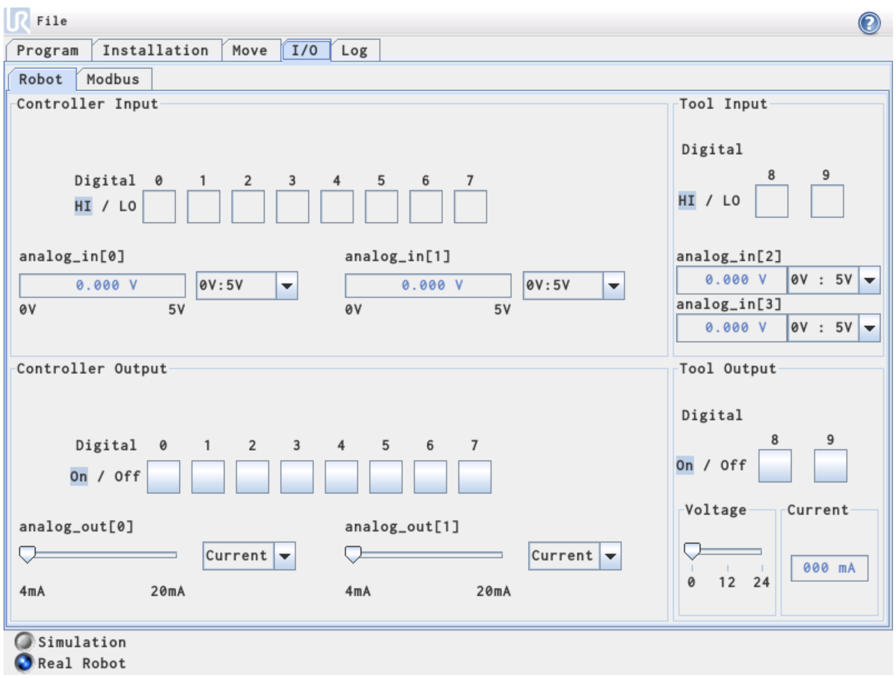
WARNING:

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 sittings and the robot mounting sittings are set correctly before operating the "Teach" button. If these sittings are not correct the robot will move when the "teach" button is activated.

3. The teach 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.

4.3.2 I/O Tab



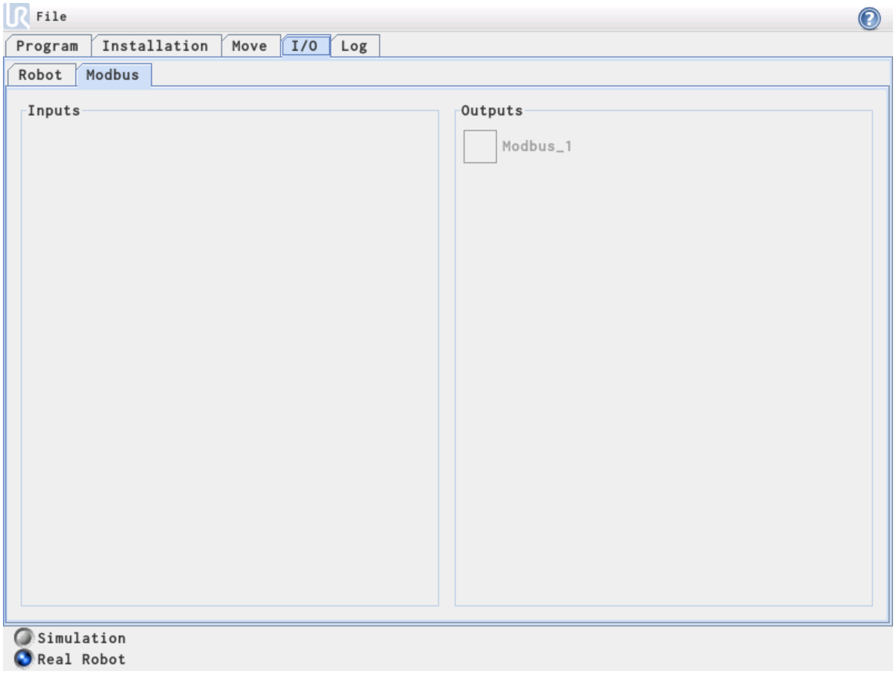
On this screen you can always monitor and set the live I/O signals from/to the robot. The screen displays the current state of the I/O, inluding during program execution. If anything is changed during program execution, the program will stop. At program stop, all output signals will retain their states. The screen is updated at only 10Hz, so a very fast signal might not display properly.

The electrical details of the signals are described in section 3.1.

Analog Range Settings The analog output can be set to either current (4-20mA) or voltage (0-10V) output. The analog input ranges adjusted to be from (-10-10V) to (0-5V). The settings will be remembered for eventual later restarts of the robot controller when a program is saved.

4.3.3 Modbus I/O

Here, the digital modbus 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.



Inputs

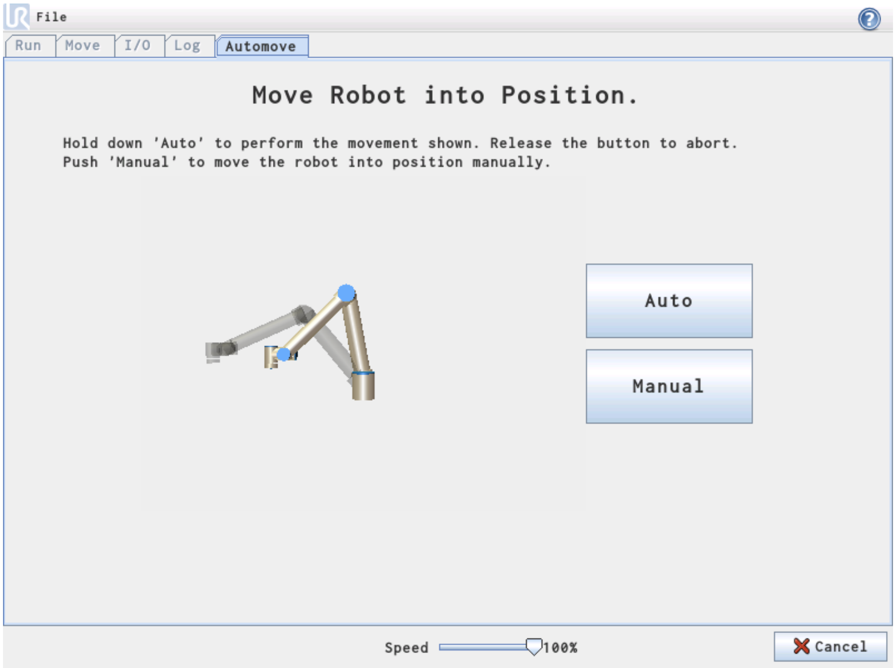
View the state of digital modbus inputs.

Outputs

View and toggle the state of digital modbus outputs. A signal can only be toggled if the choice for I/O tab control (described in 4.3.8) allows it.

4.3.4 AutoMove Tab

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



Animation

The animation shows the movement the robot is about to perform. Compare the animation with the position of the real robot and make sure that robot can safely perform the movement without hitting any obstacles.

CAUTION:
1. The automove function moves in joint space, not in linear (cartesian) space. Collision might damage robot or equipment.

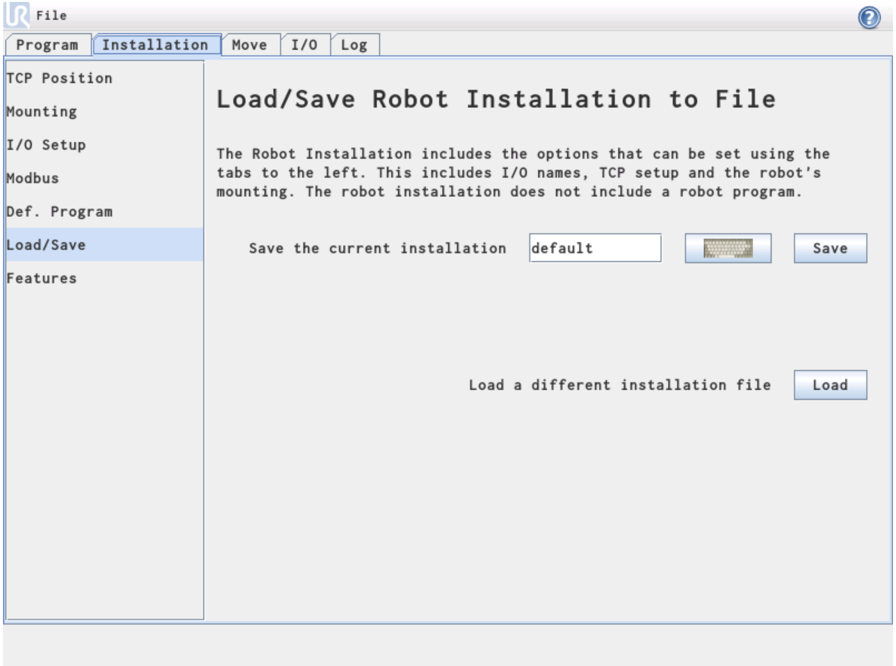
Auto

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

Manual

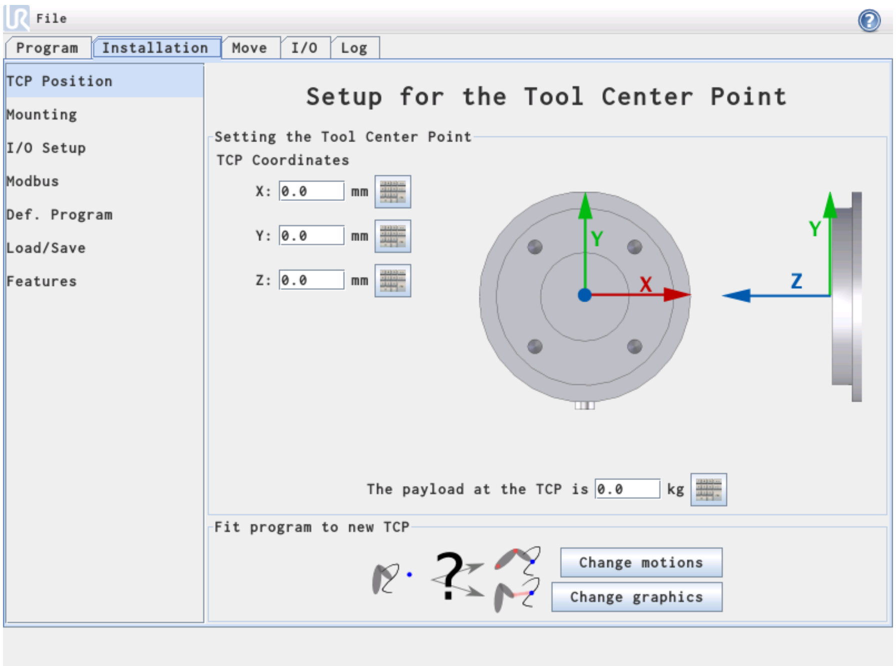
Pushing the `Manual` button will take you to the `MoveTab` where the robot can be moved manually. This is only needed if the movement in the animation is not preferable.

4.3.5 Installation → Load/Save



The installation covers aspects of how the robot is placed in its working environment, both mechanical mounting of the robot, and electrical connections to other equipment. These settings can be set using the various screens under the `Installation` tab. 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 needs to be saved to be preserved after power down. Saving an installation can be done either by pressing the `Save` button or by saving a program using the installation.

4.3.6 Installation → TCP Position



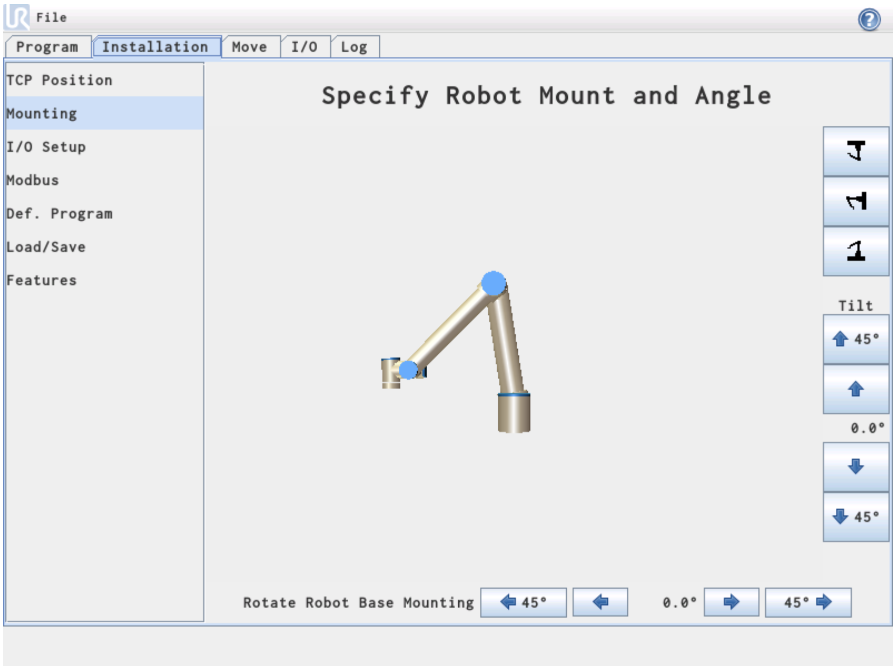
The *Tool Center Point* (TCP) is the point at the end of the robot arm that gives a characteristic point on the robot's tool. When the robot moves linearly, it is this point that moves in a straight line. It is also the motion of the TCP that is visualized on the graphics tab. The TCP is given relative to the center of the tool output flange, as indicated on the on-screen graphics.

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

The two buttons on the bottom of the screen are relevant when the TCP is changed.

- **Change Motions** recalculates all positions in the robot program to fit the new TCP. This is relevant when the shape or size of the tools has been changed.
- **Change Graphics** redraws the graphics of the program to fit the new TCP. This is relevant when the TCP has been changed without any physical changes to the tool.

4.3.7 Installation → Mounting



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

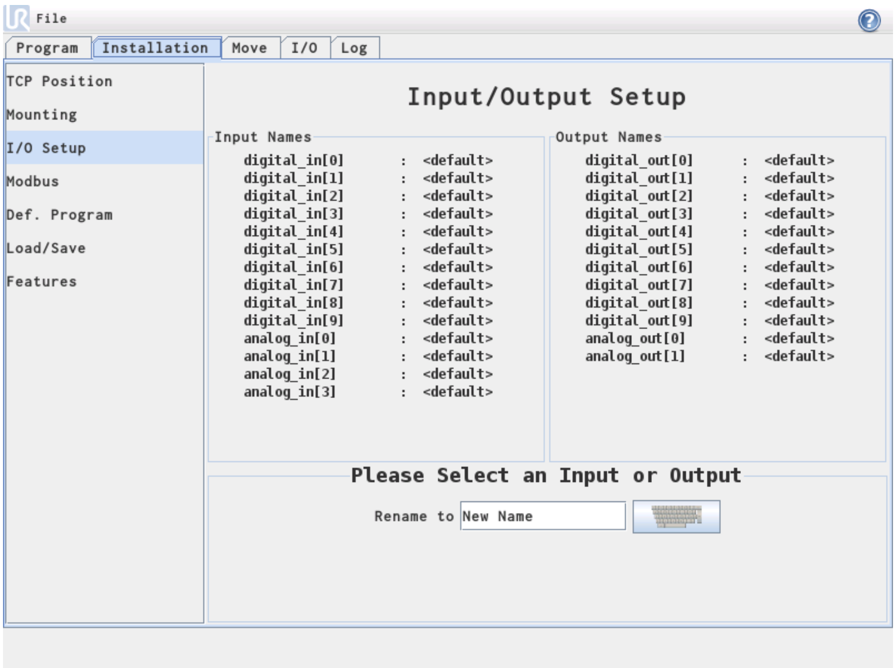
1. Making the robot look right on the screen.
2. Telling the controller about the direction of gravity.

The controller uses an advanced dynamics model to give the robot smooth and precise motions, and to make the robot hold itself when backdriven. For this reason, it is important that the mounting of the robot is set correctly.Warning: Failure to set the robot mounting correctly might result in frequent security stops, and/or a possibility that the robot will move when the teach button is pressed.

The default is that the robot is mounted on a flat table or floor, in which case no change is needed on this screen. However, if the robot is *ceiling mounted*, *wall mounted* or mounted at an angle this can be adjusted using the push-buttons. The buttons on the right side of the screen are for setting the angle of the robot'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 to match the actual mounting.

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

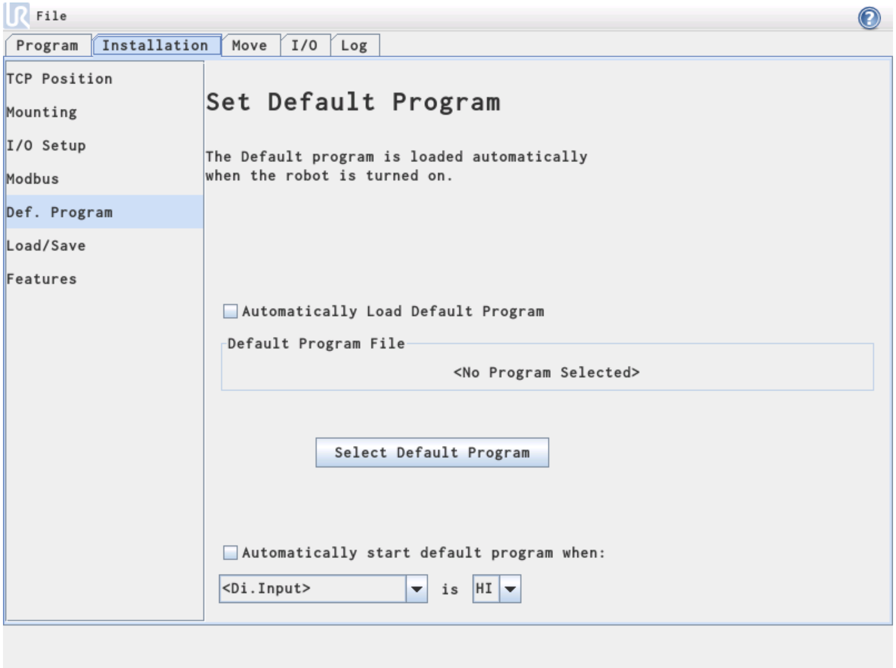
4.3.8 Installation → I/O Setup



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.

When an output is selected, a few options are enabled. Using the check box, a default value for the output can set to either low or high. This means that the output will be set to this value when a program is not running. If the check box is not checked, the output will preserve its current state after a program ends. 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.

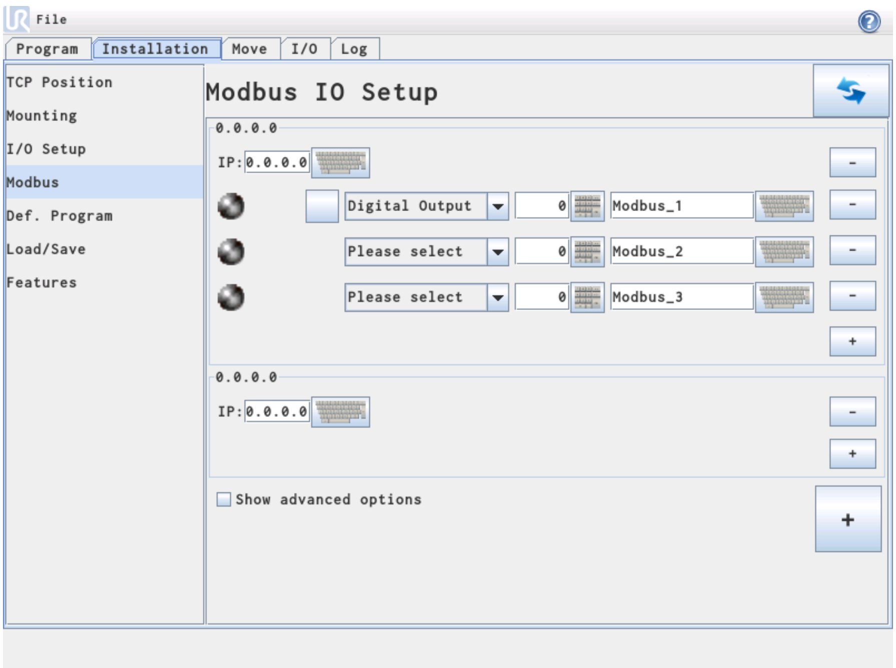
4.3.9 Installation → Default Program



The default program will be loaded when the control box is powered up.

4.3.10 Modbus I/O Setup

Here, the modbus I/O signals can be set up. Modbus units on specific IP addresses can be added/deleted and input/output signals (registers or digital) on these units can be added/deleted as well. Each signal must be supplied with a unique name. However, several signals with different names may reference the same modbus signal, but the user is advised to avoid this so as not to cause confusion for the programmer. Below, the different buttons and fields are explained in detail.



Refresh

Push this button to refresh the connectivity status of all modbus signals in the current installation.

Add unit

Push this button to add a new modbus unit to the robot installation.

Delete unit

Push this button to delete the modbus unit and all signals added to the 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 robot installation which can be found on the corresponding modbus unit.

Delete signal

Push this button to delete the modbus signal from the installation.

Set signal type

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

- **Digital input:** A digital input 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 is a one-bit quantity which can be set to either high or low according to the configuration of the corresponding modbus terminal. Until the value of this output has been set by the user, the value is read from the unit. This means that function code 0x01 (Read Coils) is used until the output has been set, and then when either 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. Until the value of the register has been set, the value of it is simply read. This means that function code 0x03 (Read Holding Registers) is used until the signal is set either by a robot program or by specifying a signal value in the "set signal value" field, and after that function code 0x06 (Write Single Register) is used onwards.

Set signal address

This field shows the address of the signal. Use the on-screen keypad to choose a different address. Valid addresses depends on the manufacturer and configuration of the modbus unit. It is necessary to have a good understanding of the internal memory map of the Modbus controller in order to make sure the signal address actually corresponds to what is the intention of the signal. Especially, it might be worth verifying the meaning of a signal address when different function codes are used. See 4.3.10 for a description of the function codes associated with the different signal types.

Set signal name

Using the on-screen keyboard, the user may give the signal a meaningful name which will provide a more intuitive programming of the robot using the signal. Signal names are unique which means that two signals cannot be assigned the same name. Signal names are restricted to be composed of no more than 10 characters.

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).

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 Modbus controller 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 that you consult the manual of your Modbus devices to verify their functionality with a changed slave address.

4.3.11 Features

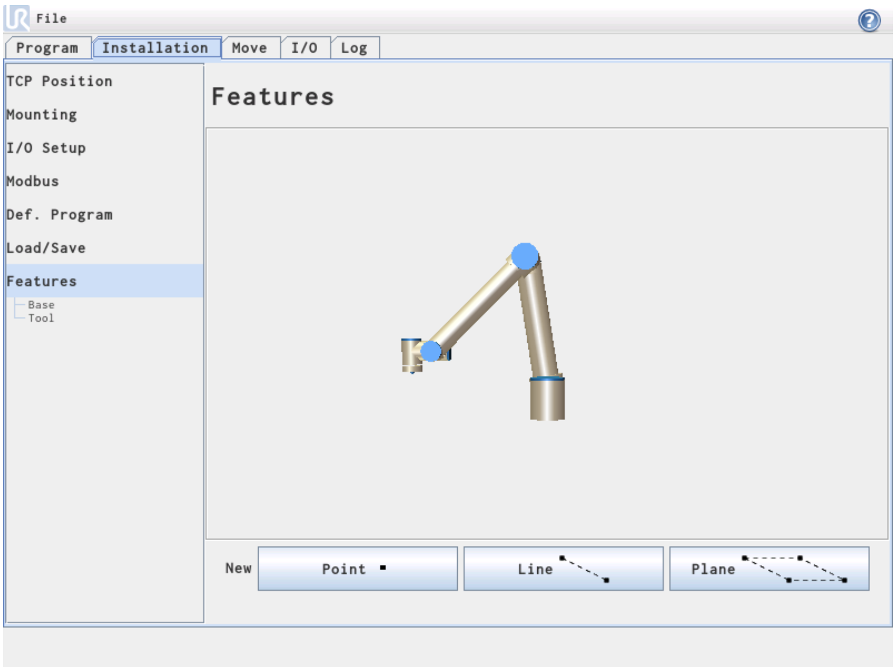
Customers that buy industrial robots generally want to be able to control or manipulate a robot, and to program the robot, relative to various objects and boundaries in the surroundings of the robot, such as machines, objects or blanks, fixtures, conveyers, pallets or vision systems. Traditionally, this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

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 move vertically downwards with the object, let the object loose, and then move the robot vertically upward again?

The meaning of such and similar questions is very straight forward to an average customer that intends to use a robot 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. 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

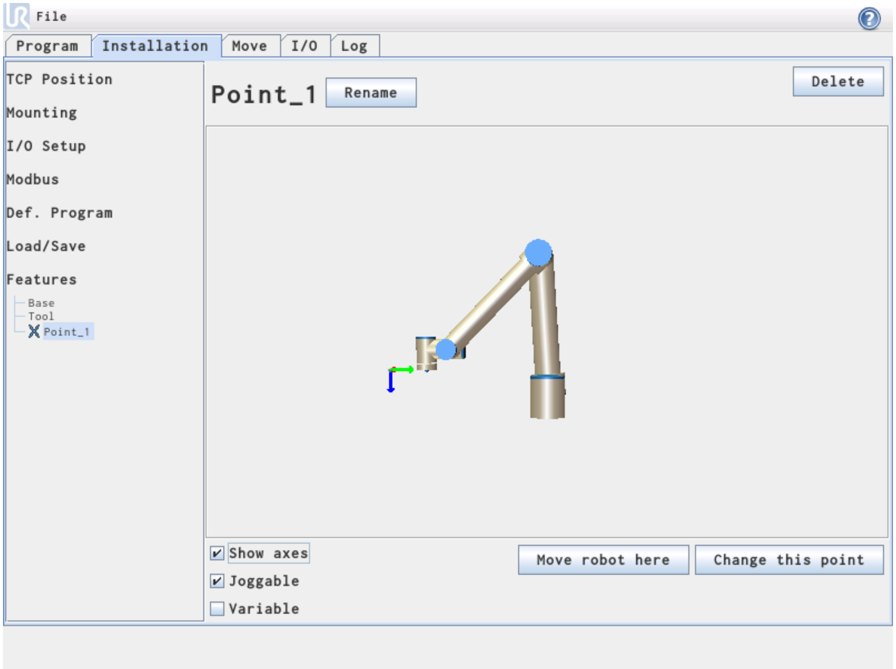
Use this button to set or change the selected feature. The Move screen will appear and a new position of the feature can be set.

Move Robot to Feature

Pressing this button will move the robot 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.

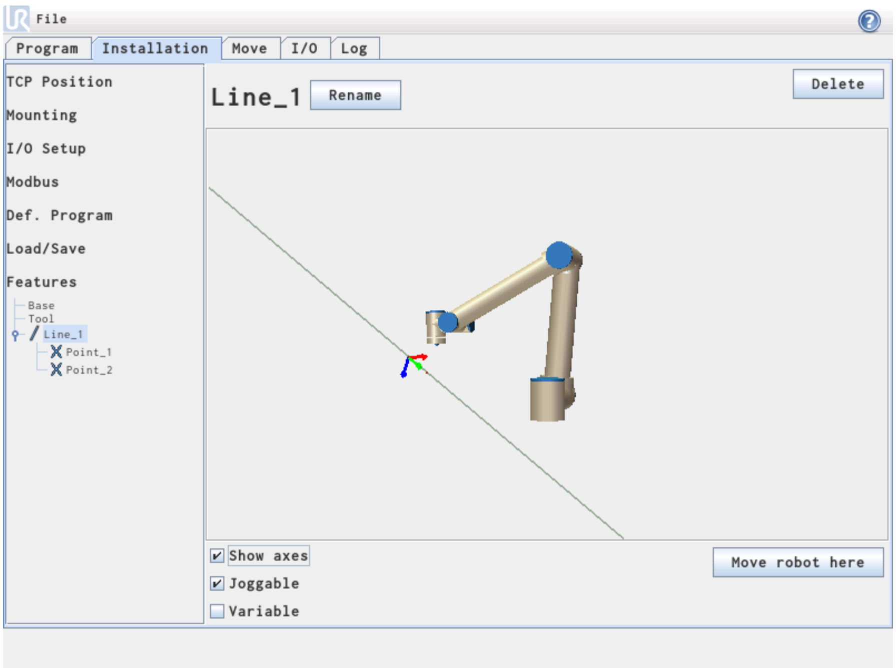
Add Point

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.



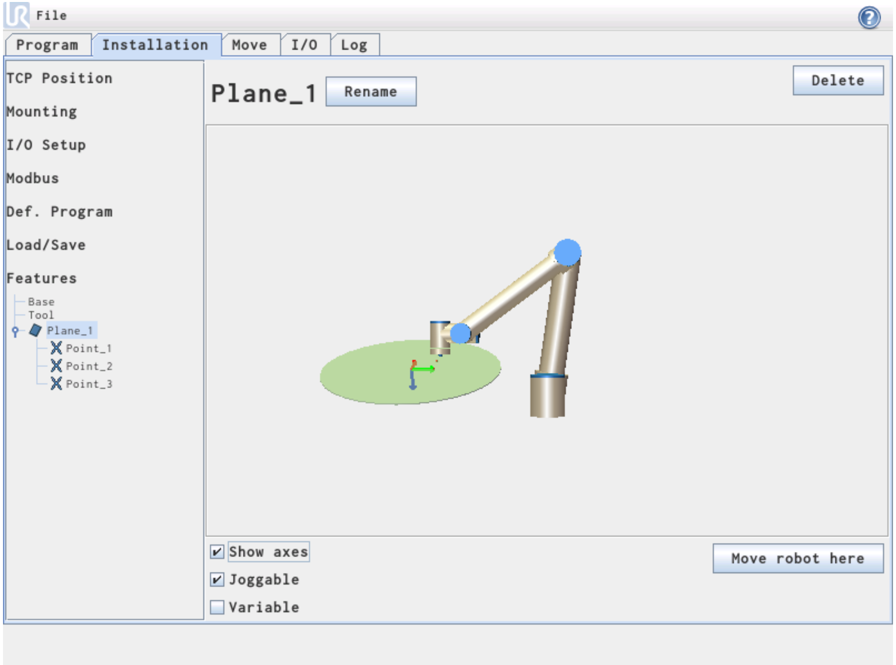
Add Line

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.

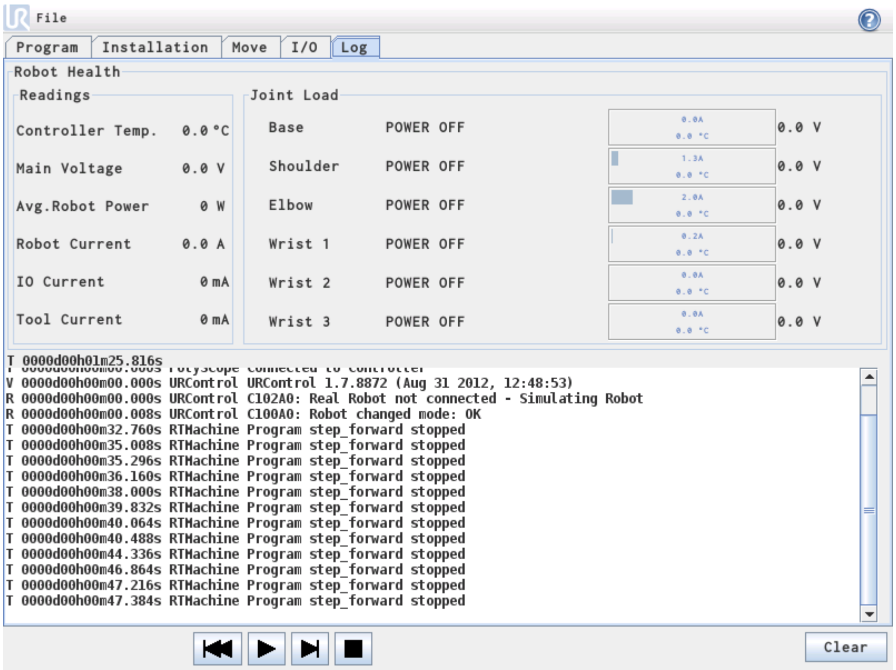


Add Plane

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.



4.3.12 Log Tab



Robot Health The top half of the screen displays the health of the robot. 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 shows the time of arrival of the message. The next column shows the sender of the message. The last column shows the message itself.

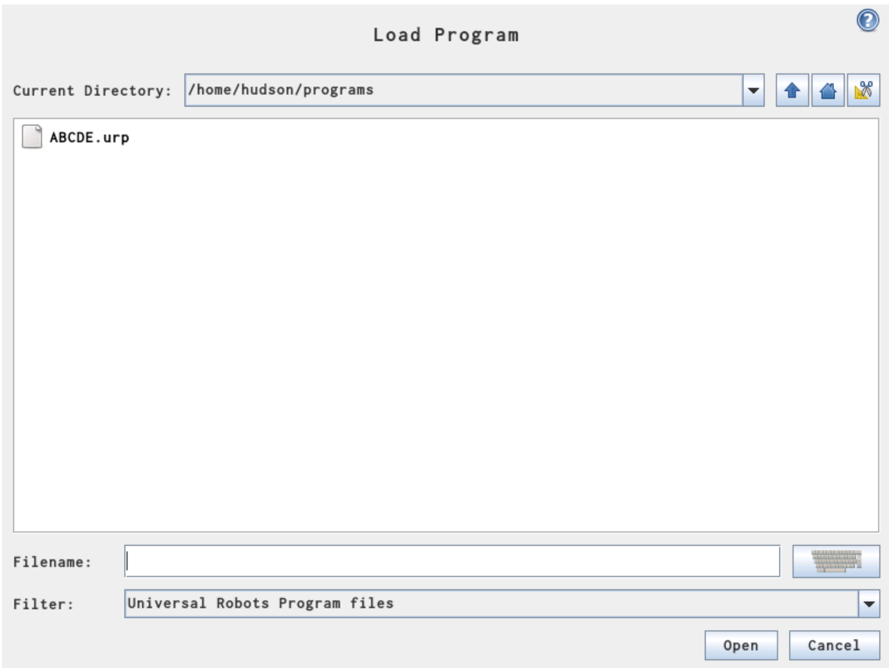
4.3.13 Load Screen

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 select and edit a files program.

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



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 4.3.13.

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.

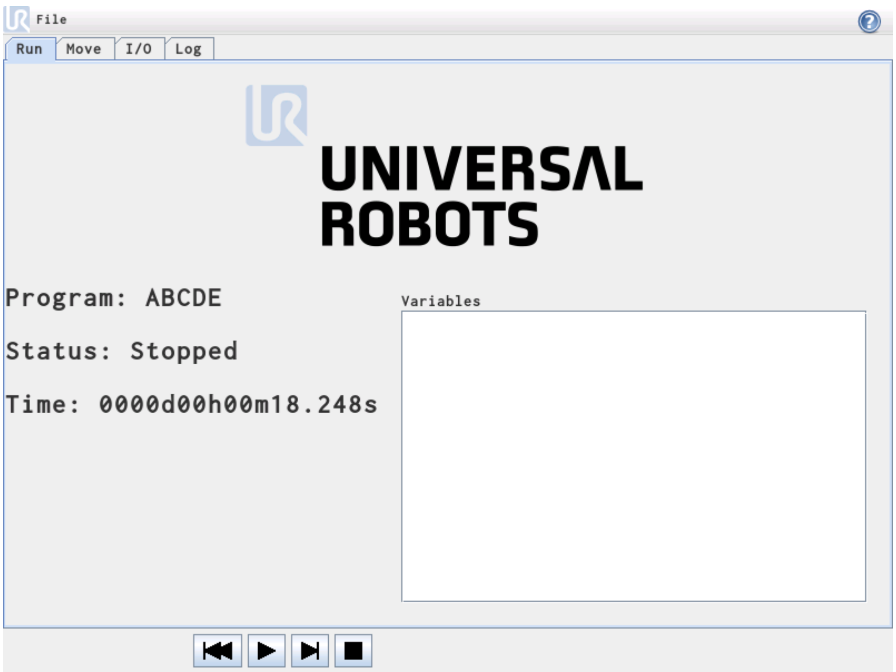
Open button Clicking on the Open button, will open the currently selected file and return to the previous screen.

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.

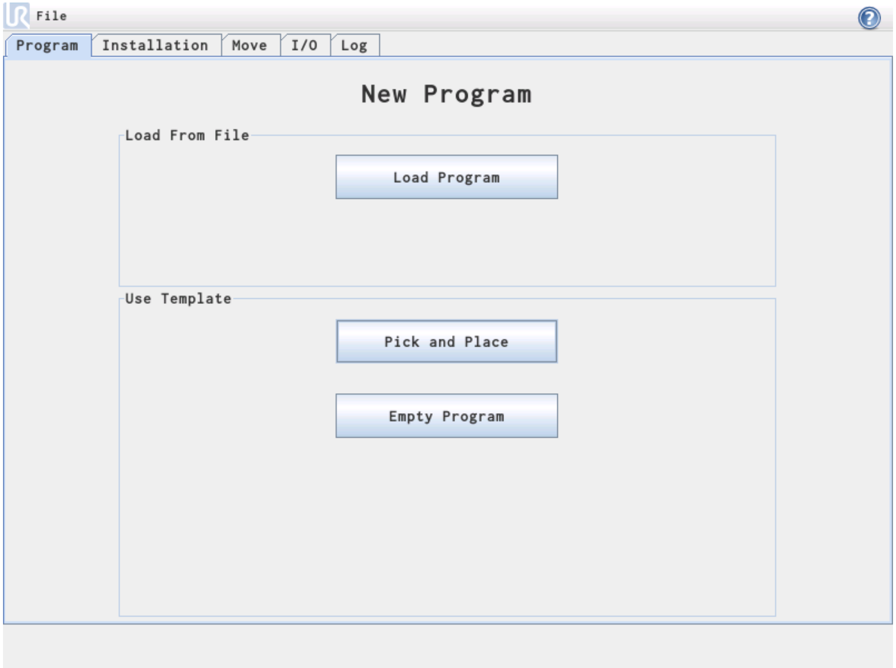
4.3.14 Run Tab



This tab provides a very simple way of operating the robot, with as few buttons and options as possible. This can be useful combined with password protecting the programming part of PolyScope (see section 4.5.5), to make the robot into a tool that can run exclusively pre-written programs.

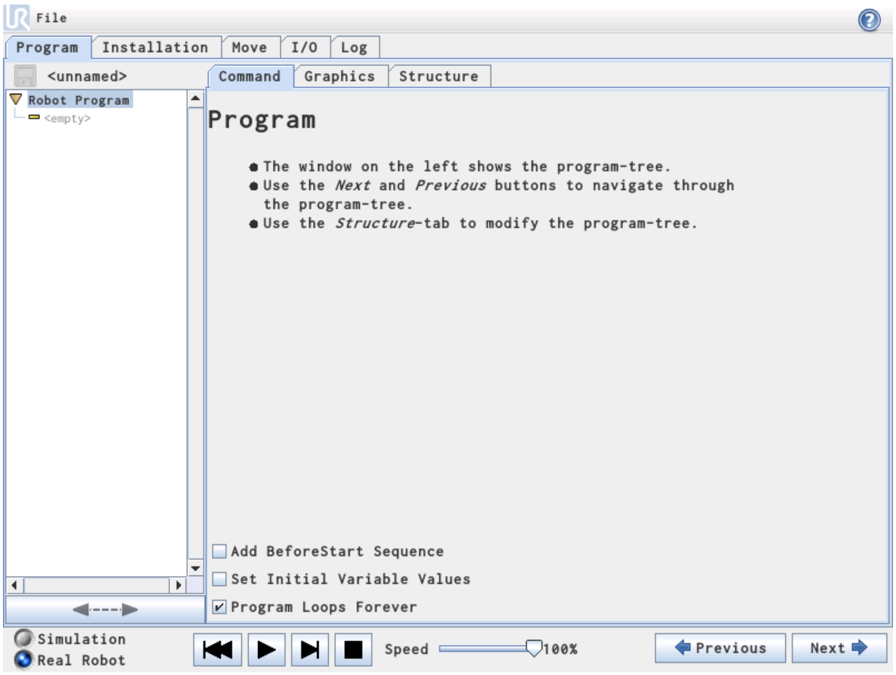
4.4 Programming

4.4.1 Program → New 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.

4.4.2 Program Tab



The program tab shows the current program being edited.

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

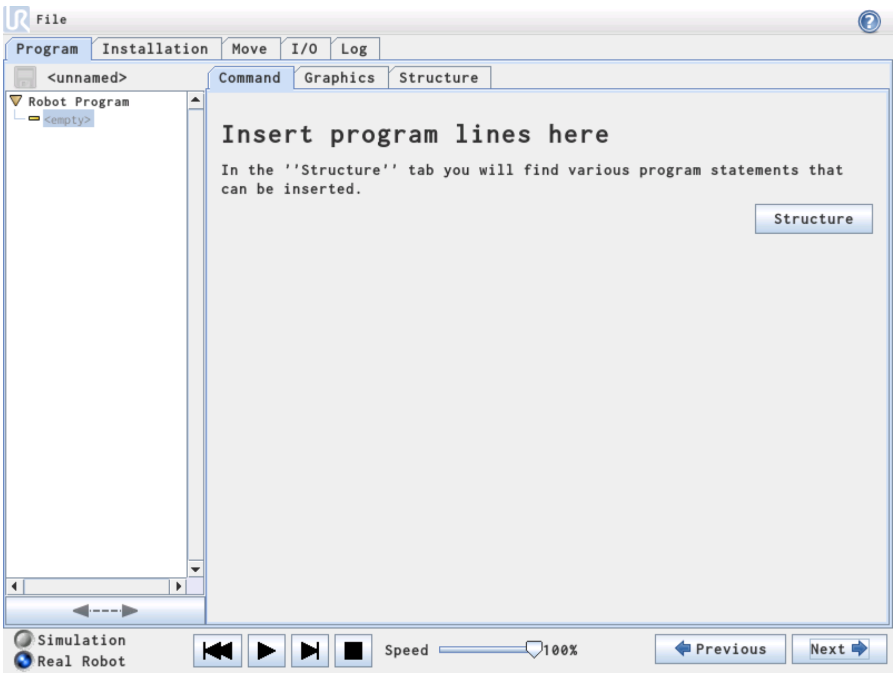
Structure tab, described in section 4.4.28. The program name is shown directly above the command list, with a small disk icon that can be clicked to quickly save the program.

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* allow you to adjust the program speed at any time, which directly affects the speed at which the robot moves. 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 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 will do.

While the program is being written, the resulting motion of the robot is illustrated using a 3D drawing on the Graphics tab, described in section 4.4.27.

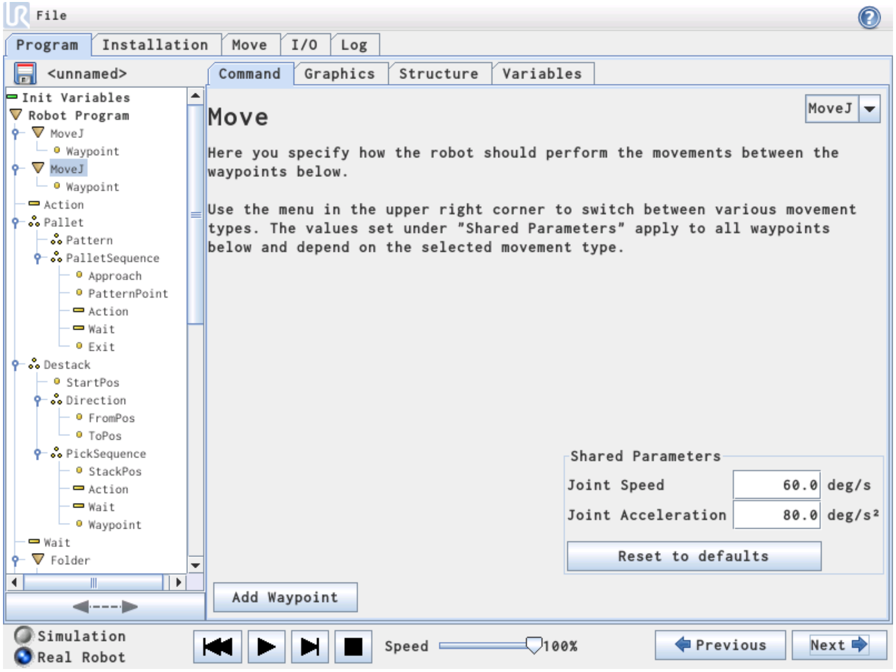
Next to each program command is a small icon, which is either red, yellow or green. A red icon means that there is an error in that command, yellow means that the command is not finished, and green means that all is OK. A program can only be run when all commands are green.

4.4.3 Program → Command Tab, <Empty>



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.

4.4.4 Program → Command Tab, Move



The Move command controls the robot motion through the underlying waypoints. Waypoints have to be under a Move command. The Move command defines the acceleration and the speed at which the robot 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. 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 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 per default a shared value between all the waypoint. A smaller value will make the path turn sharper whereas a higher value will make the path smoother. While the robot is moving through the waypoints with constant speed, the robot cannot wait for either an I/O operation or an operator action. Doing so will might stop the robots motion, or cause a security stop.

Feature selection

For *MoveL* and *MoveP* it is possible to select in 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.

- **Fixed feature**: If a fixed feature, such as e.g. *Base*, is selected this will not have any effect on *Fixed* and Relative waypoints. The behavior for *Variable* waypoints is described below.

- **Variable feature**: If any of the features in the currently loaded installation are selected to be variable, these corresponding variables will also be selectable in the feature selection menu. If a feature variable (named by the name of the feature and proceeded by "_var") is selected, the robot movements (except to *Relative* waypoints) will depend on 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. This means that the movements will only change if the feature variable is actively changed by the robot program.

- **Variable waypoint**: When the robot moves to a variable waypoint, the tool target position will always be calculated as the coordinates of the variable in the space of the selected feature. Therefore, the robot movement for a variable waypoint will always change if another feature is selected.

The settings of the Shared Parameters of a Move command apply to the path from the robot's current position 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.

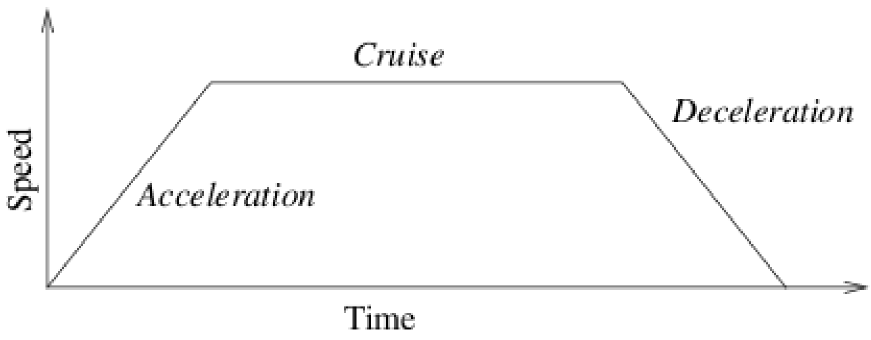
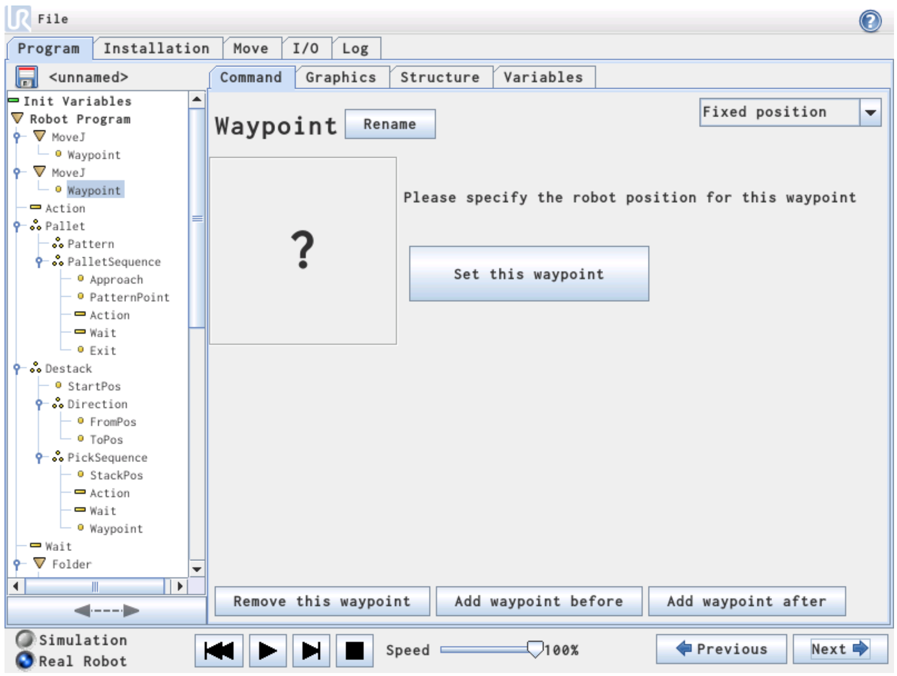


Figure 4.1: Speed profile for a motion. The curve is divided into three segments: *acceleration*, *cruise* and *deceleration*. The level of the *cruise* phase is given by the speed setting of the motion, while the steepness of the *acceleration* and *deceleration* phases is given by the acceleration parameter.

4.4.5 Program → Command Tab, Fixed Waypoint



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

4.4.6 Setting the waypoint

Press this button to enter the Move screen where you can specify the robot position for this waypoint. If the waypoint is placed under a Move command in linear space (`moveL` or `moveP`), there need 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.

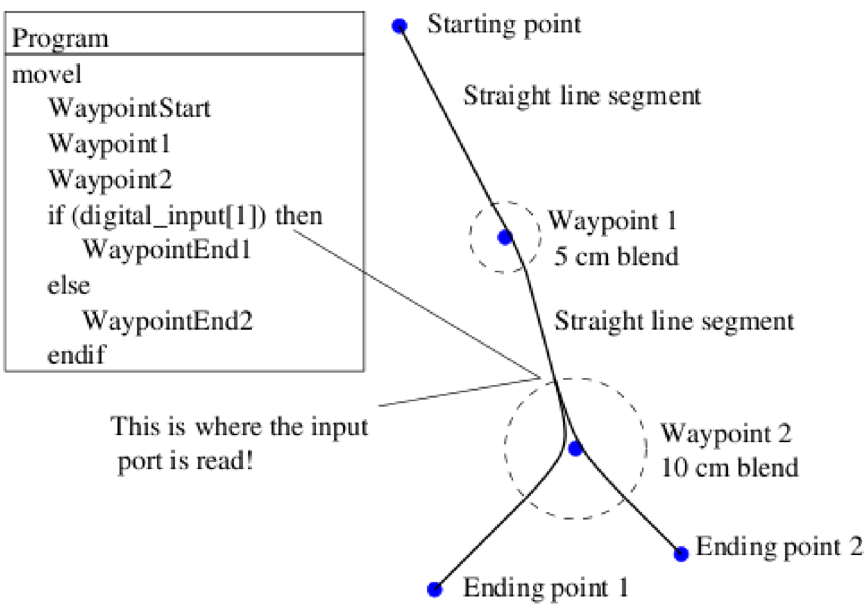
Blend radius

If a blend radius is set, the robot trajectory blends around the waypoint, allowing the robot 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

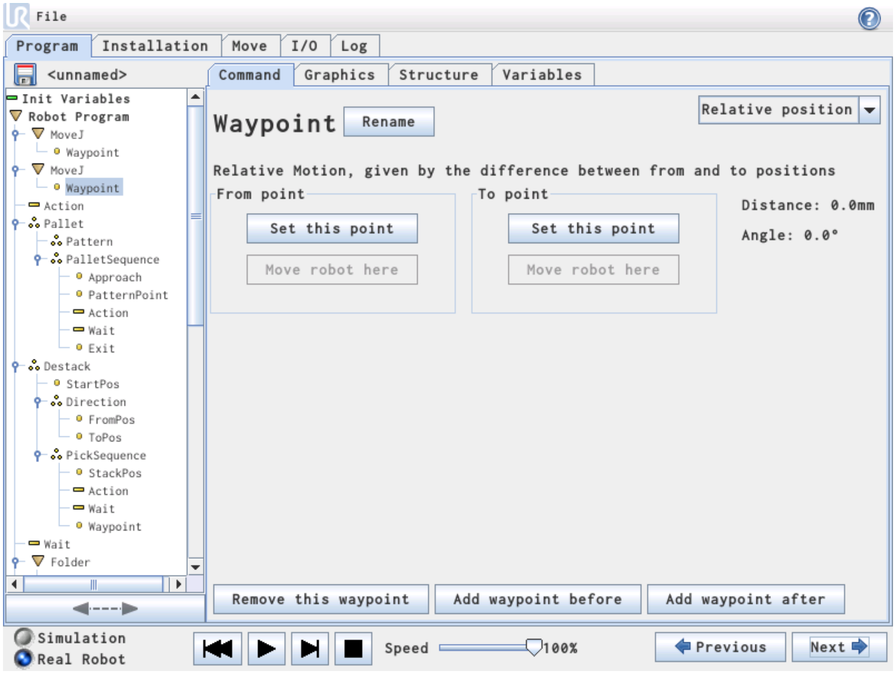
If a waypoint is a stop point with an I/O command as the next command, the I/O command is executed when the robot stops at the waypoint. However, if the waypoint has a blend radius, the following I/O command is executed when the robot enters the blend.

Example



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 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 allow the robot to select the right blend path.

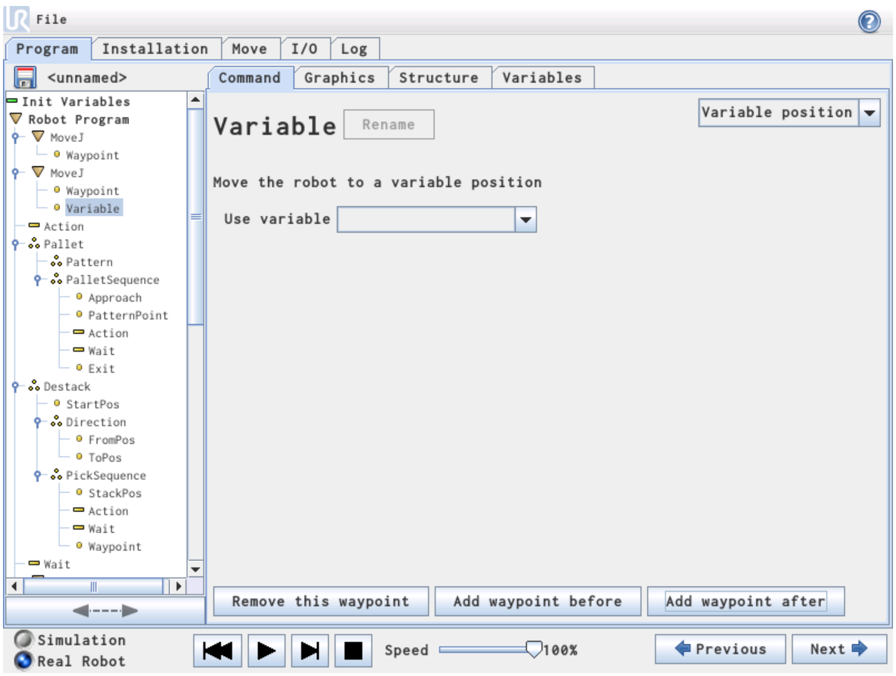
4.4.7 Program → Command Tab, Relative Waypoint



A waypoint with the position given relative to the robot'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 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.

4.4.8 Program → Command Tab, Variable Waypoint

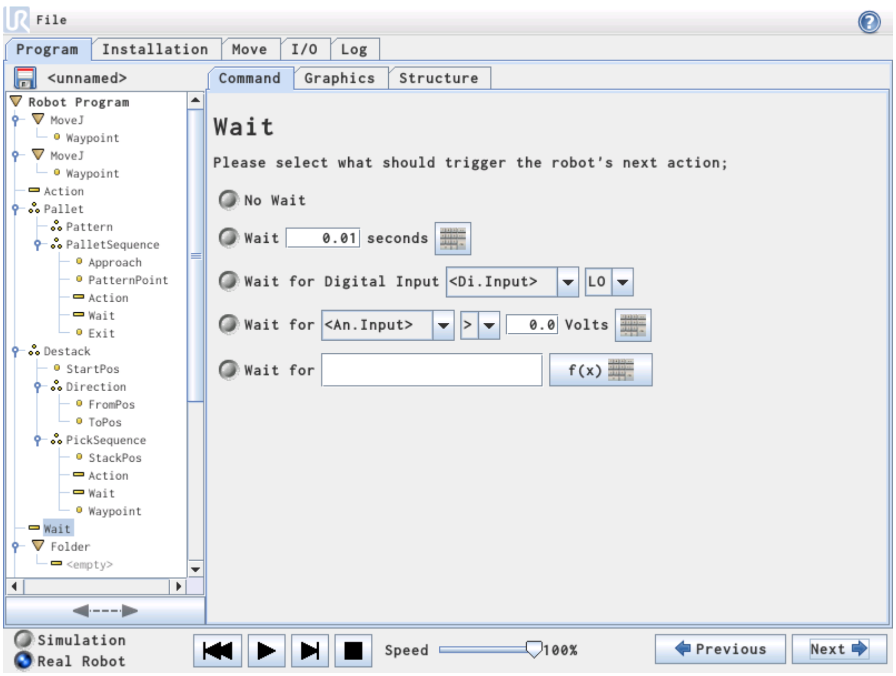


A waypoint with the position given by a variable, in this case `calculated_pos`. The variable has to be a *pose* such as `var=p[0.5,0.0,0.0,3.14,0.0,0.0]`. The first three are *x,y,z* and the last three are the orientation given as a *rotation vector* given by the vector *rx,ry,rz*. The length of the 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 always moves linearly to a variable waypoint.

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

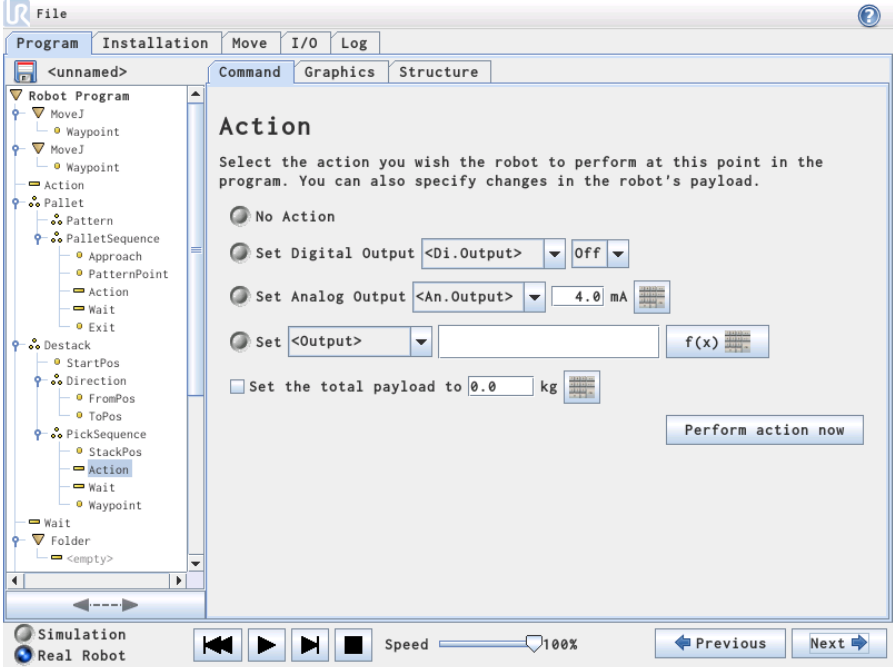
```
var_1=p[0,0,0.02,0,0,0]
MoveL
  Waypoint_1 (varibale position): Use variable=var_1, Feature=Tool
```

4.4.9 Program → Command Tab, Wait



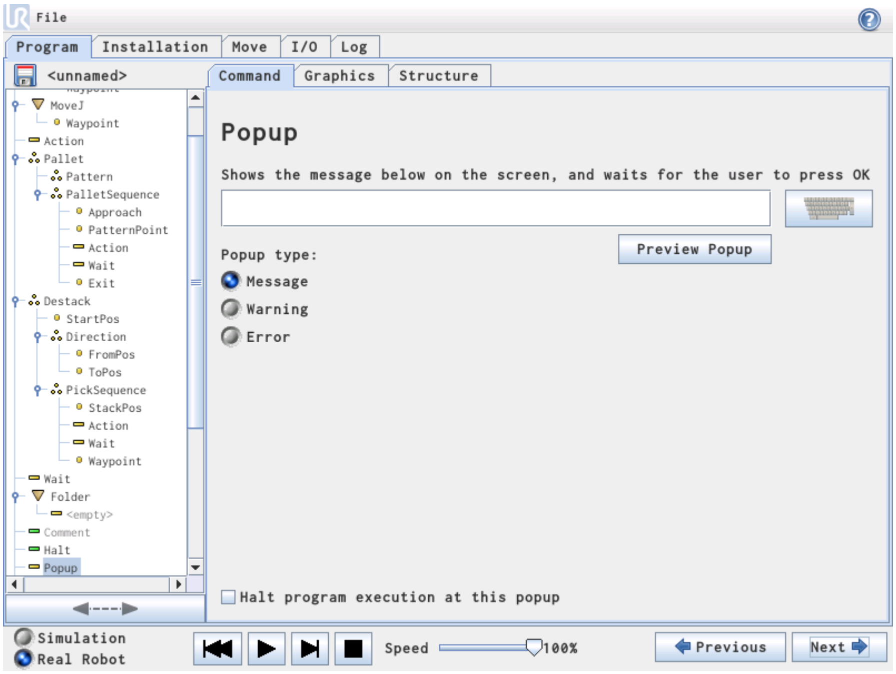
Waits for a given amount of time or for an I/O signal.

4.4.10 Program → Command Tab, Action



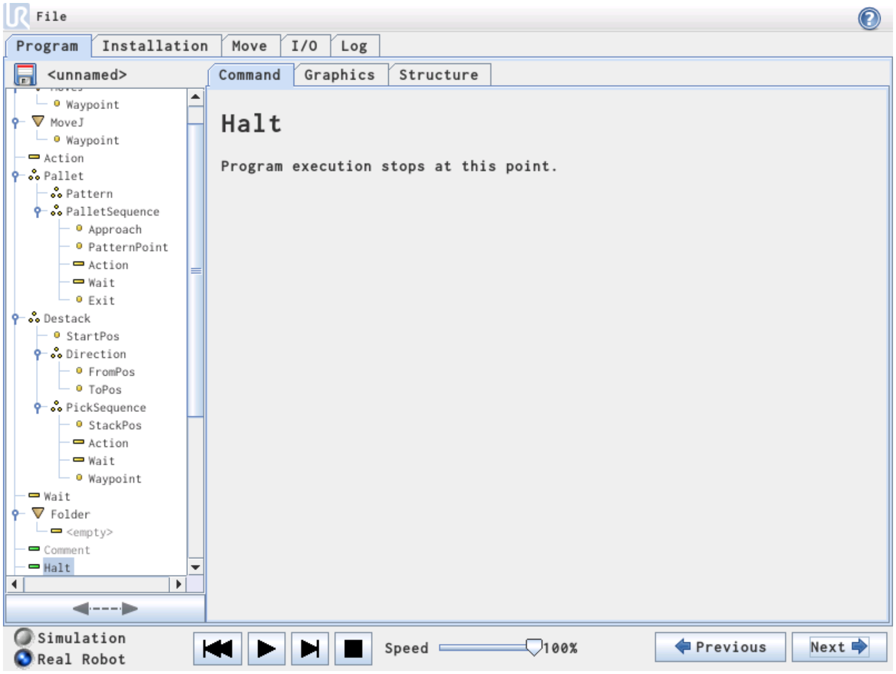
Sets either digital or analog outputs to a given value. Can also be used to set the payload of the robot, for example the weight that is picked up as a consequence of this action. Adjusting the weight can be neccesary to prevent the robot from security stopping unexpectedly, when the weight at the tool is different to that which is expected.

4.4.11 Program → Command Tab, Popup



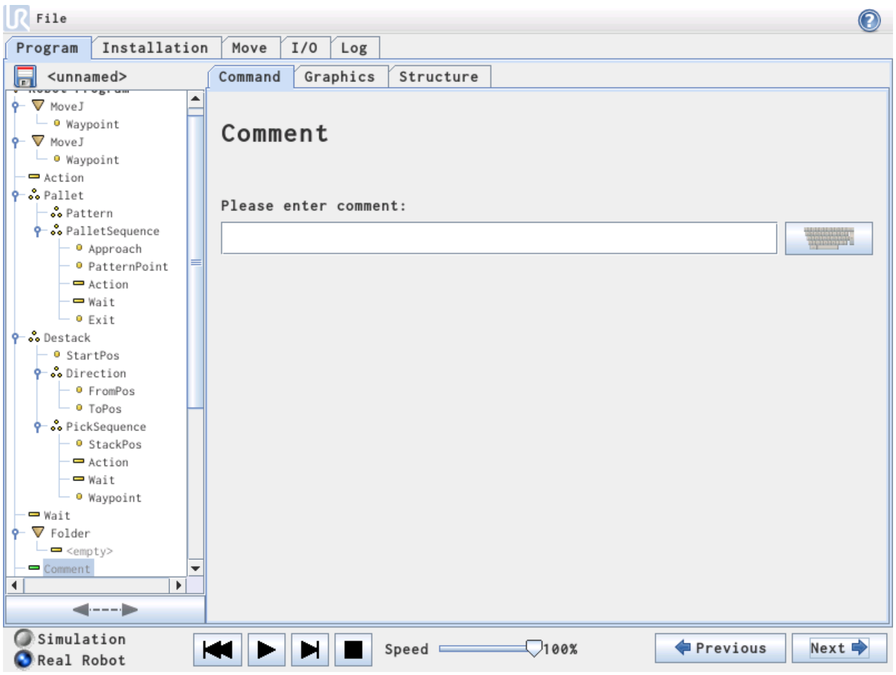
The popup is a message that appears on the screen when the program reaches this command. The style of the message can be selected, and the text itself can be given using the on-screen keyboard. The robot waits for the user/operator to press the "OK" button under the popup before continuing the program.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

4.4.12 Program → Command Tab, Halt



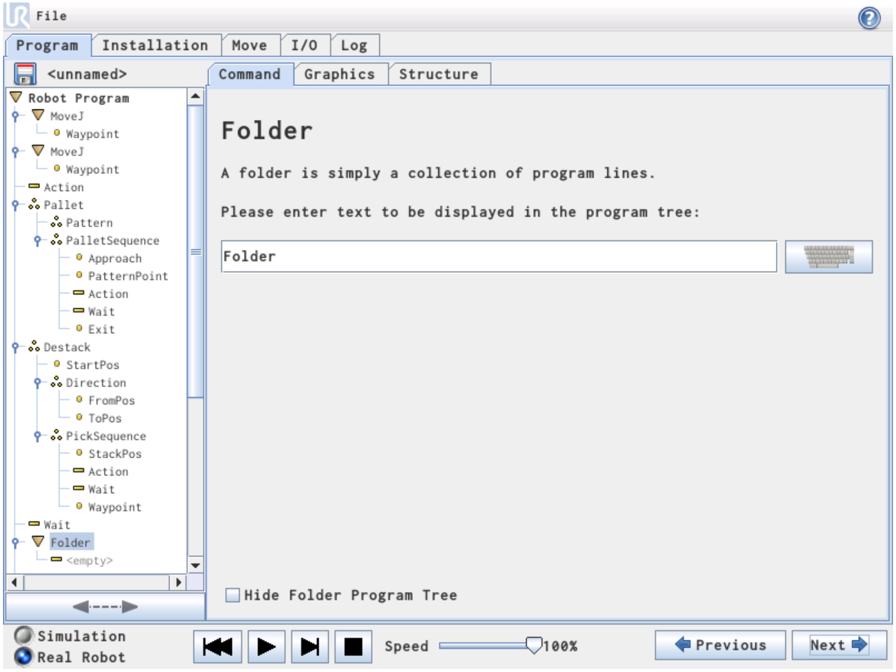
The program execution stops at this point.

4.4.13 Program → Command Tab, Comment



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

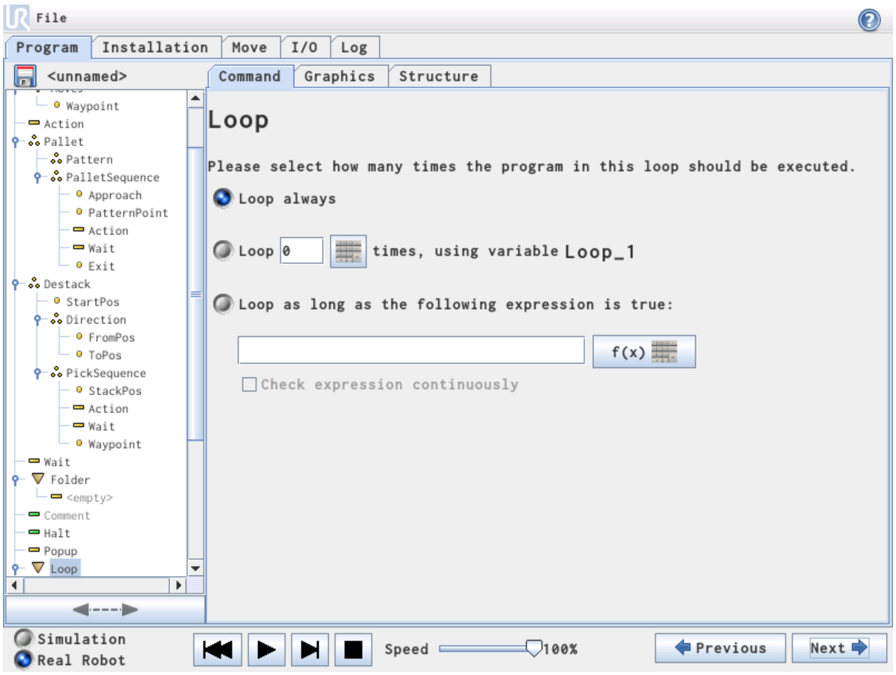
4.4.14 Program → Command Tab, Folder



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

A folder does not in itself do anything.

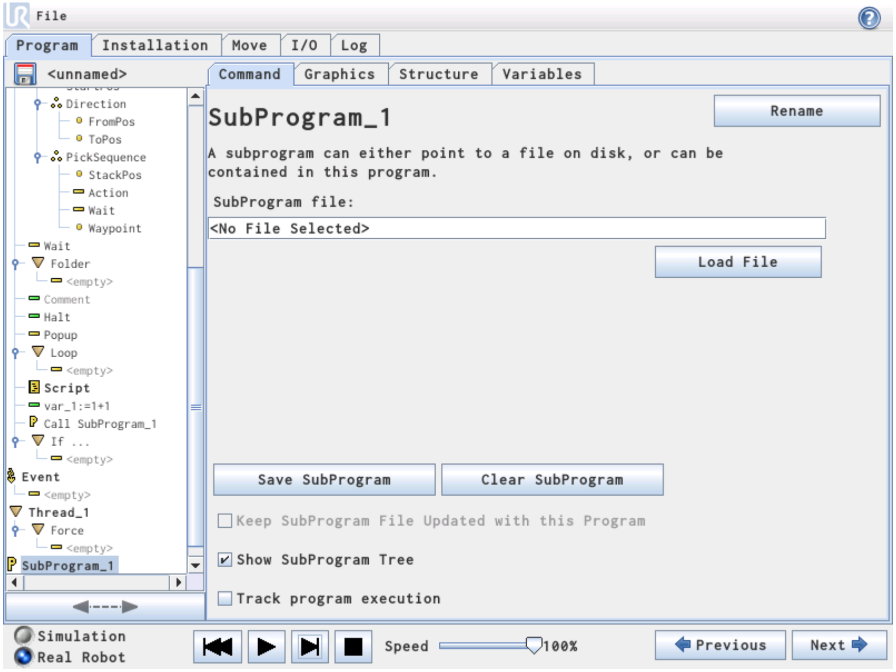
4.4.15 Program → Command Tab, Loop



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

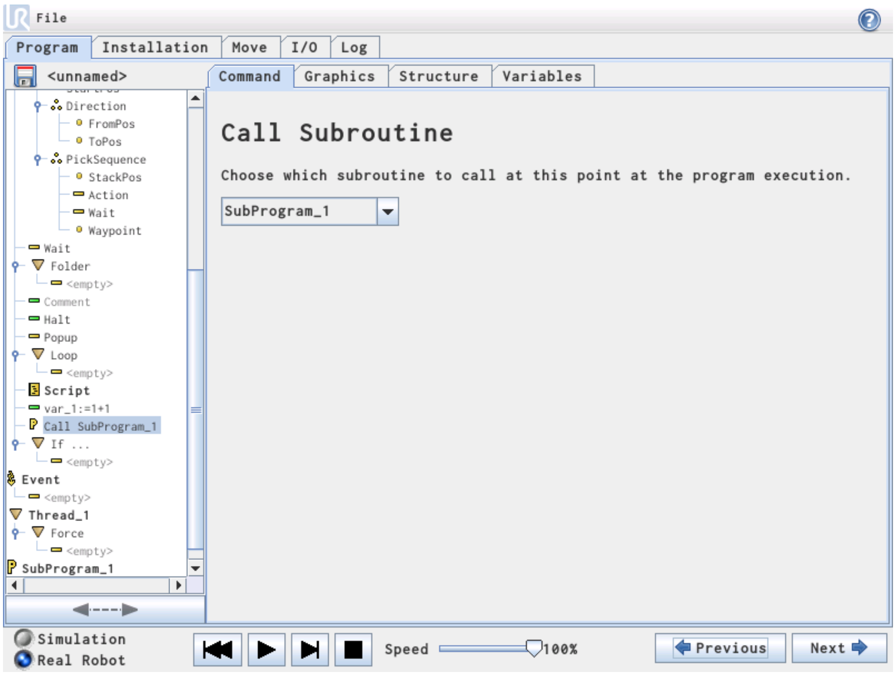
When looping using an expression as end condition, PolyScope provides an option for continuously evaluating that expression, so that the "loop" can be interrupted anytime during its execution, rather than just after each iteration.

4.4.16 Program → Command Tab, SubProgram



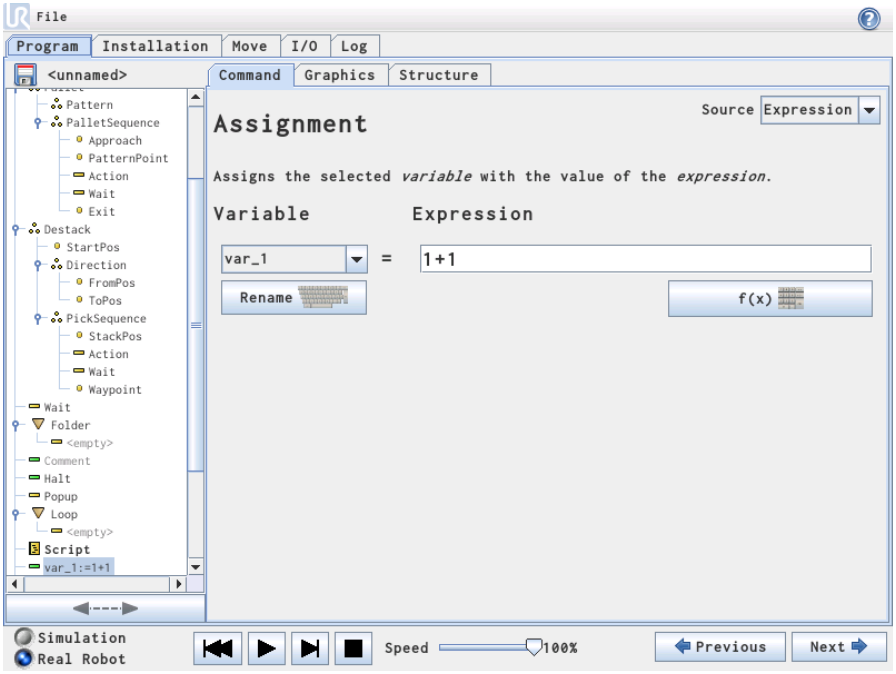
A Sub Program can hold program parts that are needed several places. A Sub Program can be a seperate file on the disk, and can also be hidden to protect against accidental changes to the SubProgram.

Program → Command Tab, Call SubProgram



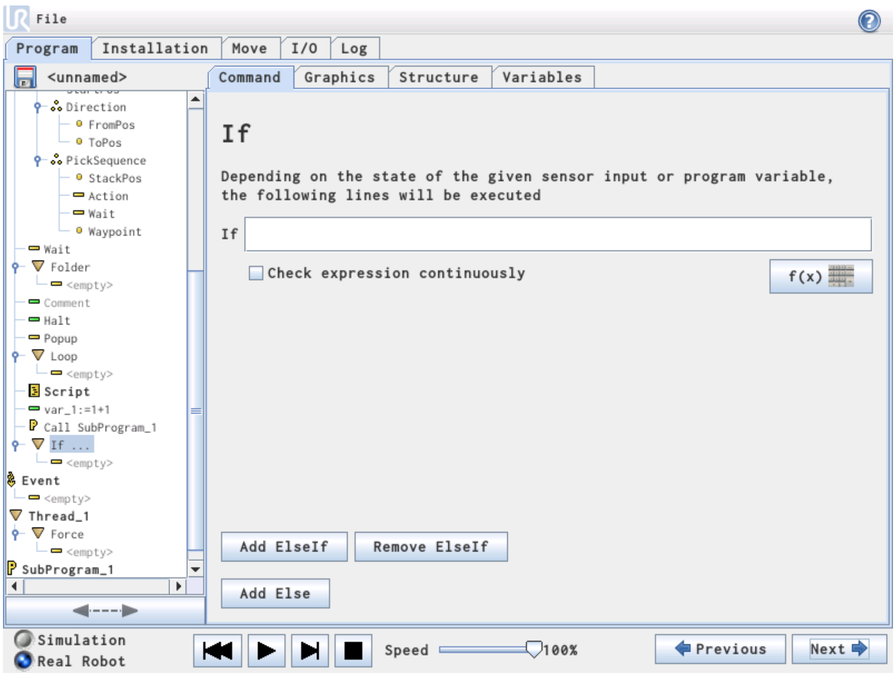
A call to a sub program will run the program lines in the sub program, and then return to the following line.

4.4.17 Program → Command Tab, Assignment



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.

4.4.18 Program → Command Tab, If

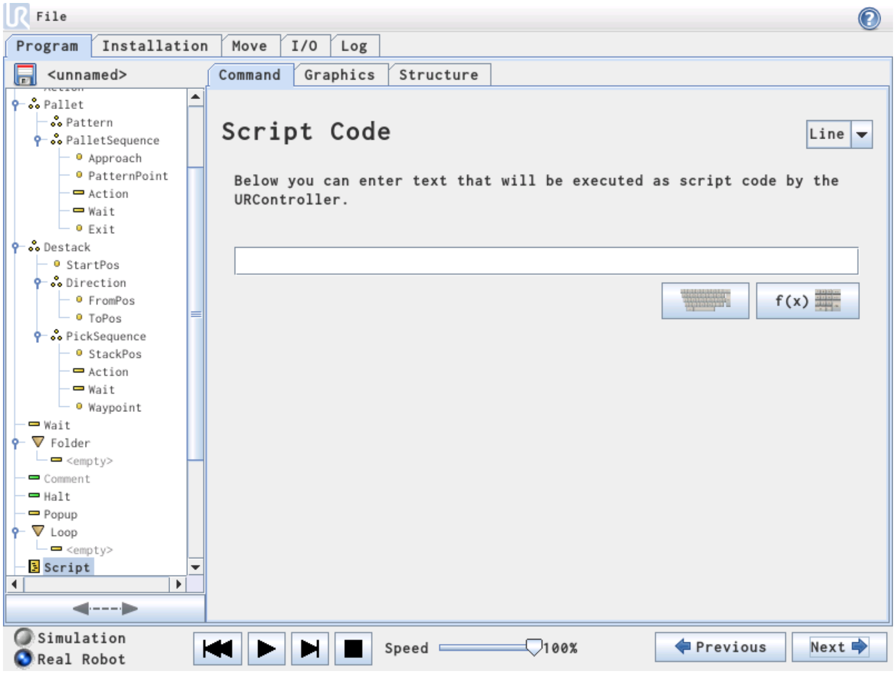


An "if..then..else" construction can make the robot change its behavior based on sensor inputs or variable values. Use the expression editor to describe the condition under which the robot should proceed to the sub-commands of this `If`. If the condition is evaluated to `True`, the lines inside this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

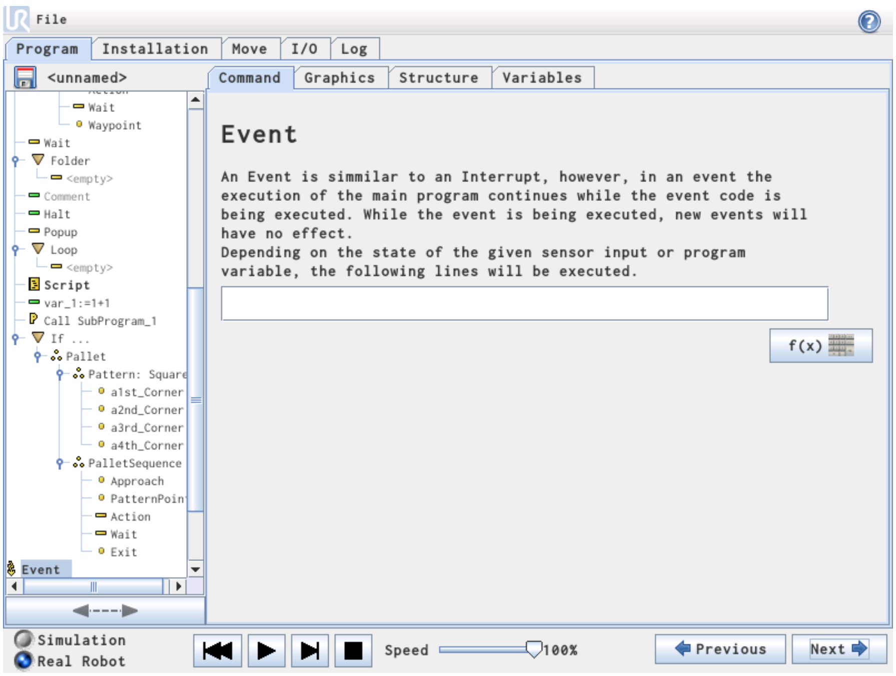
4.4.19 Program → Command Tab, Script



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.

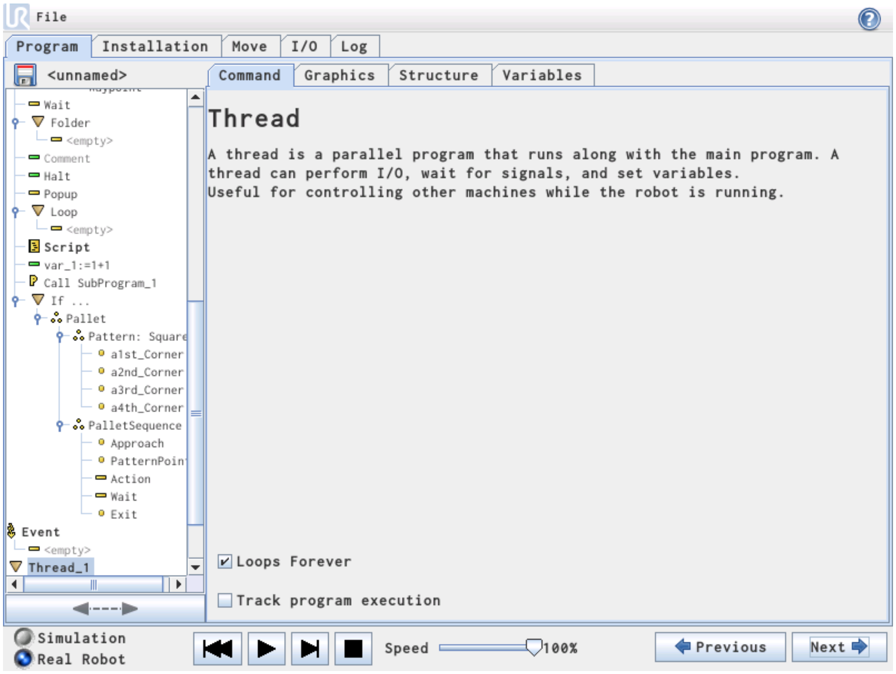
If the "File" option in the top left corner is choosen, it is possible to create and edit script programs files. This way, long and complex script programs can be used together with the operator-friendly programming of PolyScope.

4.4.20 Program → Command Tab, Event



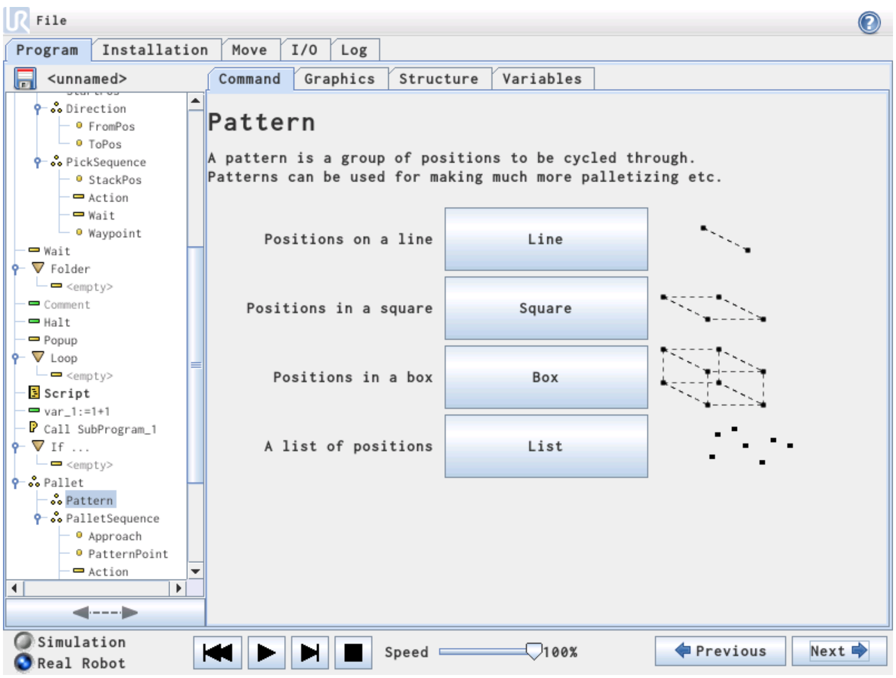
An event can be used to monitor an input signal, and perform some action or set a variable when that input signal goes high. For example, in the event that an output signal goes high, the event program can wait for 100ms and then set it back to low again. This can make the main program code a lot simpler in the case on an external machine triggering on a rising flank rather than a high input level.

4.4.21 Program → Command Tab, Thread



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

4.4.22 Program → Command Tab, Pattern



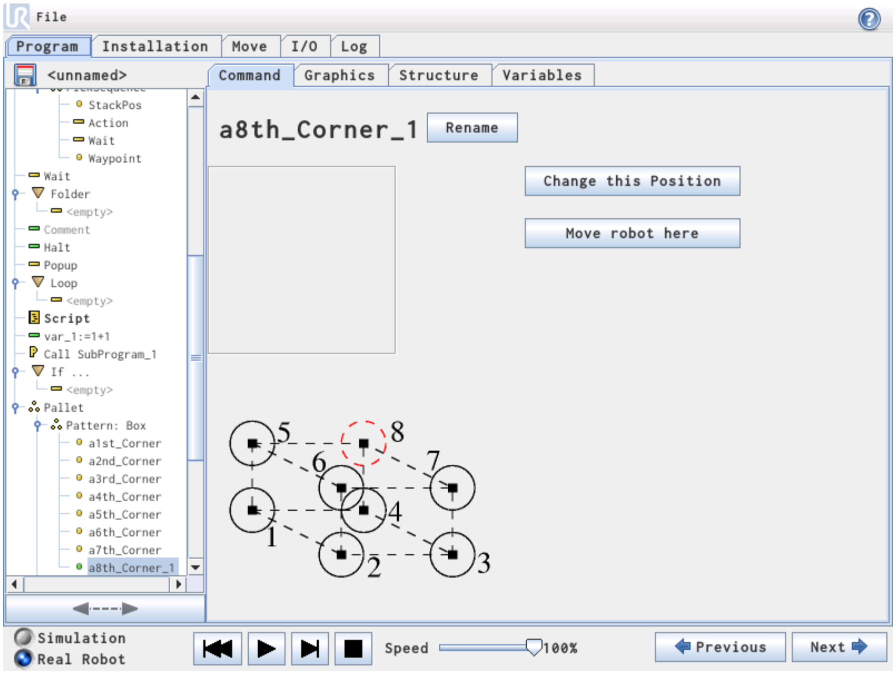
The Pattern command can be used to cycle through positions in the robots program. The pattern command corresponds to one position at each execution.

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.



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.

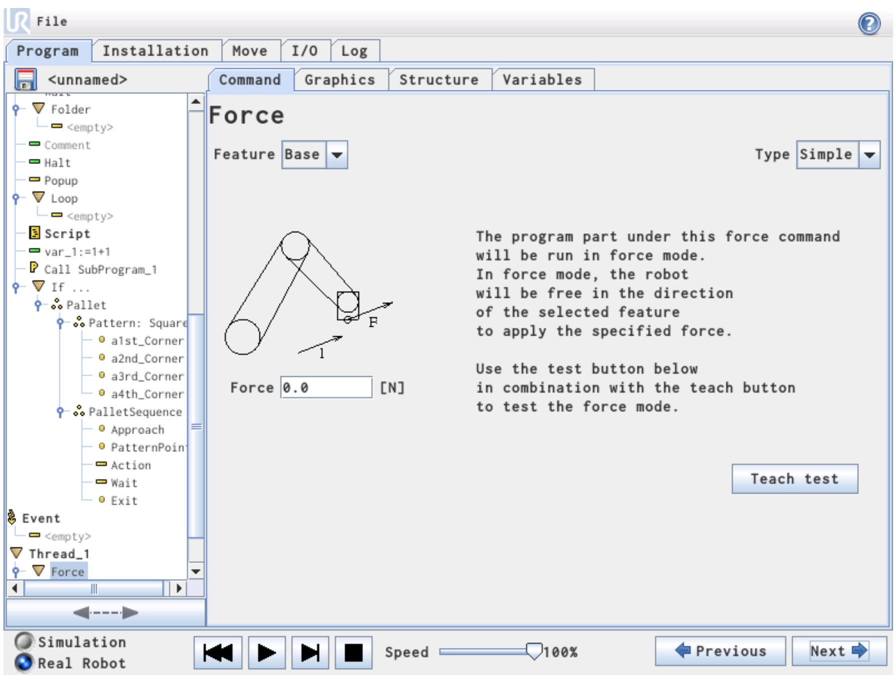
4.4.23 Program → Command Tab, Force

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

WARNING:
1. 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.



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 4.3.11.

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 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 is not moving: If force mode is entered with the robot standing still, there will no compliant axes until the tcp speed is above zero. If, later on while still in force mode, the robot 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 (4.4.27), 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 will adjust its position to achieve the selected force.
- **Non-compliant**: The robot 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

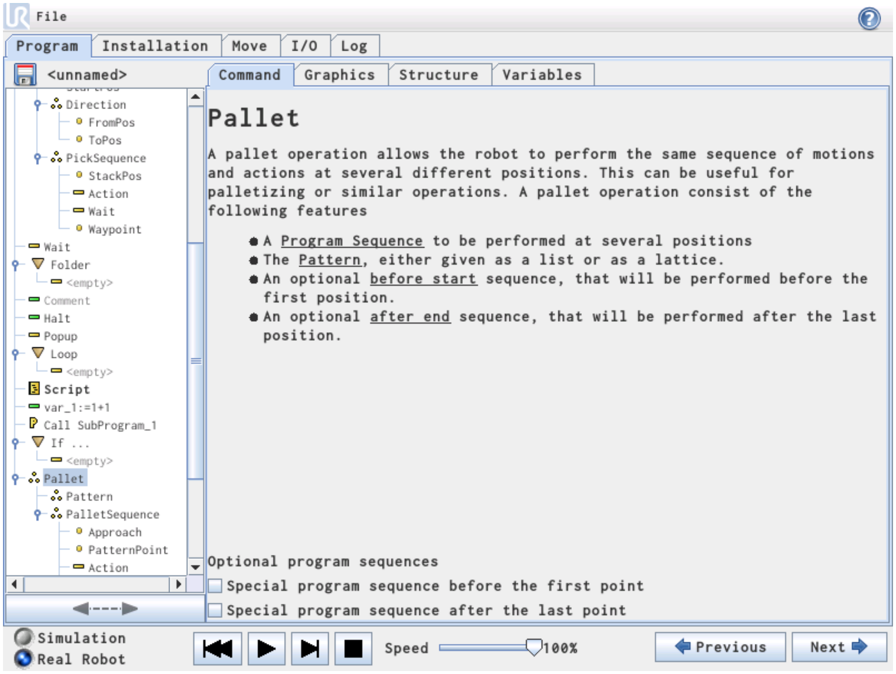
For all axes a limit can be set, but these have different meaning corresponding to the axes being complian or non-compliant.

- **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 security stops. Units are (mm) and (deg).

Test force settings

The on/off button, Teach Test, toggles the behavior of the Teach button on the back of the Teach Pendant from normal teaching mode to testing the force command. When the Teach Test button is on and the Teach 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 Teach button with the other, and notice in which directions the robot can/cannot be moved. Upon leaving this screen, the Teach Test button automatically switches off, which means the Teach button on the back of the Teach Pendant button is again used for free teach mode. Note: The Teach button will only be effectual when a valid feature has been selected for the Force command.

4.4.24 Program → Command Tab, Pallet



A pallet operation can perform a sequence of motions in a set of places given as a pattern, as described in section 4.4.22. 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 are relative to the pallet position. The behavior of a sequence is such that the robot 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"

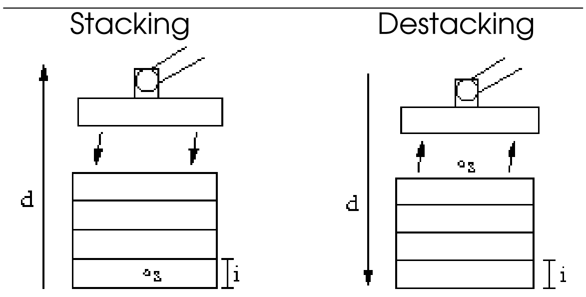
The optional `BeforeStart` sequence is run just before the operation starts. This can be used to wait for ready signals.

"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.

4.4.25 Program → Command Tab, Seek

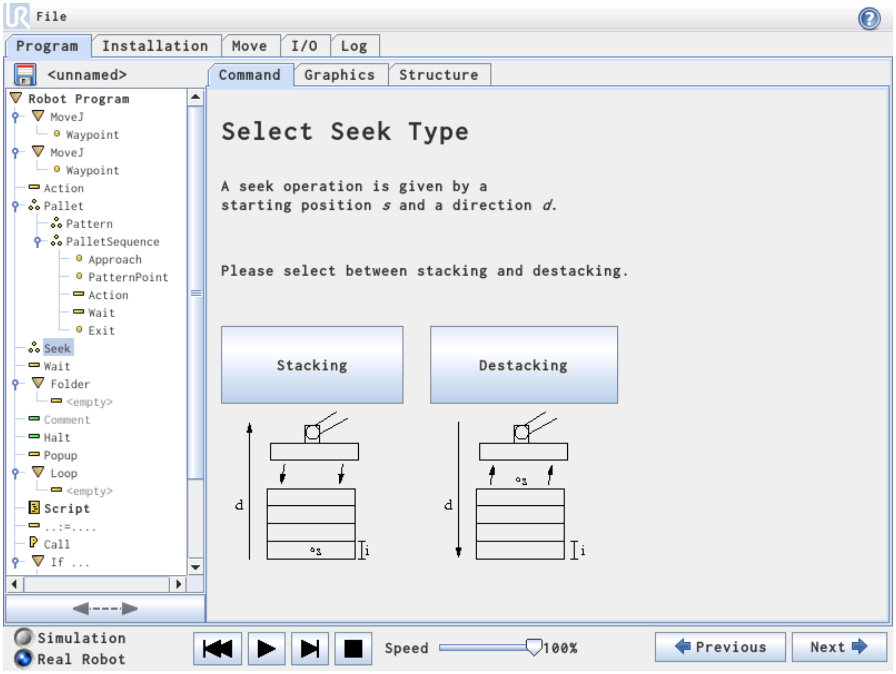
A seek function uses a sensor to determine when the correct position is reached to grab or drop an item. The sensor can be a push button switch, a pressure sensor or a capacitive sensor. This function is made for working on stacks of items with varying item thickness, or where the exact positions of 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.

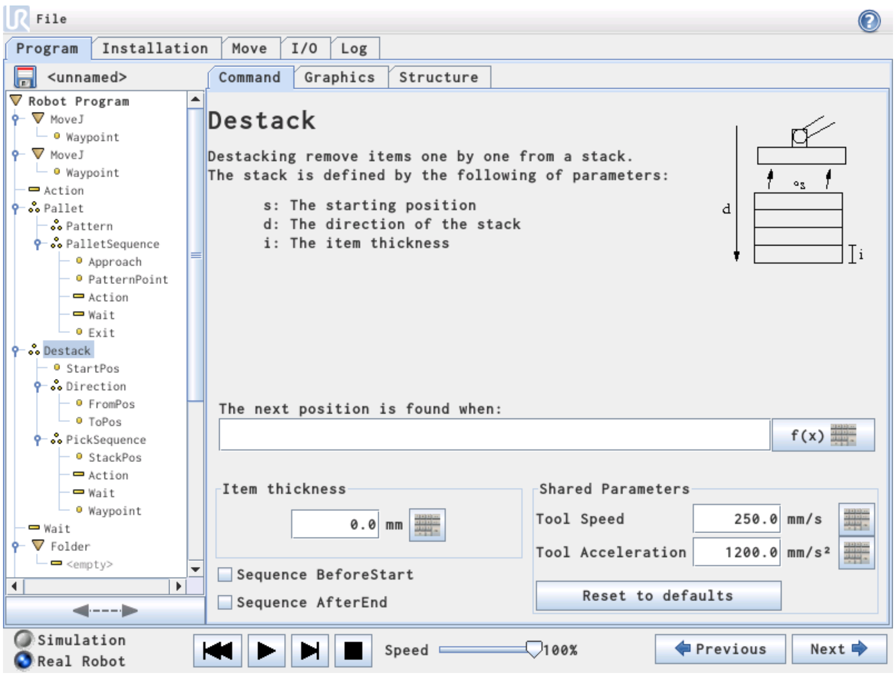
On top of this, one must define the condition for when the next stack position is reached, and a special program sequence that will be performed at each of the stack positions. Also speed and accelerations need to be given for the movement involved in the stack operation.

Stacking



When stacking, the robot 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

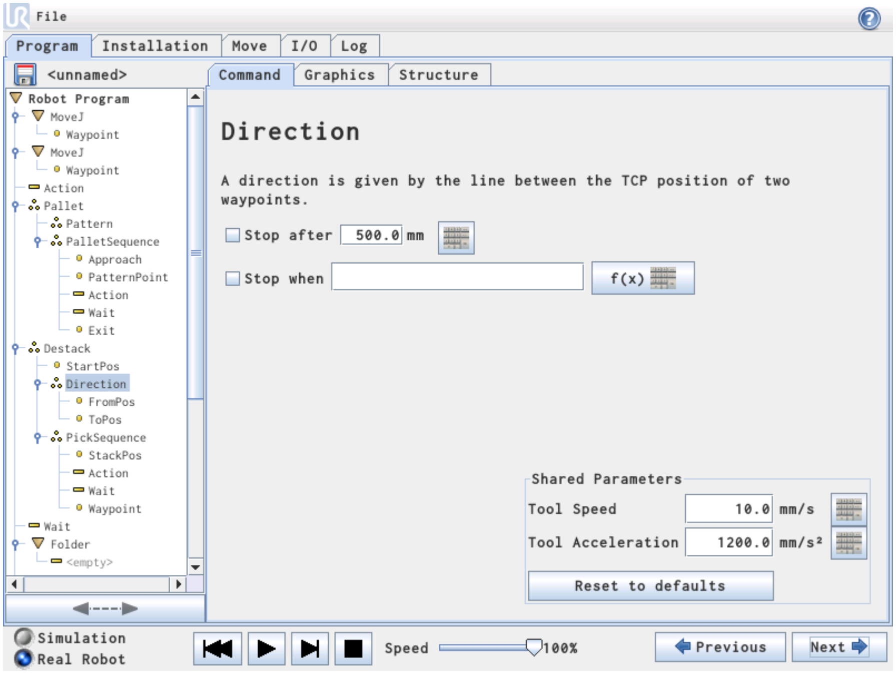


When destacking, the robot moves from the starting position in the given direction to search for the next item. When found, the robot remembers the position and performs the special sequence. The next time round, the robot starts the search from the remembered position, incremented by the item thickness along the direction.

Starting position

The starting position is where the stack operation starts. If the starting position is omitted, the stack starts at the robots current position.

Direction



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 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.

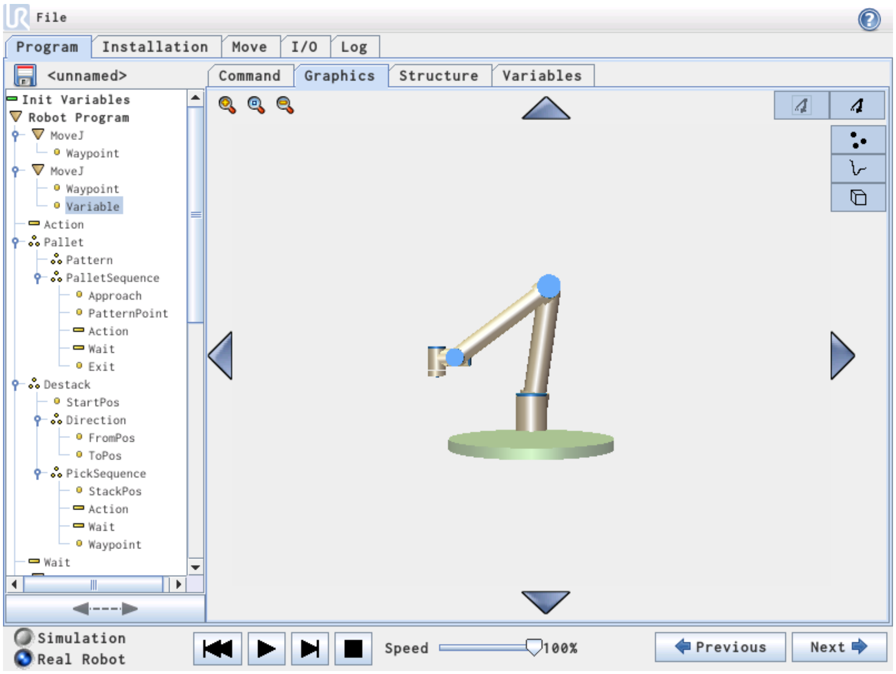
Pick/Place Sequence

Like for the Pallet operation (4.4.24), a special program sequence is performed at each stack position.

4.4.26 Program → Command Tab, 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.

4.4.27 Program → Graphics Tab

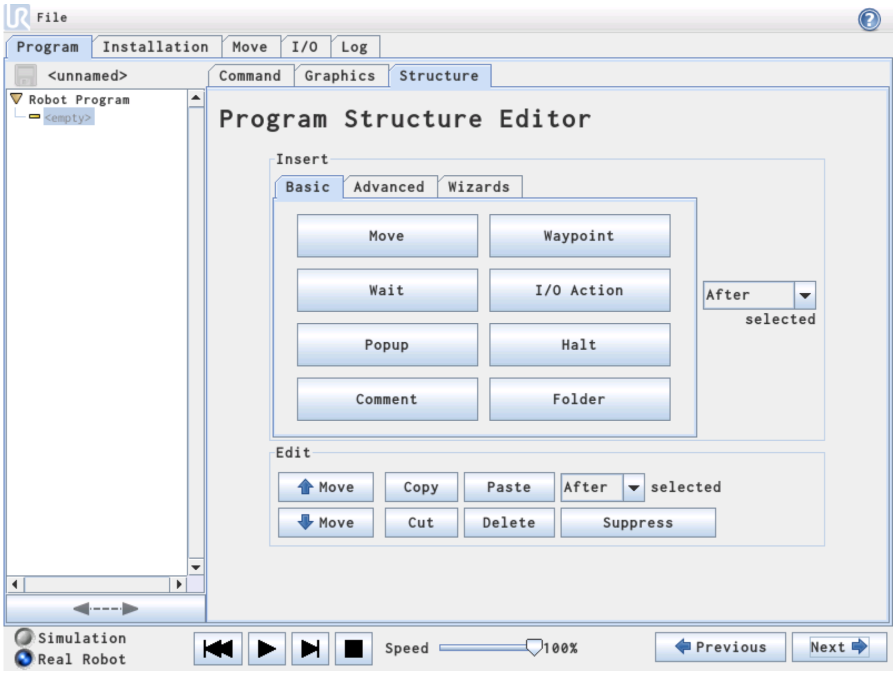


Graphical representation of the current robot program. The path of the TCP is shown in the 3D view, with motion segments in black, and blend segments (transitions between motion segments) shown in green. The green dots specify the positions of the TCP at each of the waypoints in the program. The 3D drawing of the robot shows the current position of the robot, and the "shadow" of the robot shows how the robot intends to reach the waypoint selected in the left hand side of the screen.

The 3D view can be zoomed and rotated to get a better view of the robot. The buttons in the top-right side of the screen can disable the various graphical components in the 3D view.

The motion segments shown depends 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.

4.4.28 Program → Structure Tab



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.

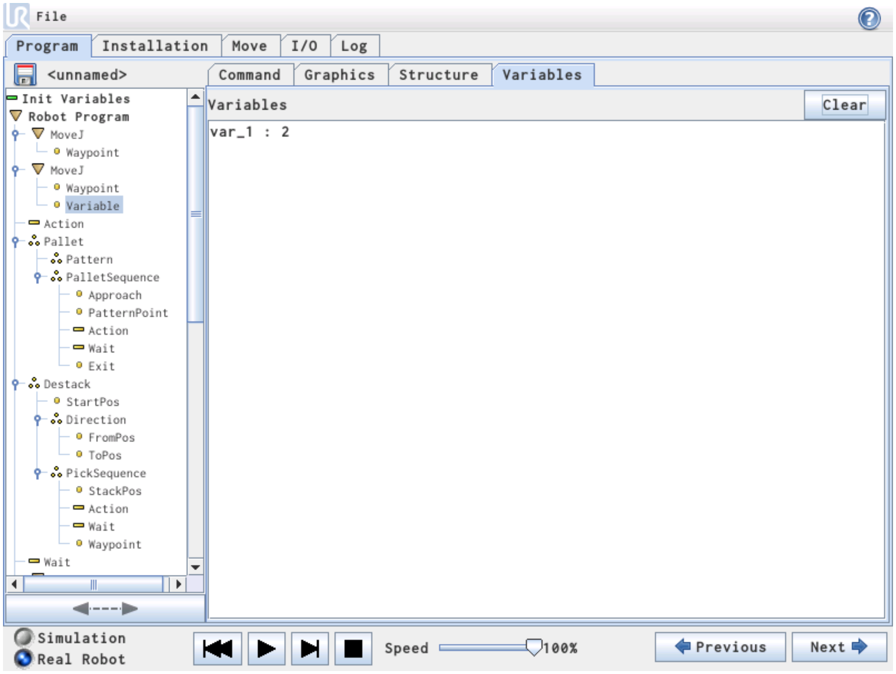
2) Select whether the new command should be inserted above or below the selected command.

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.

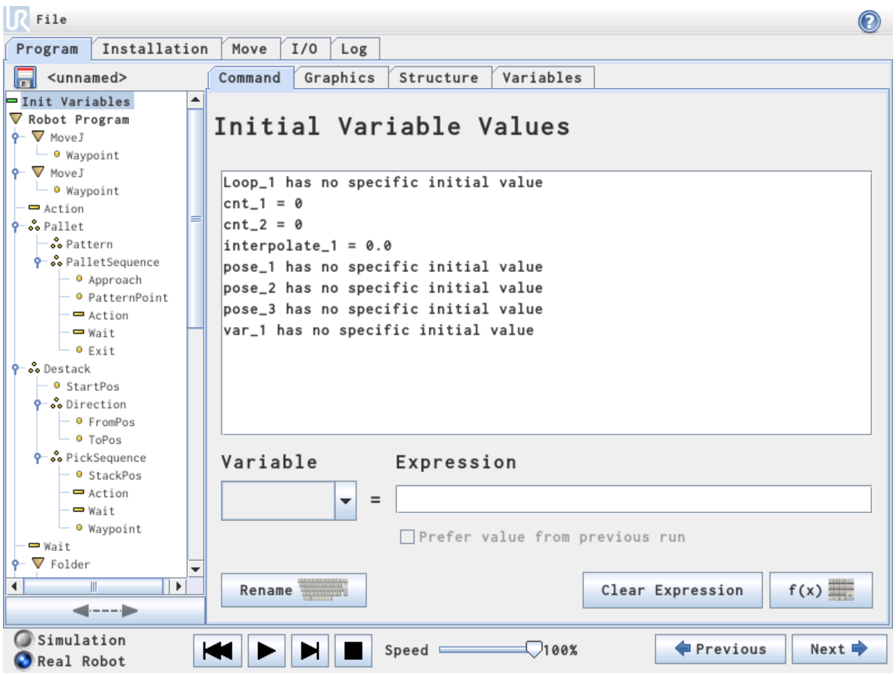
Not all commands fit at all places in a program. `Waypoints` must be under a Move command (not necessarily directly under). `ElseIf` and `Else` commands are required to be after an `If`. In general, moving `ElseIf` commands around can be messy. Variables must be assigned values before being used.

4.4.29 Program → Variables Tab



The Variables tab shows the live values of the variables in the running program, and keeps a list of variables and values between program runs. The variables tab appears only when it has information to display. 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.

4.4.30 Program → Command Tab, Variables Initialization



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

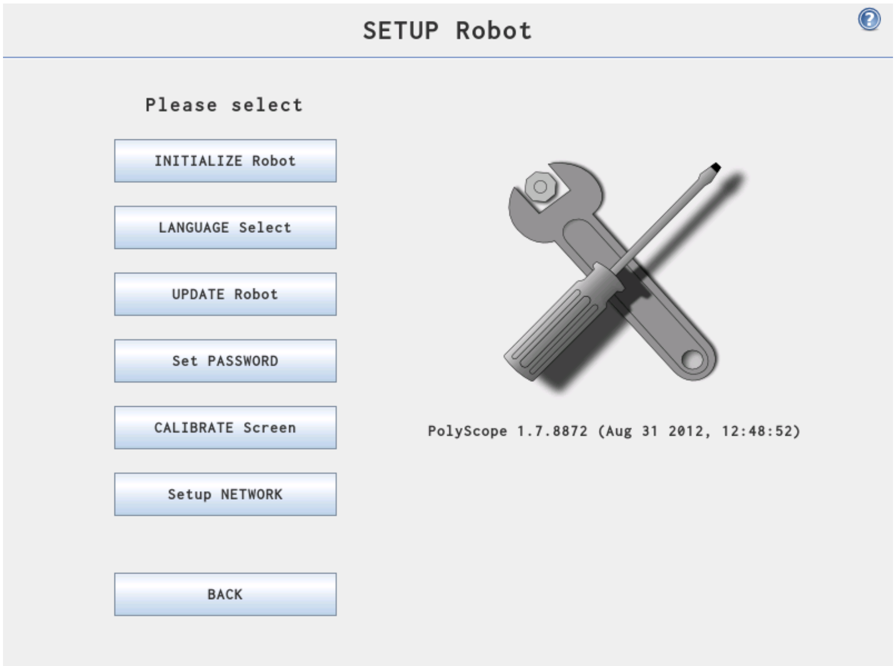
Select a variable from the list of variables by clicking on it, or by using the variable selector box. For a selected variable, an expression can be entered that will be used to set the variable value at program start.

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 section 4.4.29. 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).

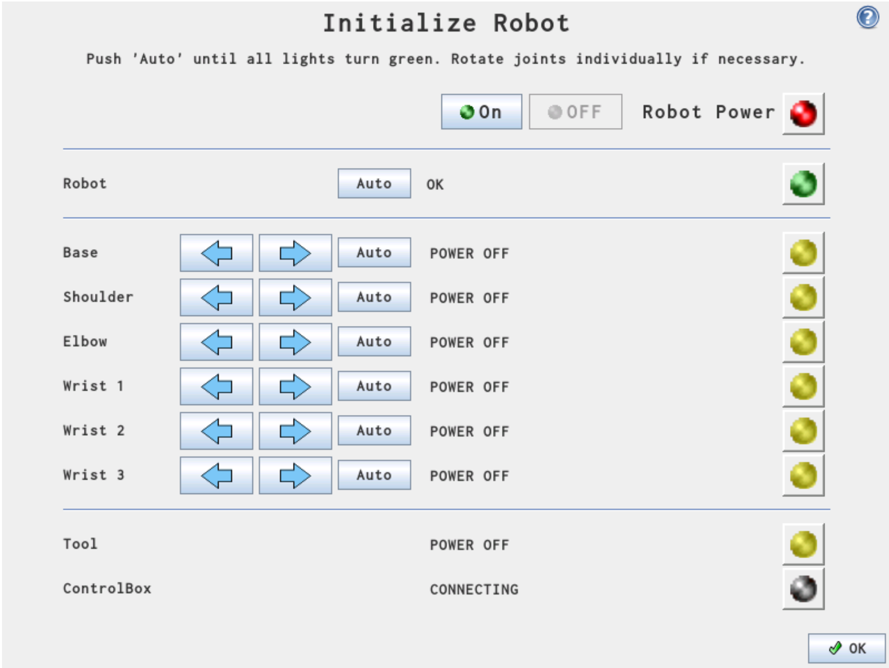
4.5 Setup

4.5.1 Setup Screen



- **Initialize Robot** Goes to the initialization screen, see section 4.5.2.
- **Update** Upgrades the robot software to a newer version via the Internet, see section 4.5.4.
- **Set Password** Provides the facility to lock the programming part of the robot to people without a password, see section 4.5.5.
- **Calibrate Screen** Calibrates the "touch" of the touch screen, see section 4.5.6.
- **Setup Network** Opens the interface for setting up the Ethernet network for the robot, see section 4.5.7.
- **Back** Returns to the Welcome Screen.

4.5.2 Setup Screen → Initialize

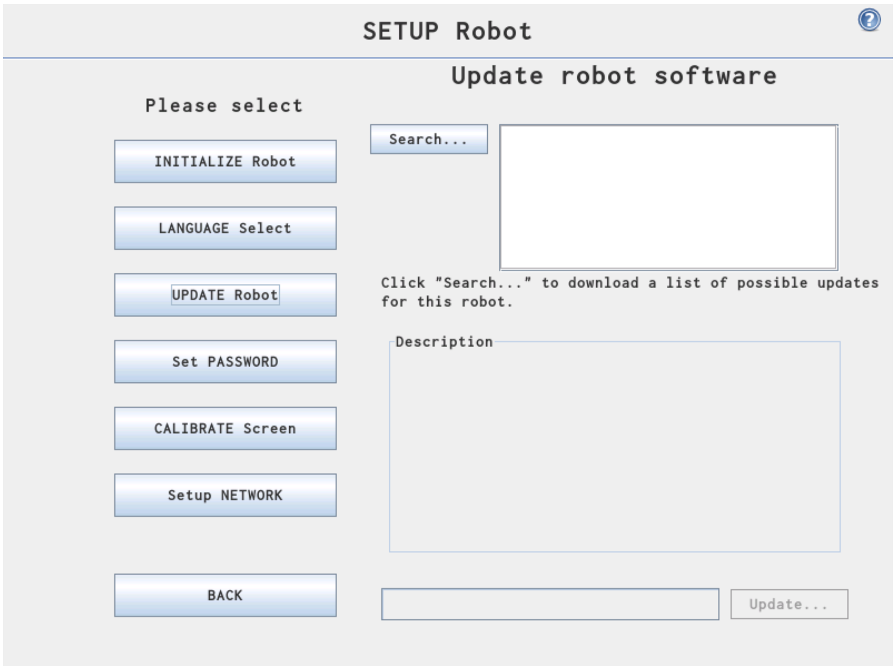


This screen is used when powering up the robot. Before the robot can operate normally, each joint needs to move a little (about 20°) to finds its exact position. The `Auto` button drives all joints until they are `OK`. The joints change drive direction when the button is released and pressed again.

4.5.3 Setup Screen → Language Select

Select the language to be used for the PolyScope software, and for the help function. The GUI needs to restart for changes to take effect.

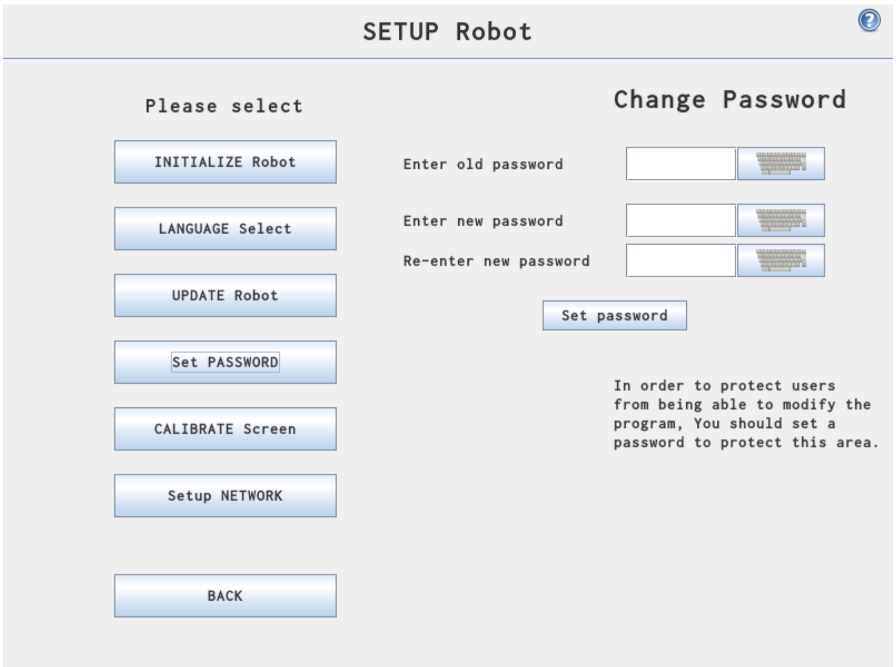
4.5.4 Setup Screen → Update



Provided the robot is attached to the Internet, new software can be downloaded.

WARNING:
1. 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.

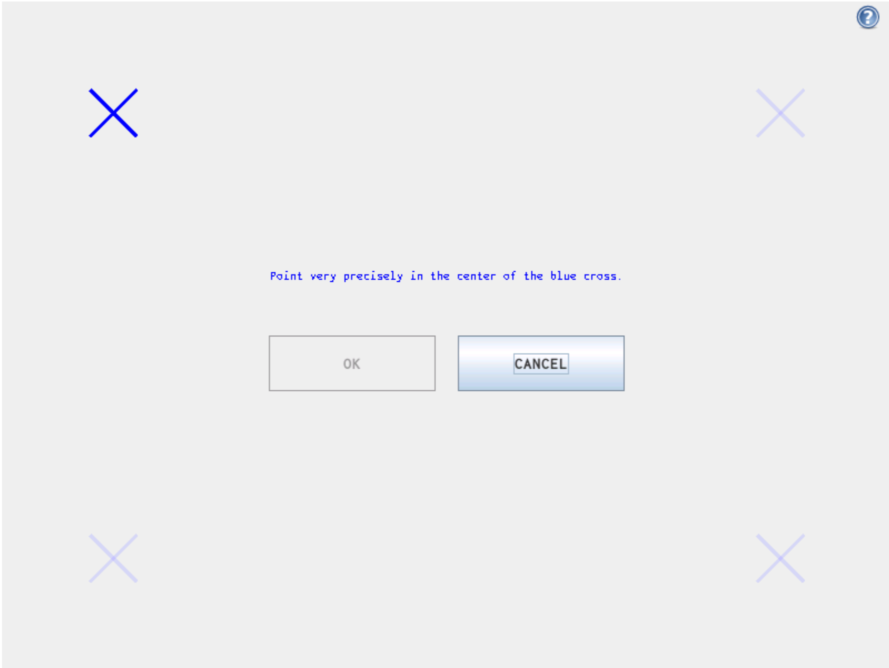
4.5.5 Setup Screen → Password



The programming part of the software can be locked using a password. When locked, programs can be loaded and run without the password, but a password is required to create or change programs.

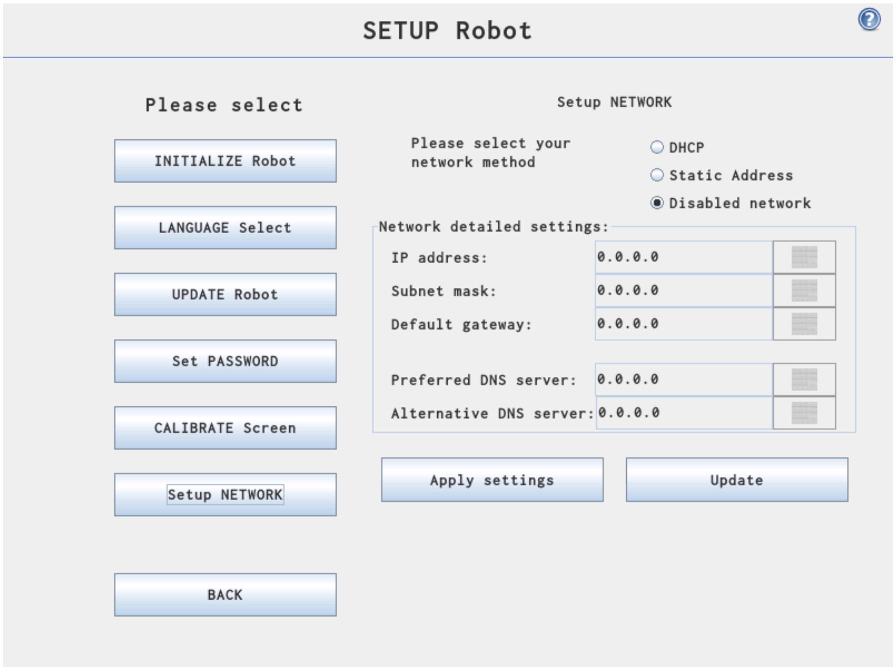
WARNING:
1. Add a password to prevent non-authorized personnel from changing the robot installation.

4.5.6 Setup Screen → Calibrate Touch Screen



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.

4.5.7 Setup Screen → Network



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

Chapter 5

Warranties

5.1 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 cover any consequential damage or loss, such as loss of production or damage to other production equipment.

5.2 Disclaimer

Universal Robots continues to improve reliability and performance of its products, and therefore reserves the right to upgrade 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.

Chapter 6

Applied standards

6.1 List of used standards

Below is a list of applied documents and standards.

Applied EU directives	2006/42/EC Machinery Directive 2004/108/EC EMC Directive 2002/95/EC RoHS Directive 2002/96/EC WEEE Directive
Applied harmonized standards (Under applied EU directives)	ISO 13849-1:2006 ISO 13849-2:2003 ISO 10218-1:2006 (Partly) ISO 10218-1:2011 (Partly) ISO 10218-2:2011 (Partly) ISO 13850:2006 ISO 12100:2010 ISO 3745:2003 IEC 61000-6-2 ED 2.0:2005 IEC 61000-6-4 AMD1 ED 2.0:2010 IEC 61131-2 ED 3.0:2007 (Partly) EN ISO 13849-1:2008 EN ISO 13849-1/AC:2009 EN ISO 13849-2:2008 EN ISO 10218-1:2008 (Partly) EN ISO 10218-1:2011 (Partly) EN ISO 10218-2:2011 (Partly) EN ISO 13850:2008 EN ISO 12100:2010 EN ISO 3745:2009 EN 61000-6-2:2005 EN 61000-6-4/A1:2011 EN 61131-2:2007 (Partly) EN 1037:2010
Applied general standards (Not all standards are listed)	ANSI/RIA R15.06-2012 (Preliminary) ISO 9409-1:2004 (Partly) ISO 9283:1999 (Partly) ISO 9787:2000 (Partly) ISO 9946:2000 (Partly) ISO 8373:1996 (Partly) ISO/TR 14121-2:2007 ISO 1101:2004 ISO 286-1:2010 ISO 286-2:2010 IEC 60664-1 ED 2.0:2007 IEC 60947-5-5:1997 IEC 60529:1989+A1:1999 IEC 60320-1 Ed 2.0:2001 IEC 60204-1 Ed 5.0:2005 (Partly) EN ISO 9409-1:2004 (Partly) EN ISO 9283:1999 (Partly) EN ISO 9787:2000 (Partly) EN ISO 9946:2000 (Partly) EN ISO 8373:1996 (Partly) EN ISO/TR 14121-2:2007 EN ISO 1101:2005 EN ISO 286-1:2010 EN ISO 286-2:2010 EN 60664-1:2007 EN 60947-5-5:1998 EN 60947-5-5/A1:2005 EN 50205:2003 EN 60529:1991+A1:2000 EN 60320:2003 EN 60204:2006 (Partly)

Appendix A

Certifications

DANISH TECHNOLOGICAL INSTITUTE

Teknologiparken
Kongsvang Allé 29
DK-8000 Århus C
Tel. +45 72 20 10 00
Fax +45 72 20 10 19

Universal Robots ApS
Attn.: Lasse Kieffer
Svendborgvej 102
DK-5260 Odense S

16 March 2012
1302213-424098
TGR/BBJ

Test of UR5 Robot

Danish Technological Institute, Centre for Materials Testing has tested a UR5 robot for Universal Robots ApS, see report 1302213-424098.

The test is performed in accordance with the following standards:

- EN ISO 10218-1:2011 5.10, item 5.10.5
- EN ISO 10218-2:2011 5.11, item 5.11.5.5
- EN ISO 10218-1:2006 5.10, item 5.10.5
- ANSI/RIA/ISO 10218-1-2007 5.10, item 5.10.5

The robot is tested in a limited workspace and a force limit of 150N at the tool center point (TCP) of the robot is used during testing. The following results are found during testing:

Test	Max. force [N]	Reason for end of test	Stop code
0	14	Security stop	URcontroller C113A0
1	48	Security stop	URcontroller C113A0
2	68	Security stop	C43A0 Elbow
3	50	Security stop	URcontroller C113A0
4	51	Security stop	URcontroller C113A0
5	16	Security stop	URcontroller C113A0
6	50	Security stop	URcontroller C113A0
7	26	Security stop	URcontroller C113A0
8	28	Security stop	URcontroller C113A0
9	51	Security stop	URcontroller C113A0
10	65	Security stop	URcontroller C113A0
11	63	Security stop	C113A0 Elbow
12	35	Security stop	URcontroller C113A0
13	64	Security stop	URcontroller C113A0
14	23	Security stop	URcontroller C113A0
15	52	Security stop	URcontroller C113A0

The test has verified that the robot is in compliance with the former mentioned items of the standards. All the forces measured during testing are below the chosen force limit of 150N at the TCP.

Yours faithfully
Centre for Materials Testing

Thomas Greve
M.Sc.

Dir. tel.: + 45 7220 2321
E-mail: TGR@teknologisk.dk