USER MANUAL

HEX

Force Torque Sensor

For the Universal Robots

Edition E10

OnRobot FT URCap Plugin Version 3.1.3

June 2018

Content

Information provided within this document is accurate to the best of our knowledge at the time of its publication. There may be differences between this document and the product if the product has been modified after the edition date.

OnRobot A/S. does not assume any responsibility for any errors or omissions in this document. In no event shall OnRobot A/S. be liable for losses or damages to persons or property arising from the use of this document.

The information within this document is subject to change without notice. You can find the latest version on our webpage at: **https://onrobot.com/**.

The original language for this publication is English. Any other languages that are supplied have been translated from English.

All trademarks belong to their respective owners. The indications of (R) and TM are omitted.

1 Preface

1.1 Target Audience

This document is intended for integrators who design and install complete robot applications. Personnel working with the sensor are expected to have the following expertise:

1. Basic knowledge of mechanical systems
2. Basic knowledge of electronic and electrical systems
3. Basic knowledge of the robot system

1.2 Intended Use

The sensor is designed for measuring forces and torques, installed on the end effector of a robot. The sensor can be used within the specified measurement range. Using the sensor outside of its range is considered misuse. OnRobot is not liable for any damage or injury resulting from misuse.

1.3 Important safety notice

The sensor is *partly completed machinery* and a risk assessment is required for each application the sensor is a part of. It is important that all safety instructions herein are followed. The safety instructions are limited to the sensor only and do not cover the safety precautions of a complete application.

The complete application must be designed and installed, in accordance with the safety requirements specified in the standards and regulations of the country where the application is installed.

1.4 Warning Symbols

DANGER:

This indicates a very dangerous situation which, if not avoided, could result in injury or death.

WARNING:

This indicates a potentially hazardous electrical situation which, if not avoided, could result in injury or 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.

CAUTION:

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

NOTE:

This indicates additional information such as tips or recommendations.

1.5 Typographic Conventions

The following typographic conventions are used in this document.

Table 1: Conventions

`Courier Text`	File paths and file names, code, user input and computer output.
Italicized text	Citations and marking image callouts in text.
Bold text	UI elements, including text appearing on buttons and menu options.
Bold, blue text	External links, or internal cross-references.
<angle brackets>	Variable names that must be substituted by real values or strings.
1. Numbered lists	Steps of a procedure.
A. Alphabetical lists	Image callout descriptions.

2 Getting Started

2.1 Scope of Delivery

In the Universal Robots OnRobot HEX Sensor Kit everything is provided that is required to connect the OnRobot force/torque sensor to your UR robot.

There are two versions of the OnRobot Universal Robots (UR) Kit, depending on the HW version of the sensor.

2.1.1 OnRobot (OptoForce) UR Kit (v1)

The contents of the OnRobot (OptoForce) UR Kit v1 are the following:

- OnRobot (OptoForce) 6-axis force/torque sensor (variant HEX-E v1 or HEX-H v1)
- OnRobot (OptoForce) Compute Box
- OnRobot (OptoForce) USB drive
- adapter-A
- overload plug
- sensor cable (4 pin M8 - 4 pin M8, 5 m)
- Compute Box power cable (3 pin M8 – open ended)
- Compute Box power supply
- UTP cable (RJ45 - RJ45)
- USB cable (Mini-B – Type A)
- PG16 cable gland
- plastic bag, containing:

4. cable holder
5. M6x30 screws (2 pcs)
6. M6x8 screws (10 pcs)
7. M5x8 screws (9 pcs)
8. M4x8 screws (7 pcs)
9. M4x12 screws (2 pcs)
10. M4 washer (8 pcs)

2.1.2 OnRobot UR Kit (v2)

The contents of the OnRobot UR Kit v2 are the following:

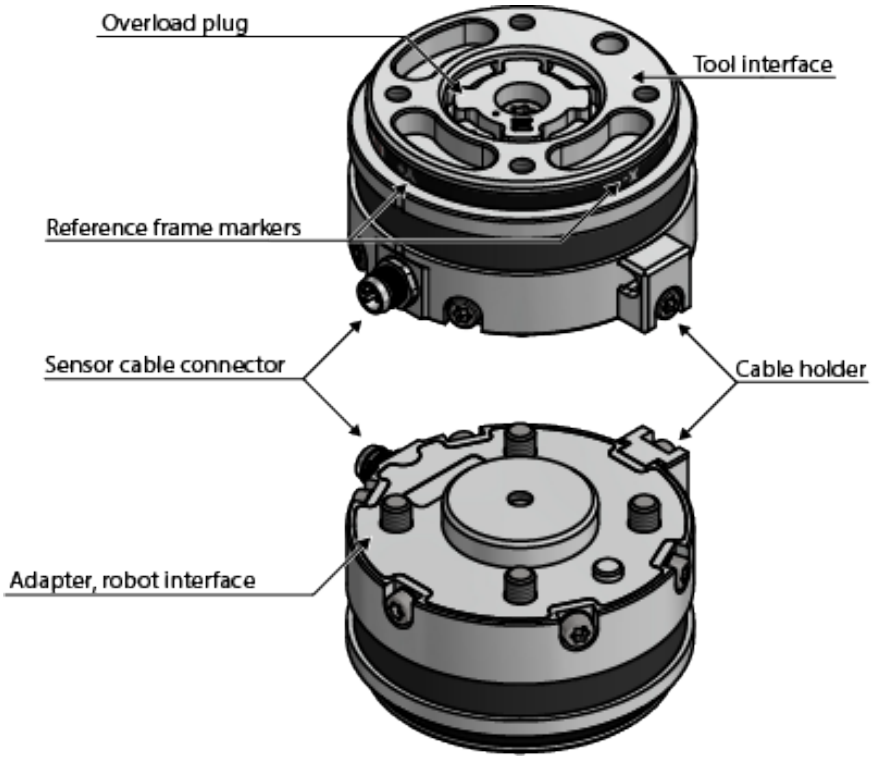
11. OnRobot 6-axis force/torque sensor (variant HEX-E v2 or HEX-H v2)
12. OnRobot Compute Box
13. OnRobot USB drive
14. adapter-A2

15. sensor cable (4 pin M8 - 4 pin M8, 5 m)
16. Compute Box power cable (3 pin M8 – open ended)
17. Compute Box power supply
18. UTP cable (RJ45 - RJ45)
19. USB cable (Mini-B – Type A)
20. PG16 cable gland
21. plastic bag, containing:
22. cable holder, with integrated screw
23. M6x8 Torx screws (10 pcs)
24. M5x8 Torx screws (9 pcs)
25. M4x6 Torx screws (7 pcs)
26. M6 washer (10 pcs)
27. M5 washer (9 pcs)

2.2 Sensor Description

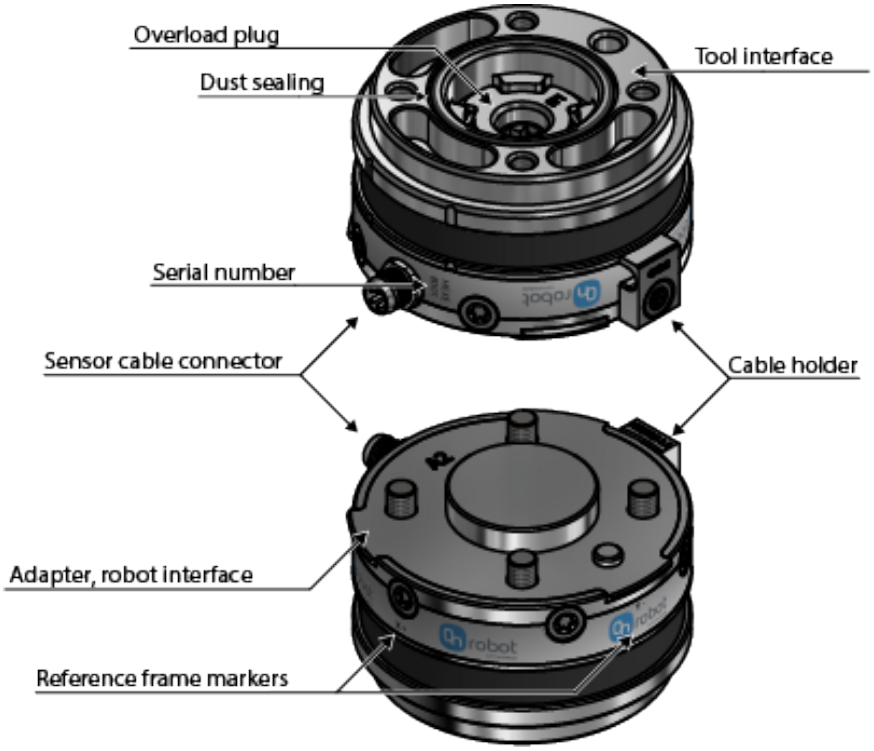
2.2.1 HEX-E v1 and HEX-H v1

The sensor consists of a sensor body, an adapter, and a overload plug. The sensor cable connector, the cable holder, and markers for the reference frame are on the sensor body. The tool is fastened to the sensor body directly, onto the tool interface. The sensor is fastened to the robot tool flange by the adapter.



2.2.2 HEX-E v2 and HEX-H v2

The sensor consists of a sensor body, an adapter, and a overload plug. The sensor cable connector, the cable holder, dust sealing, serial number and markers for the reference frame are on the sensor body. The tool is fastened to the sensor body directly, onto the tool interface. The sensor is fastened to the robot tool flange by the adapter.



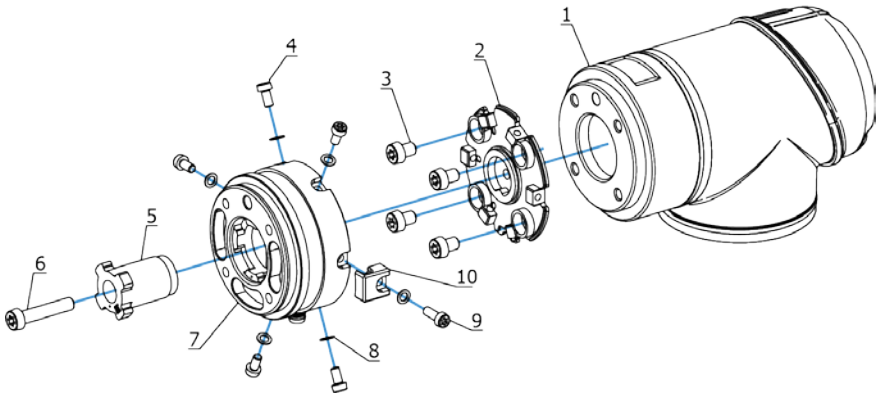
2.3 Mounting

Use only the screws provided with the sensor. Longer screws could damage the sensor or the robot.

2.3.1 HEX-E v1 and HEX-H v1

To mount the sensor, follow this process:

1. Fasten Adapter-A to the Robot by four M6x8 Screws. Use 6 Nm tightening torque.
2. Fasten the Sensor to the adapter by five M4x8 screws with M4 washers. Use 1,5 Nm tightening torque.
3. Fasten the cable to the Sensor with the Cable Holder by one M4x12 screw and M4 washer. Use 1,5 Nm tightening torque.
4. Fasten the Plug to the Sensor by one M6x30 Screw. Use 6 Nm tightening torque.



Legend: 1 –robot tool flange, 2 – Adapter A, 3 - M6x8 screws, 4 – M4x8 screws, 5 – overload plug, 6 – M6x30 screw, 7 – sensor, 8 – M4 washer, 9 – M4x12 screw, 10 – cable holder

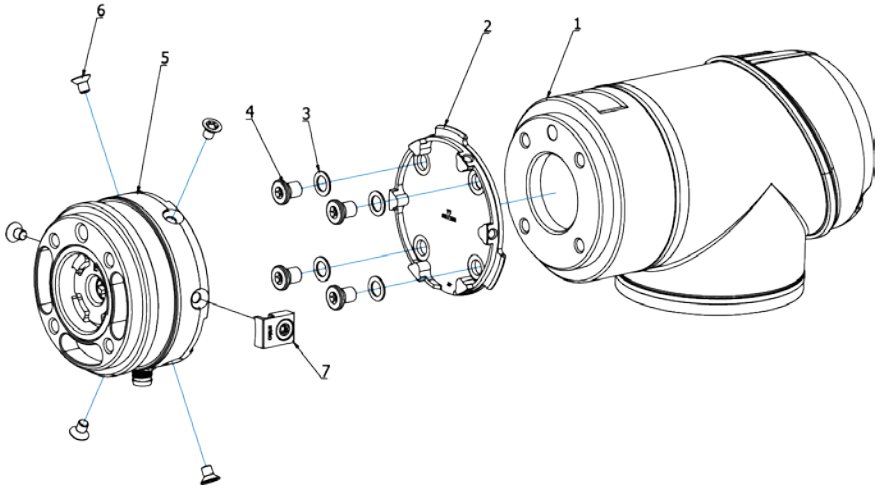
5. Fasten the tool to the Sensor, according to the instructions from the tool manufacturer.

> The overload protection is not fully functional, if the tool is not joined to the Sensor with a flat surface.

2.3.2 HEX-E v2 and HEX-H v2

To mount the sensor, follow this process:

1. Fasten Adapter-A2 to the Robot by four M6x8 Torx Screws with M6 washers. Use 6 Nm tightening torque.
2. Fasten the Sensor to the adapter by five M4x86 screws with M4 washers. Use 1,5 Nm tightening torque.
3. Fasten the cable to the Sensor with the Cable Holder by one M4x12 screw and M4 washer. Use 1,5 Nm tightening torque.
4. Fasten the Plug to the Sensor by one M6x30 Screw. Use 6 Nm tightening torque.



Legend: 1 –robot tool flange, 2 – Aadapter- A2, 3 - M6 washer, 4 – M6x8 Torx screws, 4, 5 – sensor, 6 – M4x6 Torx screws, 7 – cable holder

5. Fasten the tool to the Sensor, according to the instructions from the tool manufacturer.

> The overload protection is not fully functional, if the tool is not joined to the Sensor with an interface described in ISO 9409-1-50-4-.

2.4 Cable Connections

To connect the sensor, follow this process:

1. Connect the 4 pin M8 cable (5m long) to the sensor. Ensure that the holes of the cable are aligned with the pins of the connector on the sensor.

> Do not rotate the cable, only rotate the connector lock.

2. Secure the cable to the robot with cable ties.

> Make sure that enough extra cable length is available around the joints for bending.

3. Place the Compute Box somewhere near or inside the UR robot control cabinet and connect the 4 pin M8 sensor cable. The provided cable gland can be used to lead the cable into the UR Control cabinet.
4. Connect the Compute Box's Ethernet interface with the UR controller's Ethernet interface via the supplied UTP cable.
5. Use the 3 pin M8 cable (1m long) to power the Compute Box from the UR's control box. Connect the brown cable to the 24V and the black cable to the 0V.

Power	
PWR	■
GND	■
24V	■
0V	■

Configurable Inputs			
24V	■	24V	■
CI0	■	CI4	■
24V	■	24V	■
CI1	■	CI5	■
24V	■	24V	■
CI2	■	CI6	■
24V	■	24V	■
CI3	■	CI7	■

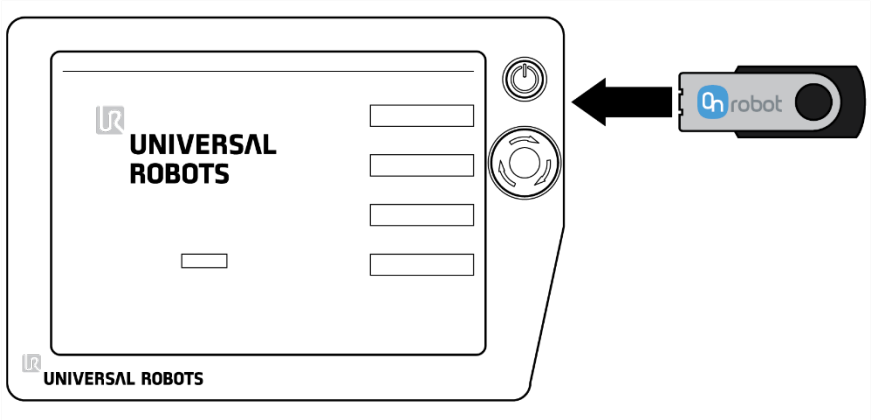
Configurable Outputs			
0V	■	OV	■
CO0	■	CO4	■
0V	■	0V	■
CO1	■	CO5	■
0V	■	0V	■
CO2	■	CO6	■
0V	■	0V	■
CO3	■	CO7	■

For more information, refer to the UR's documentation.

6. Apply the correct network settings to both the Compute Box and the UR robot. The default Compute Box IP address is 192.168.1.1, to change it, see **Changing the IP of the Compute Box**.

2.5 Software Upload to Robot Controller

To upload the OnRobot examples and URCap plugin, insert the USB drive in the USB slot on the right side of the Teach Pendant.



1. A red "!USB!" warning sign appears, indicating the upload is in progress.
2. Wait for a green "USB" sign to appear, indicating the upload is completed.
3. Remove USB drive. The example codes are now in the program folder of the UR robot.

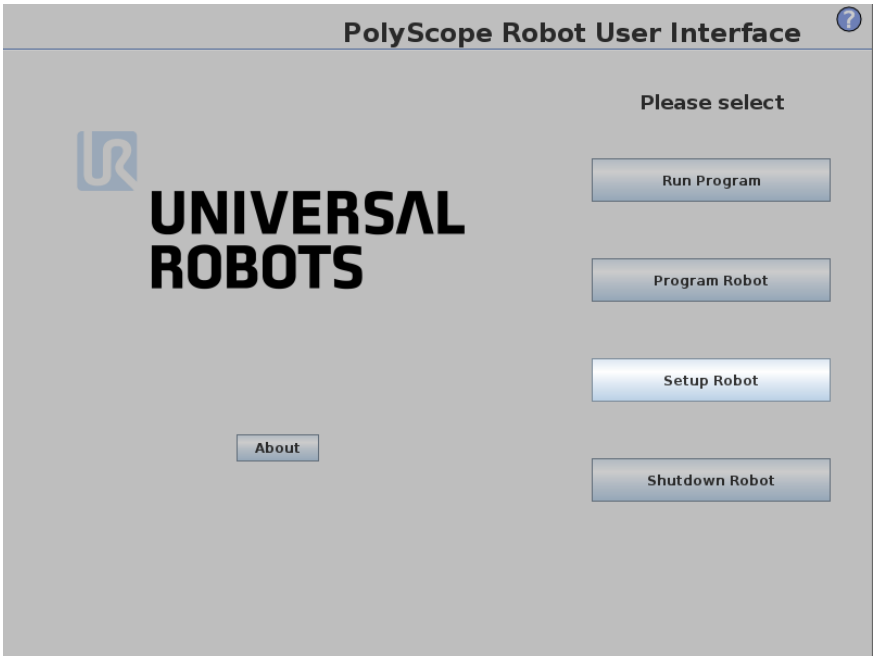
The URCap plugin and a few examples are automatically copied to a new `programs/OnRobot_UR_Programs` folder.

Continue with **URCap Plugin Installation**.

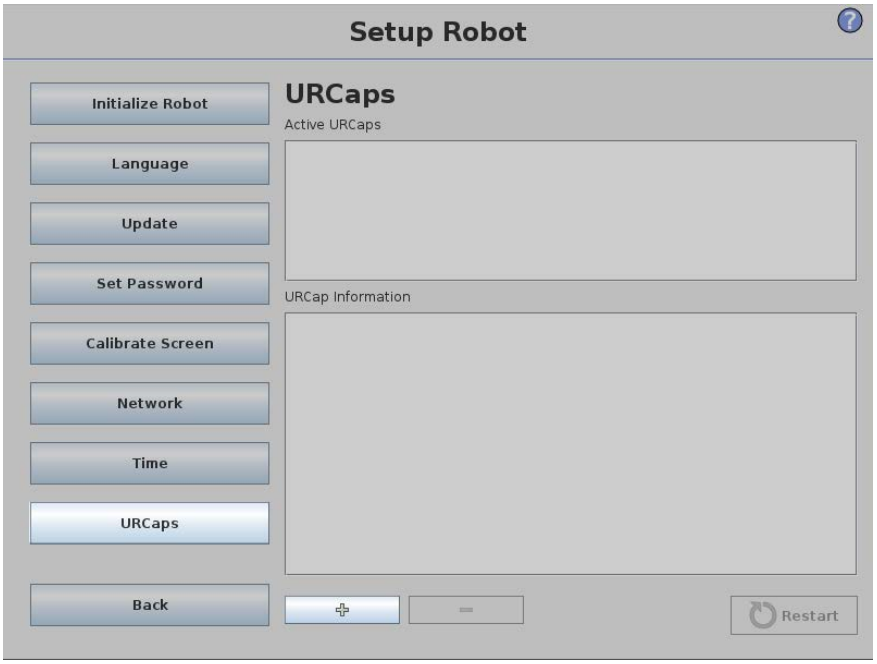
2.6 URCap Plugin Installation

Ensure that the robot controller has PolyScope version 3.5 or higher.

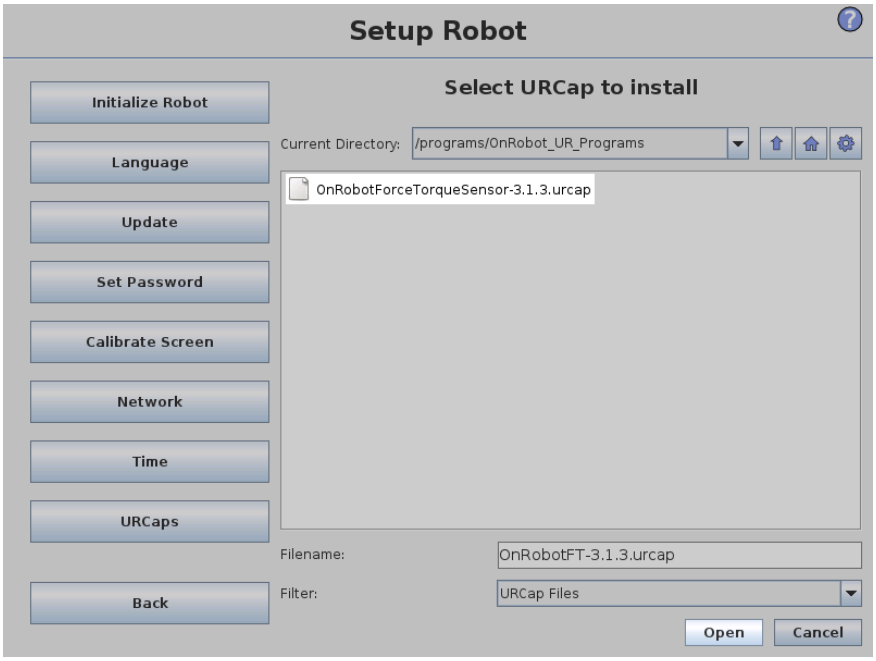
1. Select the **Setup Robot** option from the main menu.



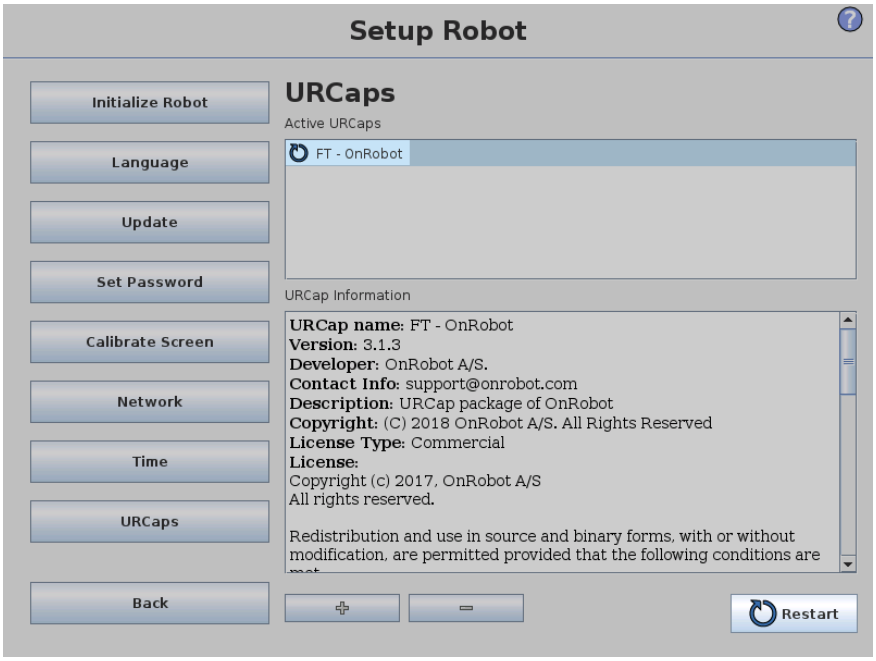
2. Press the URCaps Setup button.
3. Press the + sign.



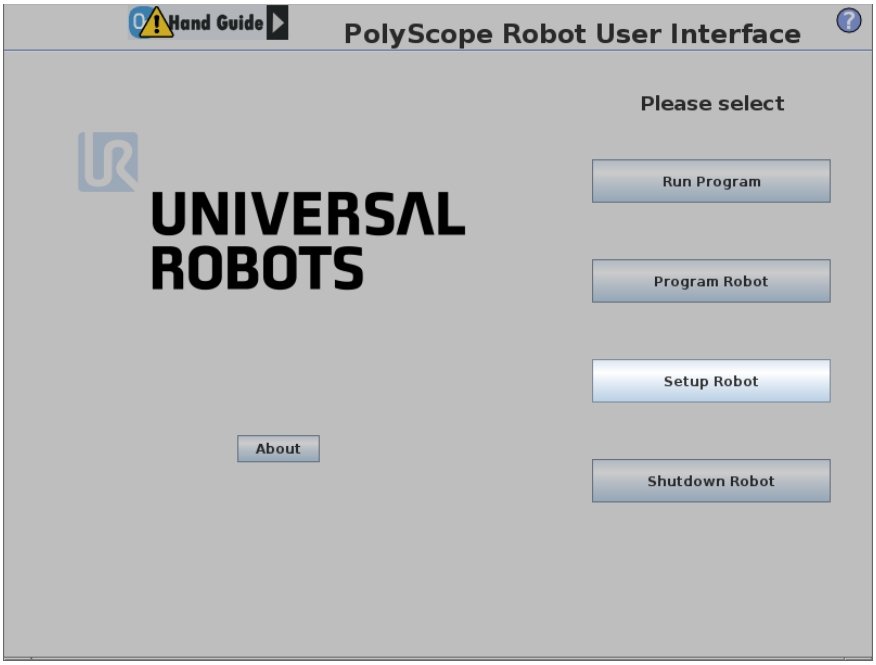
4. Select the OnRobotForceTorqueSensor-x.y.z.urcap file from the `programs/OnRobot_UR_Programs` folder.
5. Press the Open button.



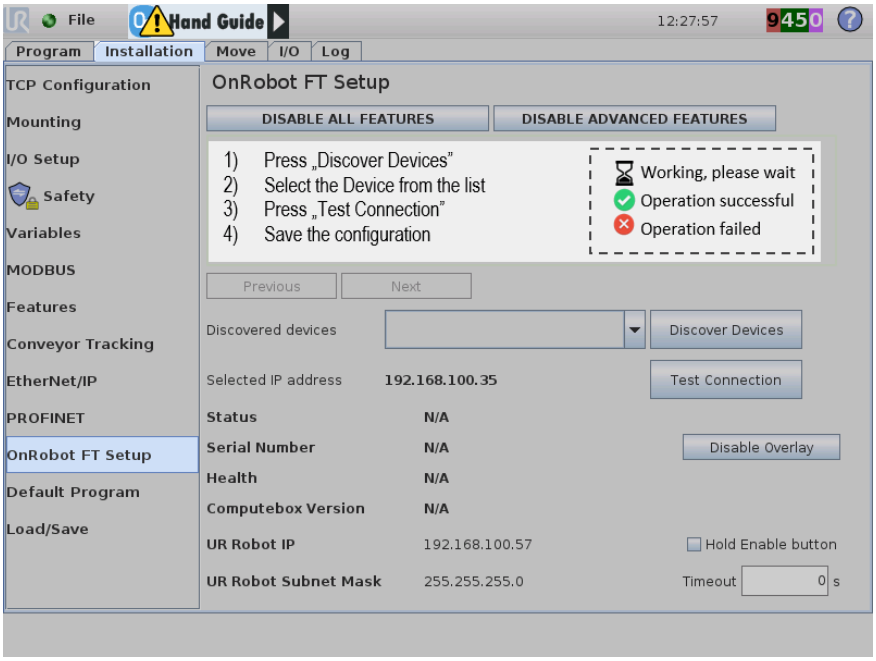
6. The screen shows that the OnRobot Force Torque Sensor URCap plugin has been installed.
7. Press the Restart button to finalize installation process.



8. Click on the Program Robot button after the restart is finished.



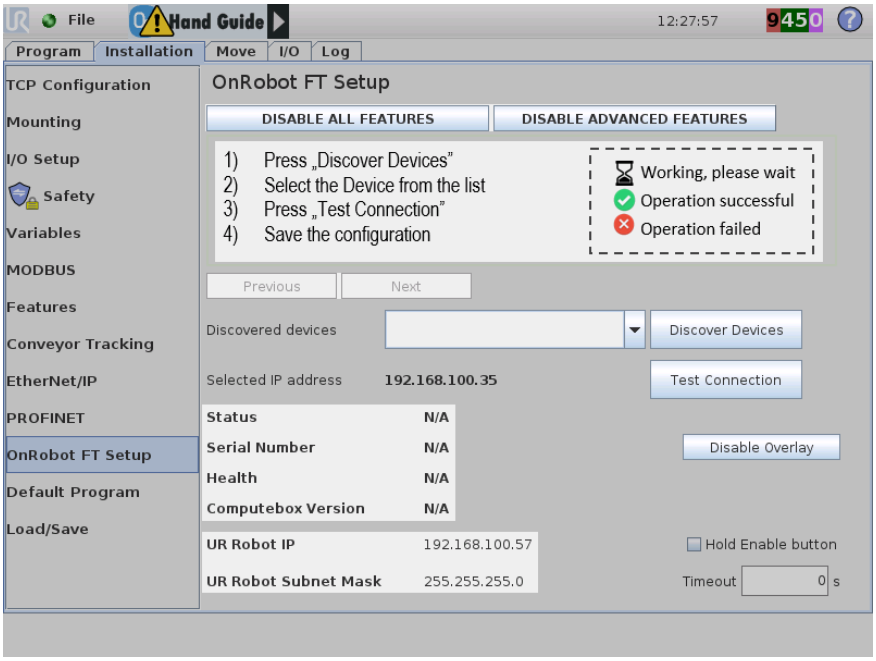
9. Select the Installation tab.
10. Select OnRobot FT Setup.
11. The welcome screen of the newly installed OnRobot Force Torque Sensor URCap plugin is shown.
12. The OnRobot Hand Guide Toolbar is now also visible.



Continue with **URCap Plugin Setup**.

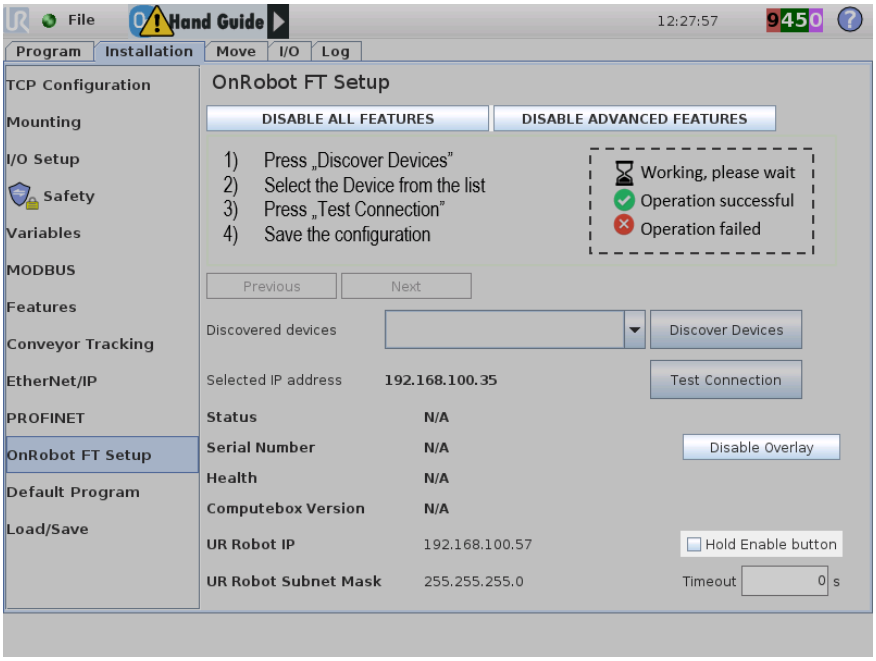
2.7 URCap Plugin Setup

Select the **Installation** tab, then select **OnRobot Setup**. The following screen is shown:



The following information is shown on this screen:

- Information about the steps needed for the sensor setup.
- Navigation buttons to move between pages of the guide.
- A list of discovered sensors which can be reached with the current network configuration. Make sure that the correct sensor is selected by checking the sensor IP and serial number.
- Connected sensors can be discovered by pressing the **Discover Devices** button.
- The selected OnRobot device must be tested first.
- General information about the tested device.
- IP and Subnet mask of the currently used UR robot.
- Enable/Disable buttons for the OnRobot Hand Guide Toolbar.



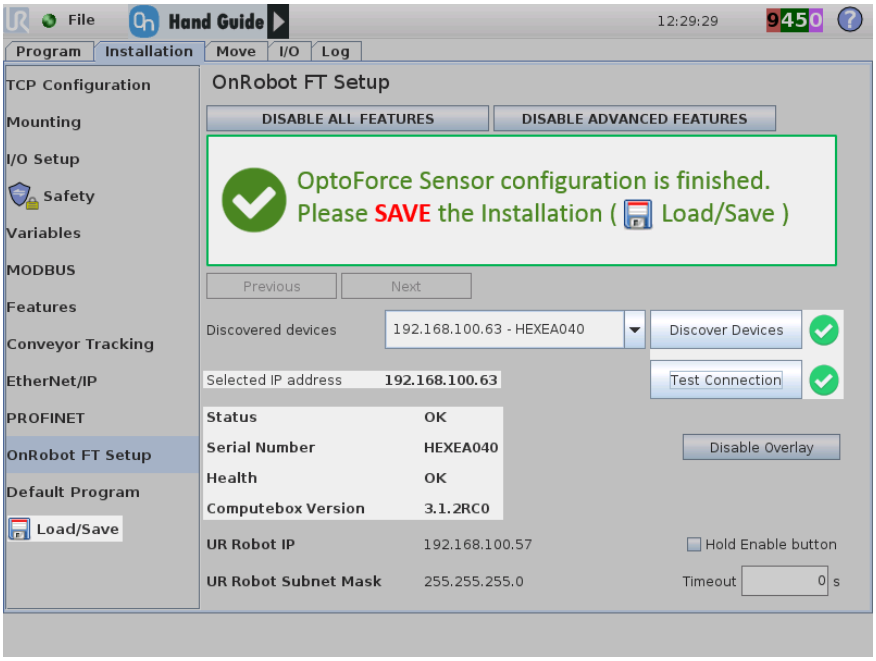
The OnRobot Force Torque Sensor URCap plugin must be enabled (as the screen above is shown). If the plugin is not enabled, the button shown under *mark A* has the text **ENABLE ALL FEATURES**. Press the button, then start the toolbar with the **Enable Overlay** button (*mark B*).

Here the Hand Guide behavior can also be set (mark C). If the **Hold Enable button** checkbox is checked, the **Enable** button on the Teach Pendant functions as a live-man switch, that is, the **Enable** button has to be pressed continuously for the Hand Guide to function. If the **Hold Enable button** checkbox is unchecked, the **Enable** button acts as a normal switch, that turns the Hand Guide function on and off. By default, the **Hold Enable button** checkbox is checked.

2.7.1 Example Setup

1. Press the **Discover Devices** button.
2. In the **Discovered Devices** drop-down list, select the Compute Box.
3. Press the **Test Connection** button.

The following screen is shown:



The following information is shown on this screen:

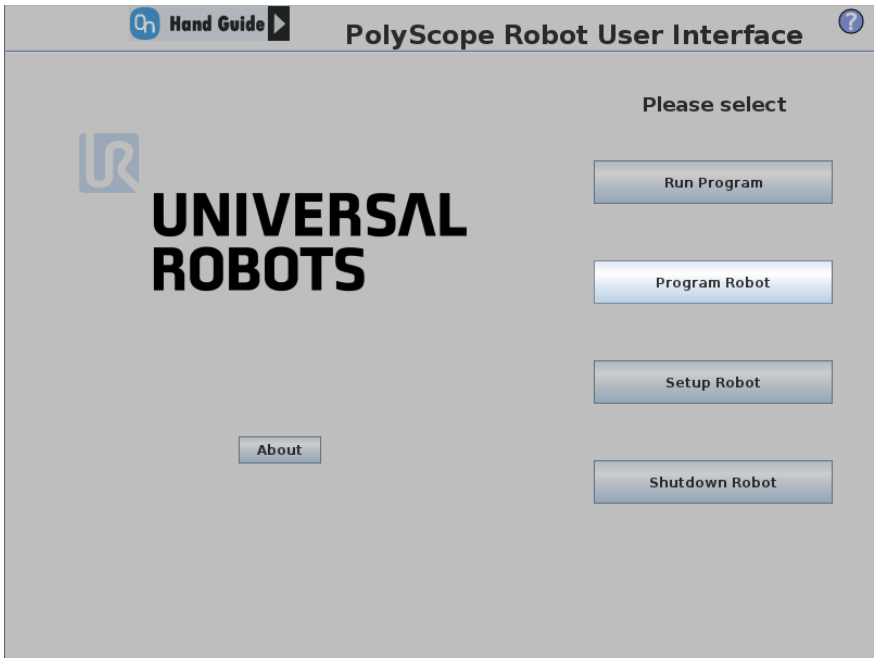
- A message indicating that the OnRobot commands can be used in your own UR programs.
- A drop-down list with the pairs of the Compute Box IP addresses and the Sensor Serial Numbers available in the network.
- In this example, the Device with serial number HEXEA040 was selected as the IP address is matched to HEXEA040.
- After pressing the Test Connection button, basic information is shown about the device:
- The status of overall process
- The serial number of the sensor
- The Health String of the sensor
- The SW version of the Compute Box
- The green check marks indicate that both the Discover Devices and Test Connection procedures were successful.
- You can save the current configuration by pressing the Load/Save button.

3 Using the URCap Plugin

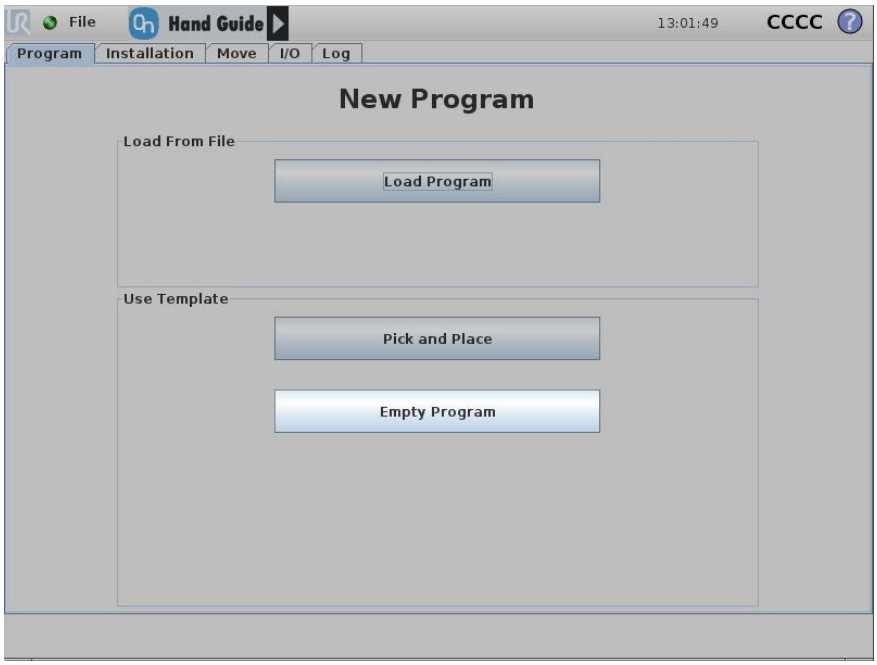
3.1 OnRobot Feedback Variables

Simple features are demonstrated in this section through an example program. The program shows how to gain data from the OnRobot sensor and how to zero the Force/Torque values of the sensor.

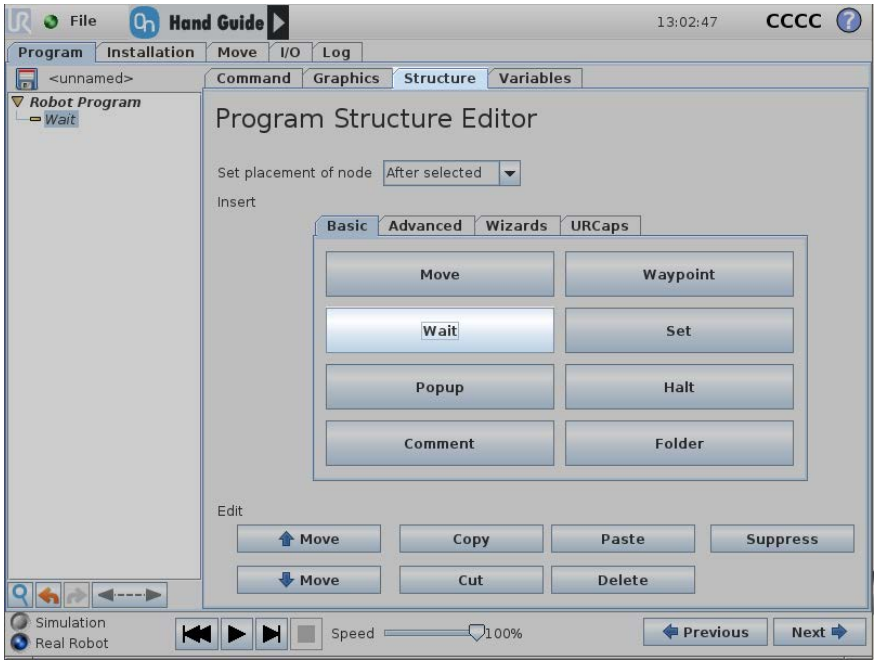
1. Click on Program Robot.



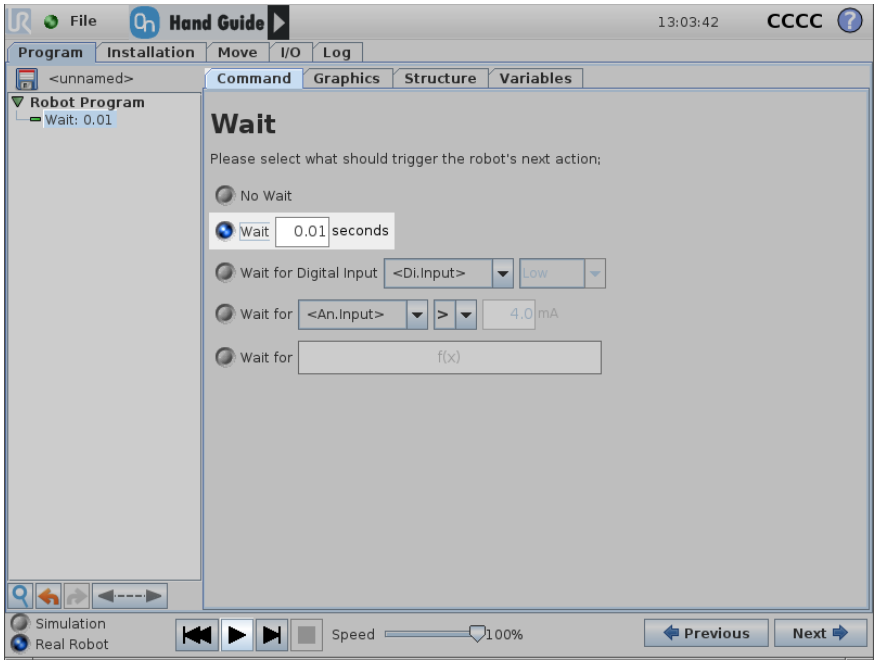
2. Click on Empty Program.



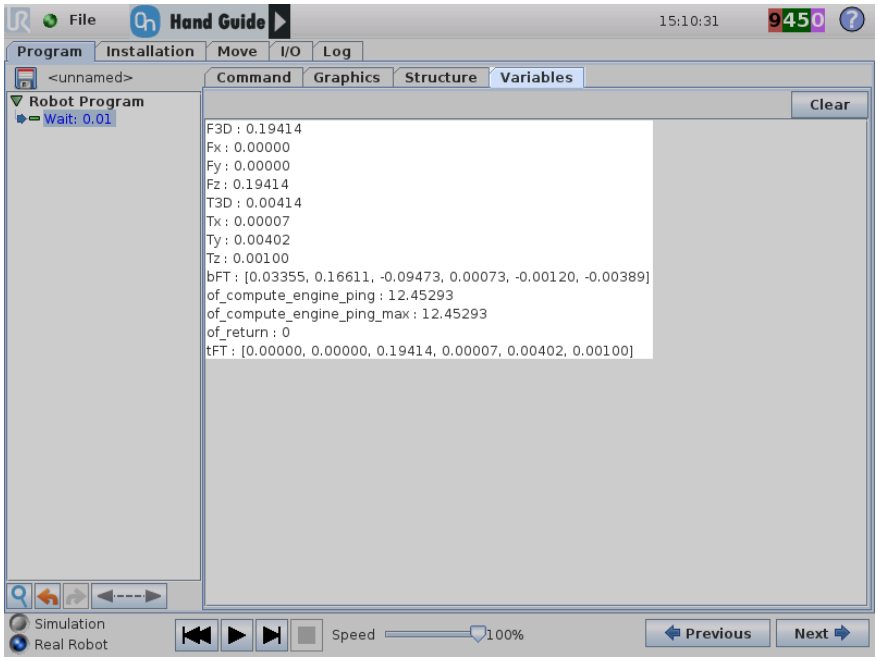
3. Select the **Structure** tab.
4. Press the **Wait** button to avoid infinite loop in the program.



5. Select the **Wait** command in the program structure.
6. Select the **Command** tab.
7. Set the **Wait** to 0.01 seconds.
8. Press the Play button to execute program.



9. Select the **Variables** tab.



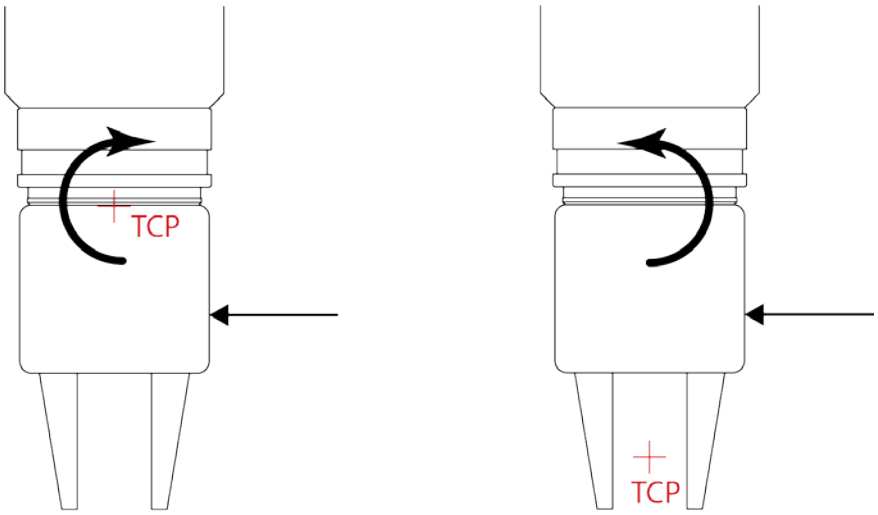
The Force values and Torque values are visible. You can use these variables in any program.

These variables are updated automatically at a rate of approximately 125Hz:

- **F3D**: Length of the 3D force vector F3D = sqrt (Fx^2 +Fy^2 +Fz^2) (N)
- **Fx**: Force vector in the X direction in Newton (N)
- **Fy**: Force vector in the Y direction in Newton (N)
- **Fz**: Force vector in the Z direction in Newton (N)
- **T3D**: Length of the 3D torque vector T3D = sqrt (Tx^2 +Ty^2 +Tz^2) (Nm)
- **Tx**: Torque in the X direction in Newton Meter (Nm)
- **Ty**: Torque in the Y direction in Newton Meter (Nm)
- **Tz**: Torque in the Z direction in Newton Meter (Nm)
- **bFT**: Force and torque values calculated in the Base Coordinate system, in an array in Newton (N) and Newton Meter (Nm)
- **of_compute_engine_ping**: the actual ping of the Compute Box Robot Controller communication
- **of_compute_engine_ping_max**: the highest ping of the Compute Box Robot Controller communication, since the last program start
- **of_return**: the variable used to store the outcome of OnRobot commands
- **tFT**: Force and torque values calculated in the Tool Coordinate system, in an array in Newton (N) and Newton Meter (Nm)

3.1.1 Effects of the TCP Position

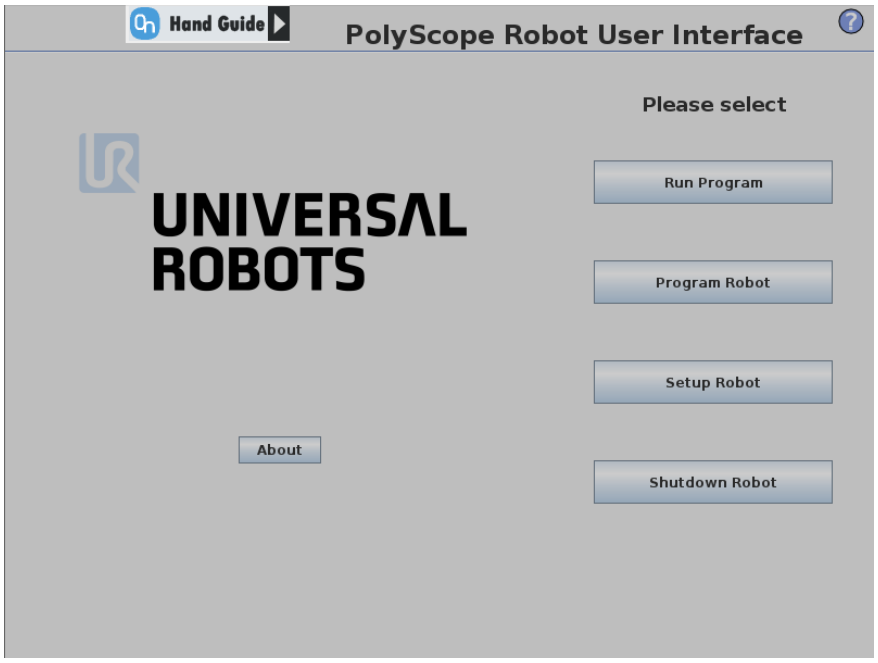
The torques are calculated based on the Tool Center Point, meaning the torque exerted by the measured forces is calculated in the Tool Center Point, not on the sensor face. See the effects of the TCP placement on the measured torque in the figure below.

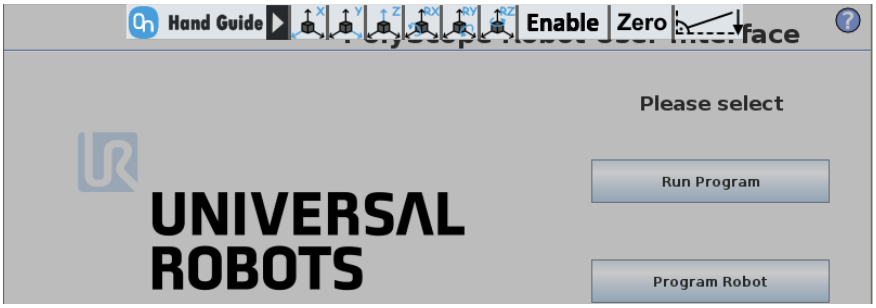


3.2 Hand Guide Toolbar

> It is assumed that the OnRobot Sensor is set up as described in **Example Setup**, and you have not disabled the plugin using the **ENABLE/DISABLE** button described in **URCap Plugin Setup** *(mark J)*.

After turning on the UR robot, the start screen of the PolyScope is visible. After 20 seconds, if it is activated, the OnRobot Hand Guide Toolbar appears *(mark A)*. It is normal to have a yellow warning signal during the boot-up for a few seconds.

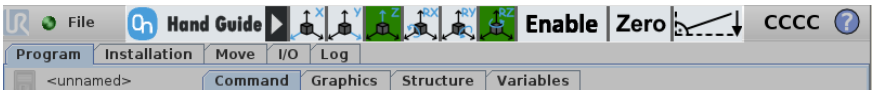




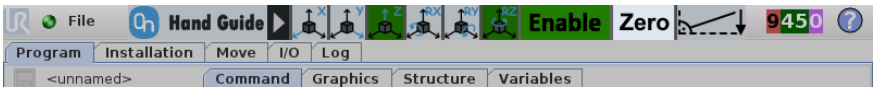
To activate the functions of the toolbar, press on any point of the Toolbar. The Toolbar expands, and the available axes, the **Enable** button the **Zero** button and the **Snap to axes** button will appear.

To select an axis, press the appropriate item. In the following example, the Z and the RZ items are selected.

> The used coordinate system is the Tool.



To deactivate any axis that has been selected, press them again. It is possible to activate or deactivate axes during hand guiding.



To start hand guiding the UR robot, first make sure that you do not touch the tool and then press and hold the **Enable** button, wait until the **Enable** button turns green and drive the robot by hand with the help of the OnRobot sensor. The **Enable** button has changed (background color is now green).

> Make sure that you do not touch the tool before the hand guiding is activated (**Enable** button turned to green), otherwise the robot can behave abnormally (e.g..: the robot could move without any external force exerted). In this case, press the **Zero** button, while you are not touching the tool.
>
> Make sure that you do not use the **Zero** button while touching the OnRobot sensor.

To stop hand guiding of the UR Robot, release the **Enable** button. Immediately after deactivating Hand Guide mode, the Enable button is deactivated for 1 second, and turns into an hourglass icon  .

> Activating hand guiding sets the speed slider of the robot to 100%.

The **Zero** button *(mark D)* can zero the OnRobot Sensor's force and torque values. It is intended to be used when the tool orientation is changed during hand guiding, so that the effects of gravity or changes in the load of the robot can be neutralized.

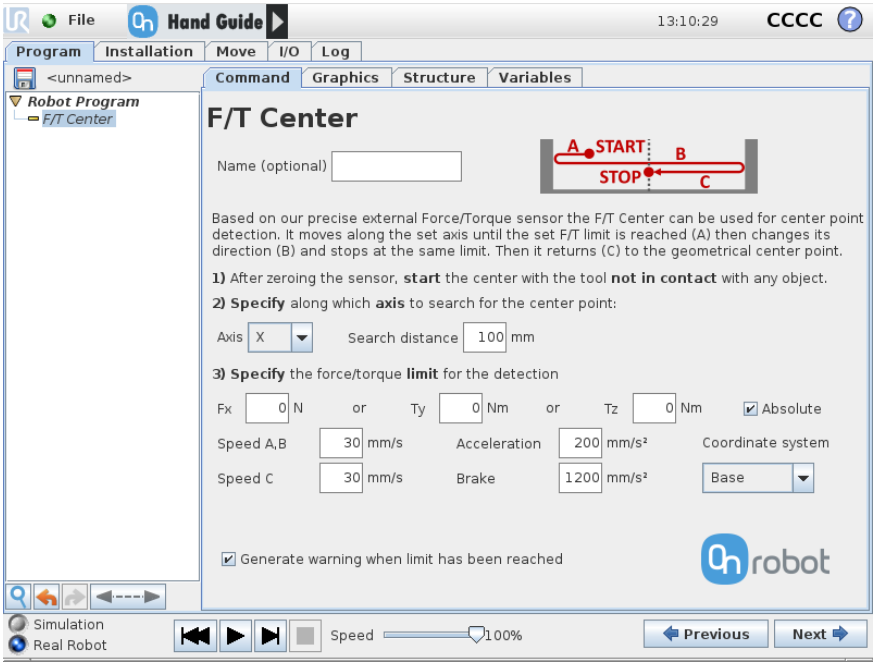
The **Snap to axes** button (*mark E*) rotates the axes of the tool coordinate system to align with the closest axes of the base coordinate system, disregarding negative or positive directions. This allows the user to set the tool to face precisely horizontally, or vertically, after guiding by hand.

3.3 OnRobot URCap Commands

3.3.1 F/T Center Command

It moves the robot along the given axis until it finds an obstacle (section A). After the collision, it moves to the opposite direction until another collision is reached (section B). After that the robot calculates the middle of the two boundary points and moves to that point (section C).

> Make sure that the `F/T Zero` command is used before the `F/T Center` is executed and the tool is not in contact with any object (the force/torque reading is zero).



Axis: Defines whether a translational movement will be carried out along the X, Y or Z axis, or a rotational movement (RX, RY or RZ). Only one axis can be selected.

Search distance: The distance from the starting point how far the command can move the robot (in both directions). The value can be given in meter, centimeter, millimeter, inch or in foot.

> Make sure that it is big enough, otherwise it will not find the right center point.

Force/torque limit: This is the detection limit. The set axis defines the available force/torque values that can be used as a limit.

> Only one of the force/torque options can be active at a time. To change the one that is used clear the previous one (delete the contents of the field) and then set the new one.
>
> If the **Absolute** option is enabled, then it is not important whether the entered value is positive or negative, otherwise the sign sets which direction the searching is started.

Speed A, B: The movement speed while searching for collision (during the A and B section of the movement). (m/s, rad/s)

> The slower the speed during the searching phase is the better to work with hard contacts (such as metal surfaces) to avoid overshoots due to the robot's and the tool's momentum.

Speed C: The movement speed once the center point is calculated and moves towards that point (during the section C of the movement). (m/s, rad/s)

Acc.: The acceleration parameter of the movement (shared parameters across the A, B, and C section). (m/s^2, rad/s^2)

Brake: The deceleration parameter of the movement (shared parameters across the A, B, and C section). (m/s^2, rad/s^2)

Coordinate system: The coordinate system used both for the movement and for the sensor reading. It can be set to `Base` or `Tool` (according to the UR's reference frames).

Generate warning (...): If enabled then a pop-up message (blocking) appears once the set limits are reached or exceeded (center point could not be found). If the center point is found, then no warning is displayed.

If disabled, then no pop-up message is shown but the user can handle any possible errors by the return value of the command. The command updates the `of_return` variable once the command exit. This global variable can be used with the UR's built in `If` conditional expressions (for example: `if of_return == 1` then do something).

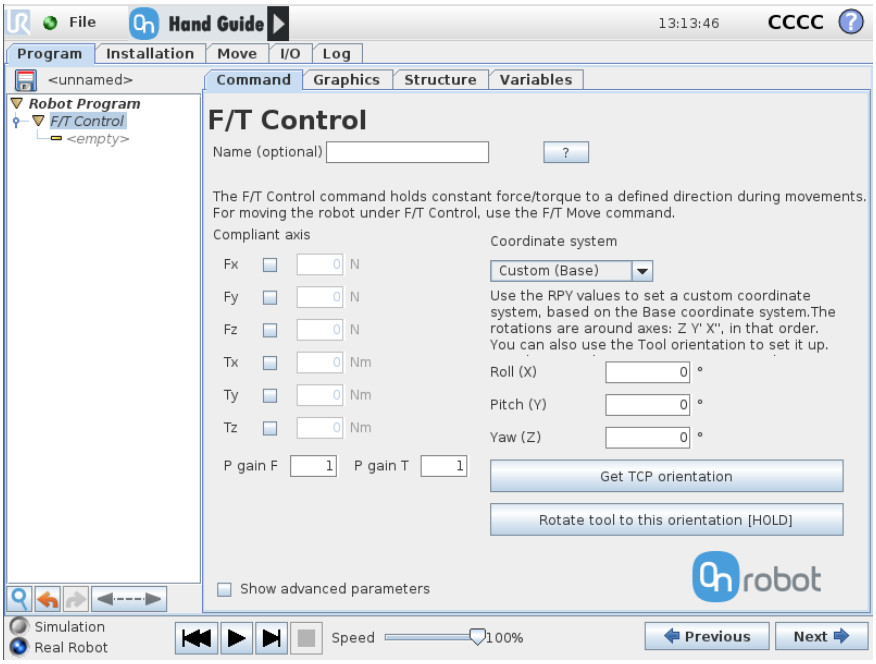
For return values, see **F/T Center Command Return Values**.

3.3.2 `F/T Control` Command

The main purpose of the `F/T Control` command is to provide easy-to-use functions to application programmers, who want to develop force-controlled applications such as polishing, sanding or grinding. A large subset of these applications may require holding constant force/torque to a defined direction during movements.

The command tries to keep the set force/torque values constant along/about the axes set to be compliant while the commands under the `F/T Control` is executed. The `F/T Control` command does not control forces in the direction that the tool is moving using the `F/T Move` and the `F/T Search` commands.

> UR's built-in Move commands cannot be used under the `F/T Control` command. To move the robot under the force control, use the `F/T Move` or `F/T Search` command instead.



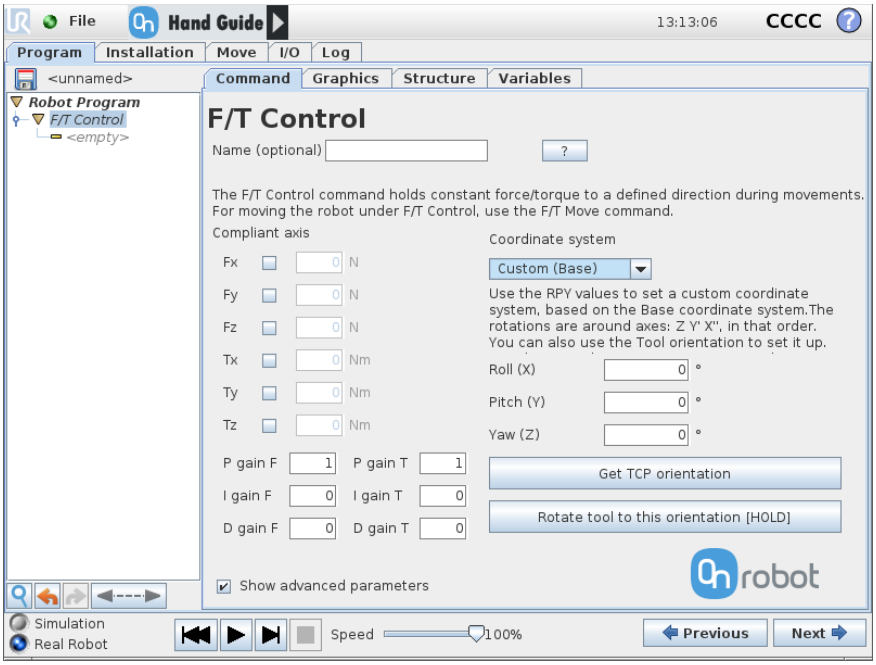
Fx, Fy, Fz, TX, TY, TZ: The axis selection that needs to be compliant. If an axis is enabled (compliant) the movement along/about that axis is force/torque controlled otherwise (non-compliant) position controlled. The enabled axis is controlled to keep the set force/torque value constant. At least one compliant axis must be selected.

Coordinate system: The coordinate system used both for the sensor reading. It can be set to `Base`, `Custom (Base)`, `Tool`, `Custom (Tool)` (according to the UR's reference frames). The Custom coordinate systems are calculated from the basis coordinate system and the given **Roll**, **Pitch** and **Yaw** values. For the Custom (Base) coordinate system, it is possible to use the **Get current TCP pose** button to specify the orientation of the coordinate system, by the orientation of the current TCP. To demonstrate the given orientation, the **Rotate tool to this orientation [HOLD]** button can be used.

P Gain F: The force controller can be tuned with this proportional gain parameter. If any overshoots or vibrations occur, try lowering the gain value (e.g.: 0.5).

P Gain T: The torque controller can be tuned with this proportional gain parameter. If any overshoots or vibrations occur, try lowering the gain value (e.g.: 0.5).

Show advanced parameters checkbox: If checked then more options become available:



I Gain F: The force controller can be tuned with this integral gain parameter. If any overshoots or vibrations occur, try lowering the gain value.

I Gain T: The torque controller can be tuned with this integral gain parameter. If any overshoots or vibrations occur, try lowering the gain value.

D Gain F: The force controller can be tuned with this derivative gain parameter. If any overshoots or vibrations occur, try lowering the gain value.

D Gain T: The torque controller can be tuned with this derivative gain parameter. If any overshoots or vibrations occur, try lowering the gain value.

> Make sure that the `F/T Zero` command is used before the `F/T Control` is started and the tool is not in contact with any object (the force/torque reading is zero).

This command has no return value.

Guidelines to PID force/torque controller settings: The PID force/torque controller continuously calculates the error value for the force/torque measured by the sensor, compared to the values set by the `F/T Control` command, and applies correction based on this error.

P Gain: The proportional term produces a correction that is proportional to the current error value. Increasing this parameter, has the following effects: faster reaction, overreaction, lower error, stability degradation.

I Gain: The integral term produces a correction that is proportional to both, the magnitude, and duration of the past error values. Increasing this parameter, has the following effects: faster reaction, overreaction, lower error, stability degradation.

D Gain: The derivative term produces a correction that is proportional to the slope, or changing speed of past error values. Increasing this parameter, has the following effects: less overreaction, stability increase.

If the force control is too slow, that is, the tool occasionally leaves the surface instead of continually touching it, try increasing the **P Gain**, and **I gain** values.

If the force control is overreacting to changes, that is, the tool bounces off the surface, try decreasing the **P Gain** (or **D Gain**, if it is above 1).

If the force control is reacting to changes too slowly, that is it keeps pushing the surface hard after touching it, try decreasing the **I Gain**.

As a rule of thumb, it is recommended to use values:

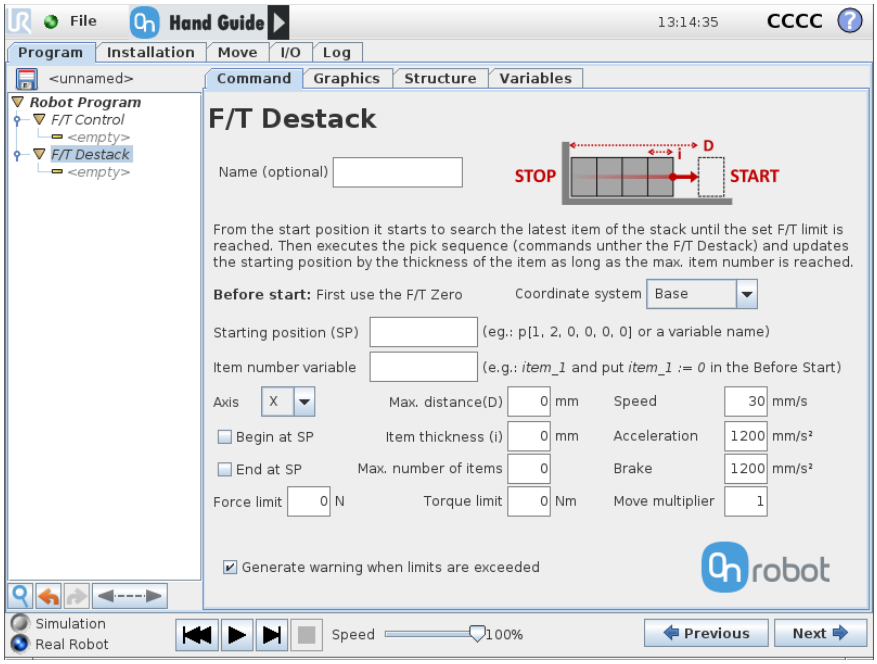
6. P Gain < 5
7. I Gain < 0.25
8. D Gain < 1
9. The ratio of P Gain/I Gain = 10

Values that can be used as a base for tuning are:

P Gain F =1, I Gain F = 0.1, D Gain F = 0.3

P Gain T =0.2, I Gain T = 0, D Gain T = 0

3.3.3 F/T Destack Command



The `F/T Destack` command tries to search for the stack top and then executes the user's pick sequence (for example: closing the gripper). It tracks how many items are destacked that makes easy to handle if the stack is empty. It also works with items that has varying item thickness.

> Make sure that the `F/T Zero` command is used before the `F/T Destack` is executed and the tool is not in contact with any object (the force/torque reading is zero).

Coordinate system: The coordinate system used both for the movement and for the sensor reading. It can be set to `Base` or `Tool` (according to the UR's reference frames).

Starting position (SP): The starting pose can be defined by a constant such as `p[0.1,0.2,0.3,0.9,0.8,0.7]` or by a variable. It needs to be higher than the full stack's top.

Item number variable: The variable that is used to track how many items are destacked successfully. Enter here the variable name that you have previously defined and set to `0`. (E.g.: Use the built-in UR Assignment command `item_1 := 0` in the Before Start section of your program).

Axis: The axis along the destacking is performed (X,Y or Z).

Max. distance (D): The stopping distance along the defined axis. It is measured from the Starting position (SP) and needs to be more than the size of the full stack. The sign defines which direction the destacking is performed along the given axis.

Item thickness (i): The stacked items thickness.

Max. number of items: Defines how many items could be destacked, so how many destacked item makes the stack empty.

Force limit (N): The force limit for the collision detection to find the stack top.

Torque limit (Nm): The torque limit for the collision detection to find the stack top.

Speed: The movement speed while searching for the stack top. (m/s, rad/s)

> The slower the speed during the searching phase is the better to work with hard contacts (such as metal surfaces) to avoid overshoots due to the robot's and the tool's momentum.

Acc.: The acceleration parameter of the movement. (m/s^2, rad/s^2)

Brake: The deceleration parameter of the movement. (m/s^2, rad/s^2)

Move multiplier: Defines how many times of the given speed and force/torque limit is used while the robot is not searching the top of the stack but moving to/from the starting point.

Begin at SP: If enabled, the command will start with moving to the Starting position (SP) at the beginning of its execution.

End at SP: If enabled, the command will exit with moving to the Starting position (SP) at the end of its execution.

Generate warning (...): If enabled then a pop-up message (blocking) appears if the next item is not found or the stack is empty.

If disabled, then no pop-up message is shown but the user can handle any possible errors by the return value of the command. The command updates the `of_return` variable once the command exit. This global variable can be used with the UR's built in `If` conditional expressions (for example: `if of_return == 1` then do something).

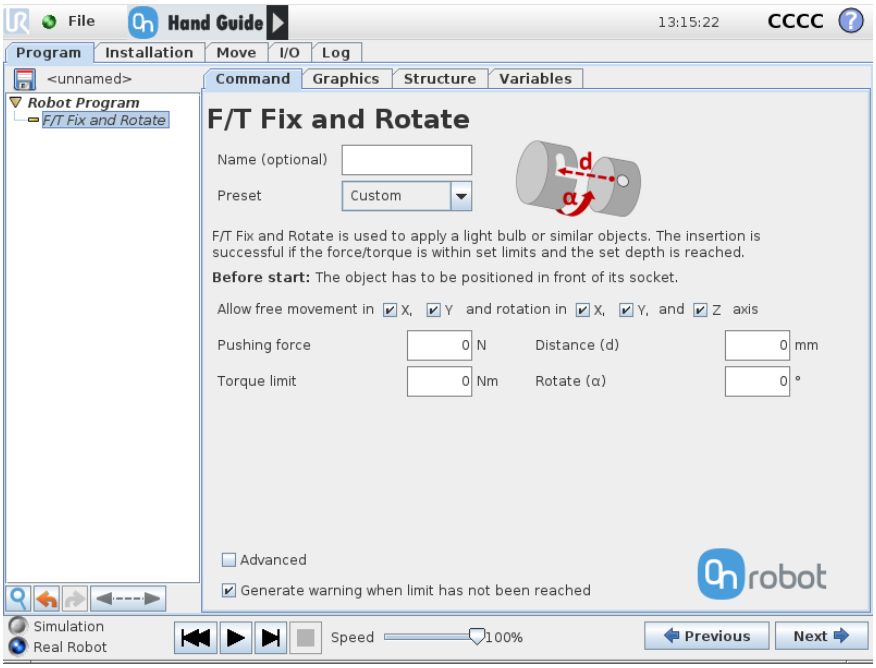
For return values, see **F/T Destack Command Return Values**.

3.3.4 `F/T Fix and Rotate` Command

First position the object that needs to be inserted to the socket pointing in the right direction and close to the socket entrance. The final position and orientation will be corrected by the `F/T Fix and Rotate` command. It tries to push the object with the predefined force limit until the defined insertion depth is reached, and then adjusts the orientation if necessary.

> It is important to set the TCP (Tool Center Point) at the tip of the object.

> Make sure that the `F/T Zero` command is used before the `F/T Fix and Rotate` is executed and the tool is not in contact with any object (the force/torque reading is zero).



X, Y, and Z axis checkboxes: Insertion is performed along the Z axis of the Tool coordinate system. To adopt to any positioning error, the remaining axes (X and Y for the translation and X, Y, and Z for the rotation) can be set to move freely.

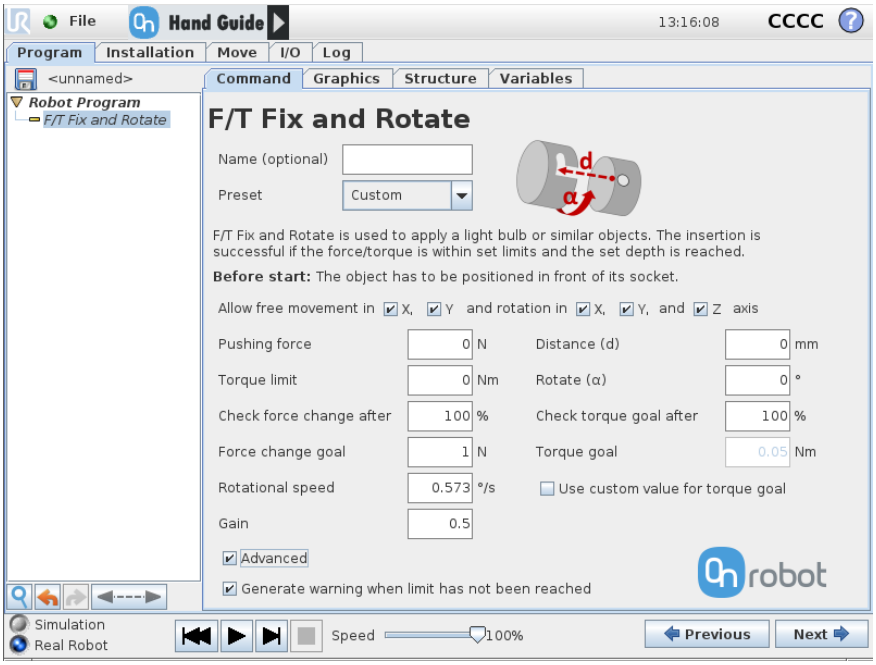
Pushing force (N): The force target used for the force control to gently push the object into the socket.

Distance (d): The distance from the Starting point along the Z axis (in Tool coordinate system).

Torque limit: During the rotation phase this limit is used to finish the movement. The lower the limit the more careful the rotation is.

Rotate (α): The angle for the rotation about the Z axis of the Tool coordinate system.

Advanced mode: If enabled then more options become available:



Check force change after (%): After the object is close to the bottom of the socket, the “bump checking” is activated. The limit how close the object needs to be set by a percent of the **Distance**.

Check torque goal after (%): During the rotational phase after the set percentage of the **Rotate (α)** angle, the torque goal checking is activated.

Force change goal: During the insertion, after the **Check force change after** percent of the **Distance** is reached the force checking is activated. The force checking is used to monitor whether the connector is pushed to the bottom of the socket. This can be set by an additional force limit that is the **Force change goal** value. The push to the bottom of the socket is considered to be reached when the force value equals or exceeds the **Pushing force** + the **Force change goal**.

Torque goal: The set torque value that is going to stop the rotational phase.

Rotational speed (rad/s): The rotational speed during the rotational phase.

Use custom value for torque goal: Check it to be able to set a custom torque goal.

Gain: The gain parameter of the force and torque control. The default value is 0.25. The smaller the value the more accurate the control of the set pushing force is.

Generate warning (...): If enabled then a pop-up message (blocking) appears if the insertion was not successful.

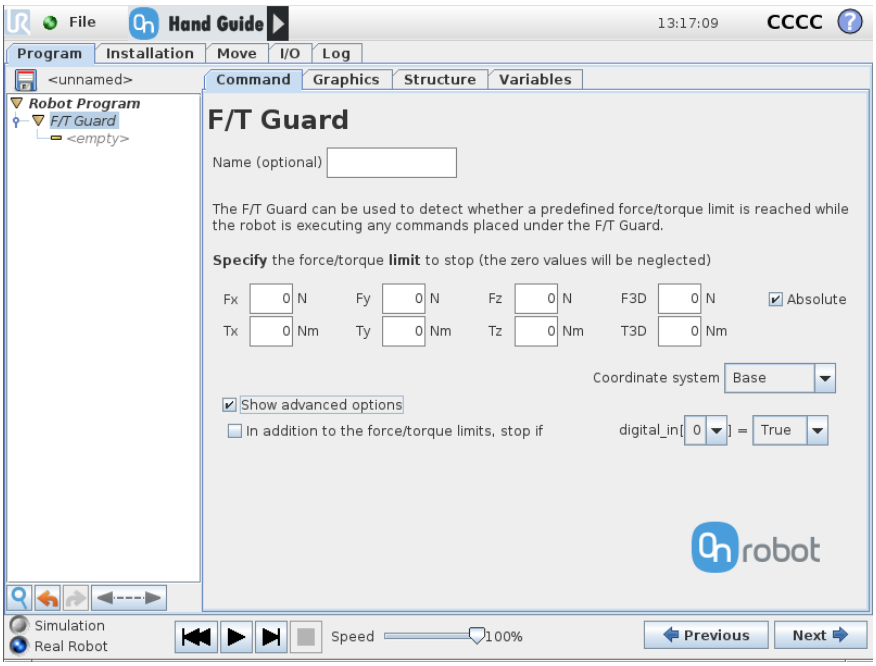
If disabled then no pop-up message is shown but the user can handle any possible errors by the return value of the command. The command updates the `of_return` variable once the command exit. This global variable can be used with the UR's built in `If` conditional expressions (for example: `if of_return == 1` then do something).

For return values, see **F/T Fix and Rotate Command Return Values**.

3.3.5 `F/T Guard` Command

Every UR commands that are put under the `F/T Guard` will be executed, but the robot will stop once one of the set limits is reached. The force limiting can be mixed with an external I/O signal (e.g.: stop if Fz>5 AND digital_in[7] == True).

> Make sure that the `F/T Zero` command is used before the `F/T Guard` is executed and the tool is not in contact with any object (the force/torque reading is zero).



Force/torque limit: This is the detection limit. From the Fx, Fy, Fz, Tx, Ty, Tz, F3D, T3D available options more than one can be set. In this case, if any of those values reached the set threshold the stop is triggered. The values equal to zero are neglected.

> If the **Absolute** option is enabled, then it is not important whether the entered value is positive or negative (e.g.: stop if |Fz| > = 3),

> otherwise the sign defines how the threshold is calculated (e.g..:
> stop if Fz > = 3 or stop if Fz <= -3)

Coordinate system: The coordinate system used both for the movement and for the sensor reading. It can be set to `Base` or `Tool` (according to the UR's reference frames).

Show advanced options: If the **In addition to the force/torque limits…** is enabled then the set digital I/O will also be monitored and once the condition is met (along with the force/torque limit) the robot will be stopped. (e.g.: stop if Fz>5 AND digital_in[7] == True).

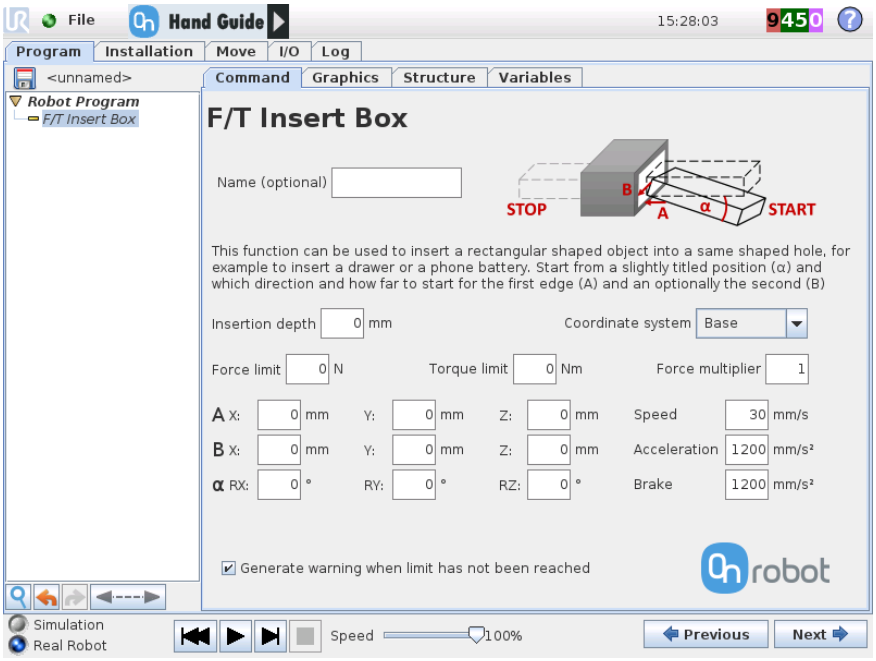
This command has no return value, and halts the program when limits are reached.

3.3.6 `F/T Insert Box` Command

First position the object close to the entrance of the hole and start from a tilted orientation (α). It will move the object in phase A along the predefined axis (for example Z) as long as the edge of the hole is not found. Optionally in phase B another edge can be found (for example the side of the hole). In phase α the orientation is changed so that the object is aligned with the hole (user must set the right angle). Finally, the object is being inserted (along the axis defined in phase A) up to the remaining insertion depth. If the force and torque limits are exceeded a warning is generated.

> It is important to set the TCP (Tool Center Point) at the tip of the object that is being inserted.

> Make sure that the `F/T Zero` command is used before the `F/T Insert Box` is executed and the tool is not in contact with any object (the force/torque reading is zero).



Insertion depth (D): The distance from the Starting point along the defined axis in phase A.

Coordinate system: The coordinate system used both for the movement and for the sensor reading. It can be set to `Base` or `Tool` (according to the UR's reference frames).

Force limit (N): The force limit for the edge detection.

Torque limit (Nm): The torque limit for the orientation adjustment.

Force multiplier: The force limit for the edge detection is multiplied by this value, to calculate the force limit for the final insertion.

Generate warning (…): If enabled then a pop-up message (blocking) appears if the insertion was not successful.

If disabled then no pop-up message is shown but the user can handle any possible errors by the return value of the command. The command updates the `of_return` variable once the command exit. This global variable can be used with the UR's built in `If` conditional expressions (for example: `if of_return == 1` then do something).

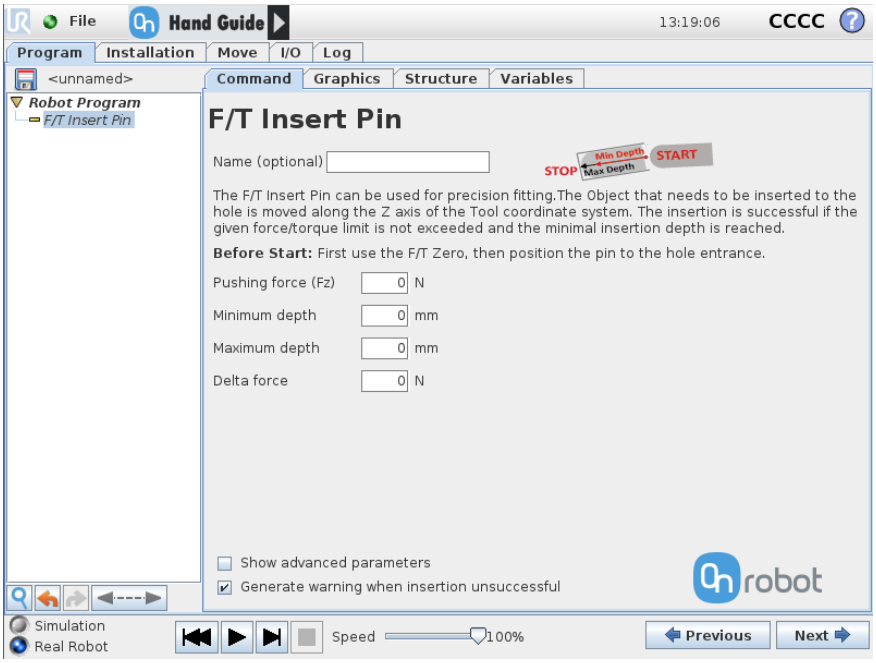
For return values, see **F/T Insert Box Command Return Values**.

3.3.7 F/T Insert Pin Command

First position the pin or peg that needs to be inserted to the hole pointing in the right direction and close to the hole entrance. The finial position and orientation will be corrected by the F/T Insert Pin command. It tries to push the pin with the predefined force limit and adjusts the orientation if necessary. It stops when the defined insertion depth is reached.

> It is important to set the TCP (Tool Center Point) at the tip of the pin or peg.

> Make sure that the F/T Zero command is used before the F/T Insert Pin is executed and the tool is not in contact with any object (the force/torque reading is zero).



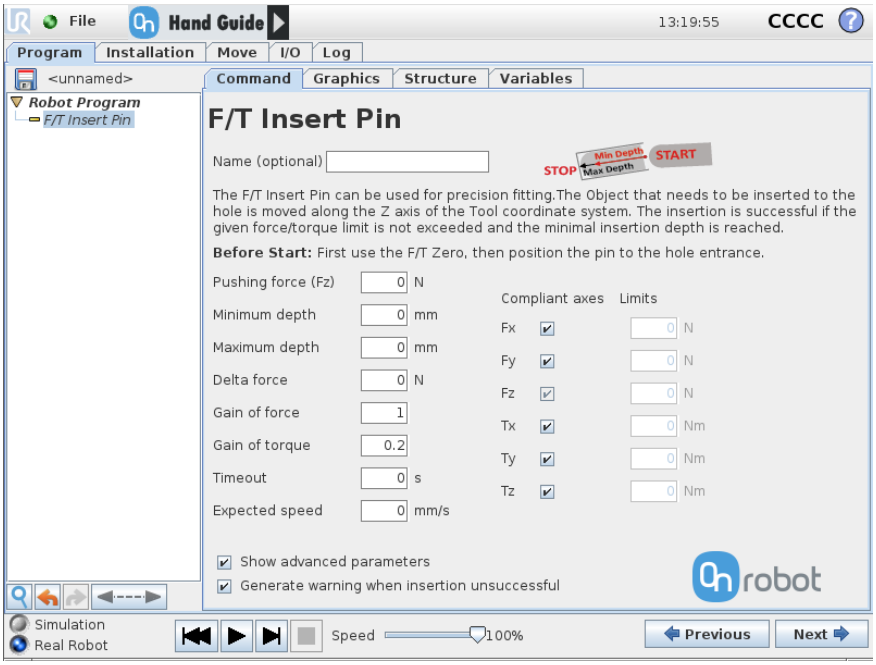
Pushing force: The force target used for the force control to gently push the pin into the hole.

Minimum depth: The minimum distance required to consider the insertion successful, from the starting point along the Z axis (in Tool coordinate system).

Maximum depth: The maximum distance the insertion is allowed to reach, from the starting point along the Z axis (in Tool coordinate system).

Delta force: If this parameter is set, after the **Minimum depth** has been reached, a 'bump', an increase in the pushing force is expected (such as closing a snap-fit joint). This parameter is the additional force on top of **Pushing force** that the insertion is performed with, between the minimum and maximum depths.

Show advanced parameters checkbox: If checked then more options become available:



Gain of force: The proportional gain parameter of the force control for the pushing force, and the side forces on compliant axes.

Gain of torque: The proportional gain parameter of the torque control for the compliant axes.

Expected speed: The minimum speed the insertion is expected to progress with. If this parameter is set, and the insertion is progressing at a slower pace, it is interrupted and considered unsuccessful. If it is set to zero, this exit criterium is neglected.

Timeout: The maximal allowed length of time for the whole insertion function. If it is set to zero, this exit criterium is neglected.

Generate warning (...): If enabled then a pop-up message (blocking) appears if the insertion was not successful.

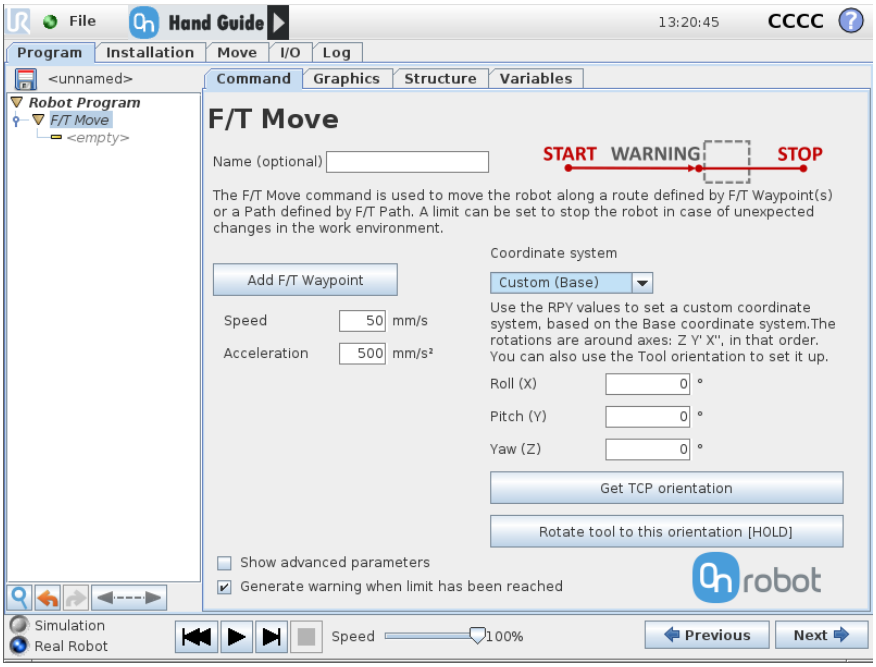
If disabled, then no pop-up message is shown but the user can handle any possible errors by the return value of the command. The command updates the `of_return` variable once the command exit. This global variable can be used with the UR's built in `If` conditional expressions (for example: `if of_return == 1` then do something).

For return values, see **F/T Insert Pin Command Return Values**.

3.3.8 F/T Move Command

The `F/T Move` command is used together with the `F/T Waypoint` command, to move the robot along a route, or together with the `F/T Path` to move the robot along a Path, and stop once the defined force/torque limits are reached (movement interrupted). In this case a warning can be generated. If the movement reaches the last waypoint, the move is successful.

> Make sure that the `F/T Zero` command is used before the `F/T Move` command is executed and the tool is not in contact with any object (the force/torque reading is zero).

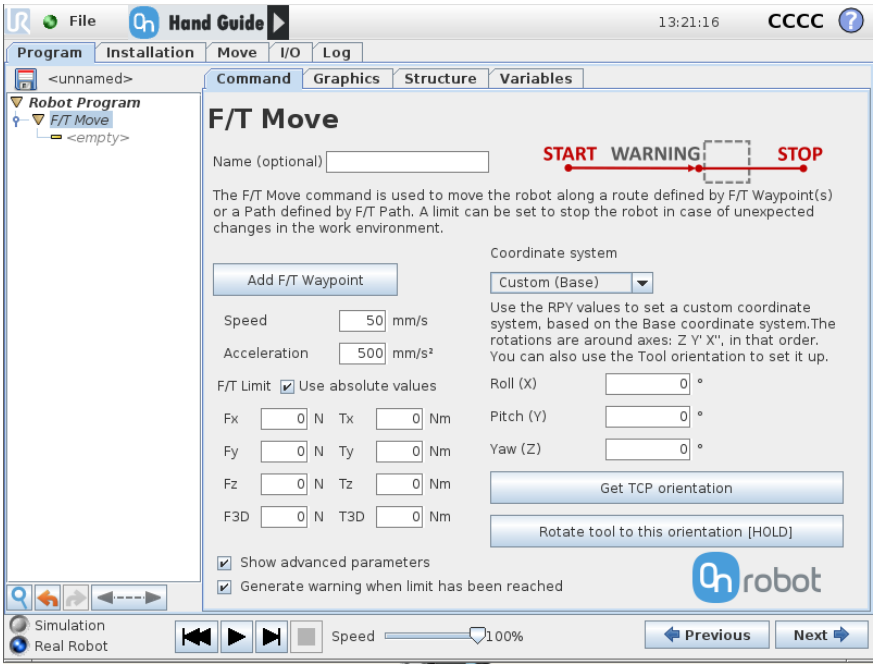


Speed: The movement speed limit while moving. The movement is carried out in a constant translational speed. If the route or path has sharp changes in the direction, or the orientation, the actual speed of the robot may be less than specified, but still constant throughout the route or Path. (mm/s)

Acceleration.: The acceleration and deceleration parameter of the movement. (m/s^2)

Coordinate system: The coordinate system used both for the movement and for the sensor reading. It can be set to `Base`, `Custom (Base)`, `Tool`, `Custom (Tool)` (according to the UR's reference frames). The Custom coordinate systems are calculated from the basis coordinate system and the given **Roll**, **Pitch** and **Yaw** values. For the Custom (Base) coordinate system, it is possible to use the **Get current TCP pose** button to specify the orientation of the coordinate system, by the orientation of the current TCP. To demonstrate the given orientation, the **Rotate tool to this orientation [HOLD]** button can be used.

Show advanced parameters checkbox: If checked then more options become available:



> If the **Use absolute values** option is enabled, then it is not important whether the entered value is positive or negative (e.g.: |Fz| > = 3), otherwise the sign defines how the threshold is calculated (eg.: Fz > = 3 or Fz <= -3)

F/T Limit: This is the detection limit. From the Fx, Fy, Fz, Tx, Ty, Tz, F3D, T3D available options more than one can be set. In this case, if any of the values reach the set threshold the stop is triggered. Values equal to zero are neglected.

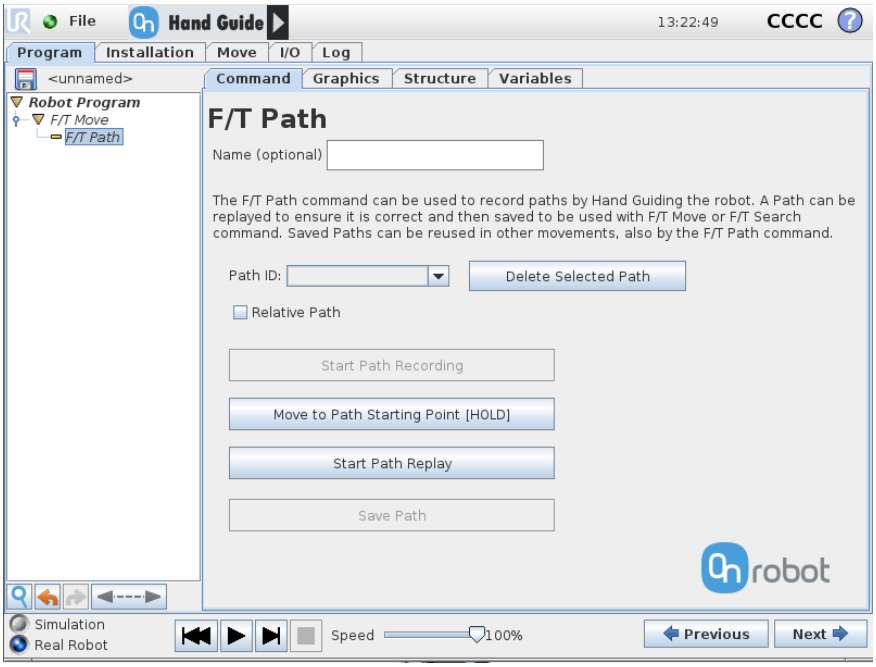
Generate warning (...): If enabled then a pop-up message (blocking) appears if the target position is not reached (the move is not successful). If the move is successful, then no warning is displayed.

If disabled, then no pop-up message is shown but the user can handle any possible errors by the return value of the command. The command updates the `of_return` variable once the it has been executed. This global variable can be used with the UR's built in `If` conditional expressions (for example: `if of_return == 1` then do something).

For return values, see **F/T Move Command Return Values**.

3.3.9 F/T Path Command

The `F/T Path` command is used together with the `F/T Move` or `F/T Search` command, to record and replay a Path.



Path ID dropdown list: Lists identifiers of all the Paths saved on the Compute Box. A Path ID is assigned to a path, when the path is saved. If there is no recorded unsaved path, there is a **Create New…** item, select it to record a new path. If there is a recorded Path that is not saved, there is an **Unsaved** item in the List. Only one unsaved Path can exist, and it can be overwritten by starting Path recording while the **Unsaved** Path is selected.

Delete Selected Path button: Deletes the path currently selected in the **Path ID** dropdown list from the Compute Box, permanently. Do not delete Paths that any other F/T Path commands also used.

Relative Path checkbox: If it is checked, the Path is replayed starting from the actual position of the Tool, instead of the absolute position where it was recorded. If it is unchecked, the Tool moves to the original starting point, and replays the Path from there.

Start Path Recording button: Starts recording a Path, together with the Hand Guide function. If the Hand guide is enabled before pressing the **Start Path Recording** button, recording is immediately started. If the Hand Guide is not enabled **Start Path Recording** button, recording starts only when enabling the Hand Guide function.

Stop Path Recording button: Stops the Hand Guide function and stores the recording to the memory. It does not permanently save the Path.

Move to Path Starting Point [HOLD] button: moves the Tool to the starting position of the Path, it can only be used if the path is not relative.

Start Path Replay button: Replays the Path, even if it is not saved, only stored in the memory.

Stop Path Replay button: Stops replaying the Path.

Save Path button: Saves the Unsaved Path to the Compute Box.

> Rotational movements related to translational movements in Path recording are limited to 4.5 degrees/mm or less, since a larger ratio would cause the Robot to replay the path at a very low translational speed. Rotational movement without translational movement therefore cannot be recorded as Path.

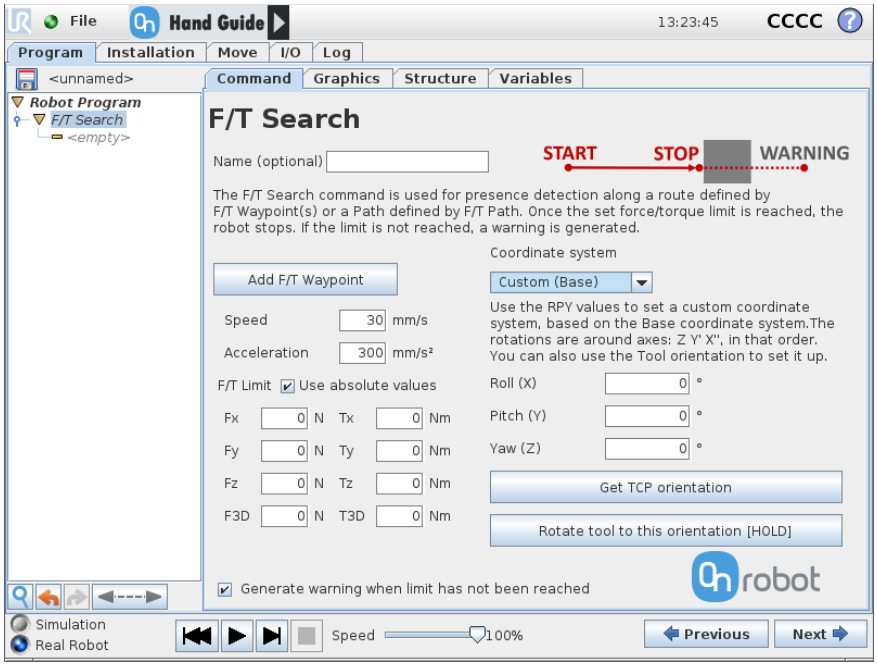
The maximum error of the replayed Path compared to the original recorded movement is 1 mm.

This command has no return value.

3.3.10 F/T Search Command

The F/T Search command is used together with the F/T Waypoint command, to move the robot along a route, or together with the F/T Path to move the robot along a Path and stop once the defined force/torque limits are reached (object found). If the movement reaches the last waypoint or the last point of the Path, the search is not successful (the object is not found) and a warning is generated.

> Make sure that the F/T Zero command is used before the F/T Search command is executed and the tool is not in contact with any object (the force/torque reading is zero).



> If the **Use absolute values** option is enabled, then it is not important whether the entered value is positive or negative (e.g.: |Fz| > = 3), otherwise the sign defines how the threshold is calculated (eg.: Fz > = 3 or Fz <= -3)

Speed: The movement speed while searching for collision. The movement is carried out in a constant translational speed. If the route or path has sharp changes in the

direction, or the orientation, the actual speed of the robot may be less than specified, but still constant throughout the route or Path. (mm/s)

> The slower the speed during the searching phase is the better to work with hard contacts (such as metal surfaces) to avoid overshoots due to the robot's and the tool's momentum.

Acceleration.: The acceleration and deceleration parameter of the movement. (m/s^2)

F/T Limit: This is the detection limit. From the Fx, Fy, Fz, Tx, Ty, Tz, F3D, T3D available options more than one can be set. In this case, if any of the values reach the set threshold the stop is triggered. Values equal to zero are neglected.

Coordinate system: The coordinate system used both for the movement and for the sensor reading. It can be set to `Base, Custom (Base), Tool, Custom (Tool)` (according to the UR's reference frames). The Custom coordinate systems are calculated from the basis coordinate system and the given **Roll**, **Pitch** and **Yaw** values. For the Custom (Base) coordinate system, it is possible to use the **Get current TCP pose** button to specify the orientation of the coordinate system, by the orientation of the current TCP. To demonstrate the given orientation, the **Rotate tool to this orientation [HOLD]** button can be used.

Generate warning (…): If enabled then a pop-up message (blocking) appears once the target position is reached or was already in collision (so the search is not successful. If the search is successful, then no warning is displayed.

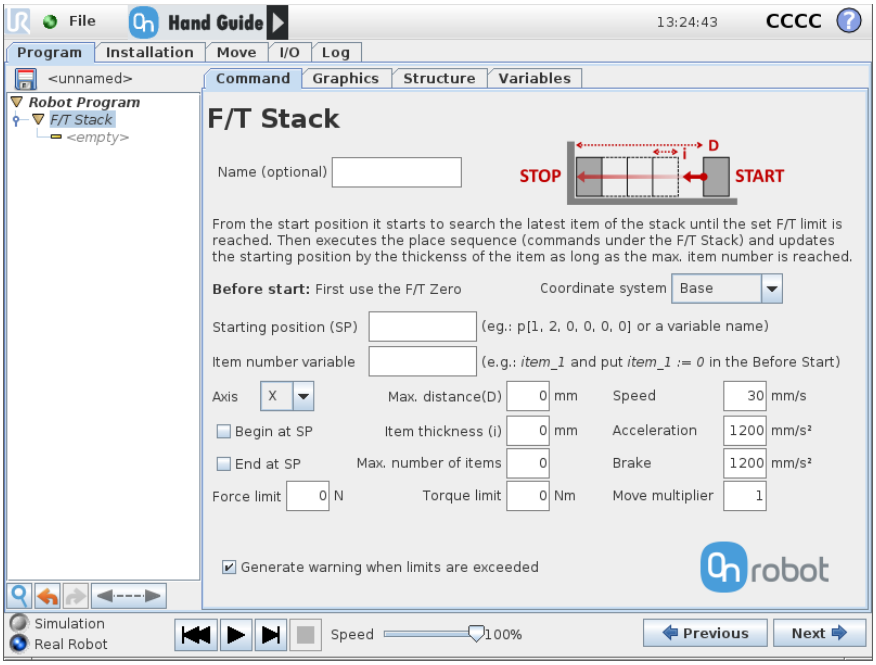
If disabled, then no pop-up message is shown but the user can handle any possible errors by the return value of the command. The command updates the `of_return` variable once it has been executed. This global variable can be used with the UR's built in `If` conditional expressions (for example: `if of_return == 1` then do something).

For return values, see **F/T Search Command Return Values**.

3.3.11 F/T Stack Command

The `F/T Stack` command tries to search for the stack top and then executes the user's place sequence (for example opening the gripper) then exits. It tracks how many items are stacked that makes easy to handle if the stack is full. It also works with items that has varying item thickness.

> Make sure that the `F/T Zero` command is used before the `F/T Stack` is executed and the tool is not in contact with any object (the force/torque reading is zero).



Coordinate system: The coordinate system used both for the movement and for the sensor reading. It can be set to `Base` or `Tool` (according to the UR's reference frames).

Starting position (SP): The starting pose can be defined by a constant such as `p[1,2,3,4,5,6]` or by a variable. It needs to be higher than the full stack's top.

Item number variable: The variable that is used to track how many items are stacked successfully. Enter here the variable name that you have previously defined and set to 0. (E.g.: Use the built-in UR Assignment command `item_1 := 0` in the Before Start section of your program).

Axis: The axis along the stacking is performed (X,Y or Z).

Max. distance (D): The stopping distance along the defined axis. It is measured from the Starting position (SP) and needs to be more than the size of the full stack. The sign defines which direction the stacking is performed along the given axis.

Item thickness (i): The stacked items thickness.

Max. number of items: Defines how many items could be stacked, so how many stacked item makes the stack full.

Force limit (N): The force limit for the collision detection to find the stack top.

Torque limit (Nm): The torque limit for the collision detection to find the stack top.

Speed: The movement speed while searching for the stack top. (m/s, rad/s)

> The slower the speed during the searching phase is the better to work with hard contacts (such as metal surfaces) to avoid overshoots due to the robot's and the tool's momentum.

Acc.: The acceleration parameter of the movement. (m/s^2, rad/s^2)

Brake: The deceleration parameter of the movement. (m/s^2, rad/s^2)

Move multiplier: Defines how many times of the given speed and force/torque limit is used while the robot is not searching the top of the stack but moving to/from the starting point.

Begin at SP: If enabled, the command will start with moving to the Starting position (SP) at the beginning of its execution.

End at SP: If enabled, the command will exit with moving to the Starting position (SP) at the end of its execution.

Generate warning (…): If enabled then a pop-up message (blocking) appears if the next item is not found or the stack is full.

If disabled, then no pop-up message is shown but the user can handle any possible errors by the return value of the command. The command updates the `of_return` variable once the command exit. This global variable can be used with the UR's built in `If` conditional expressions (for example: `if of_return == 1` then do something).

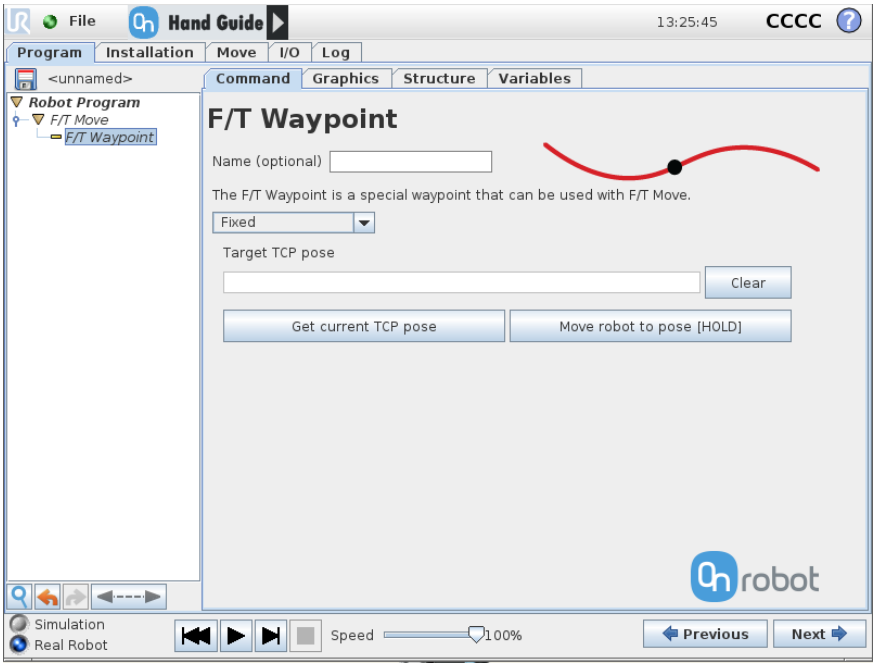
For return values, see **F/T Stack Command Return Values**.

3.3.12 `F/T Waypoint` Command

The `F/T Waypoint` command is used together with the `F/T Move` or `F/T Search` command, to move the robot along a route. There are three types of waypoints (Fixed, Relative, and Variable), that can be used in any combination.

> Do not use consecutive `F/T Waypoints` that only contain rotations in the same `F/T Move` command. Use more than one `F/T Move` commands to achieve rotations without translational movements.

Waypoint type: The type of the waypoint. It can be set to `Fixed`, `Relative` or `Variable`.

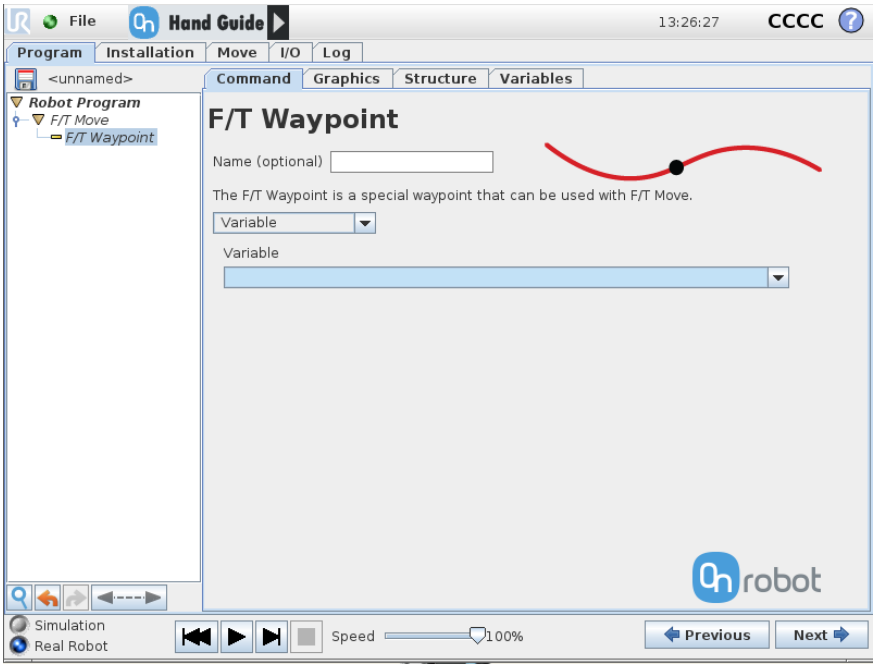


Target TCP pose: The position represented by the waypoint in the robot route. It is a read-only field, and can be filled by using the **Get current TCP pose** button.

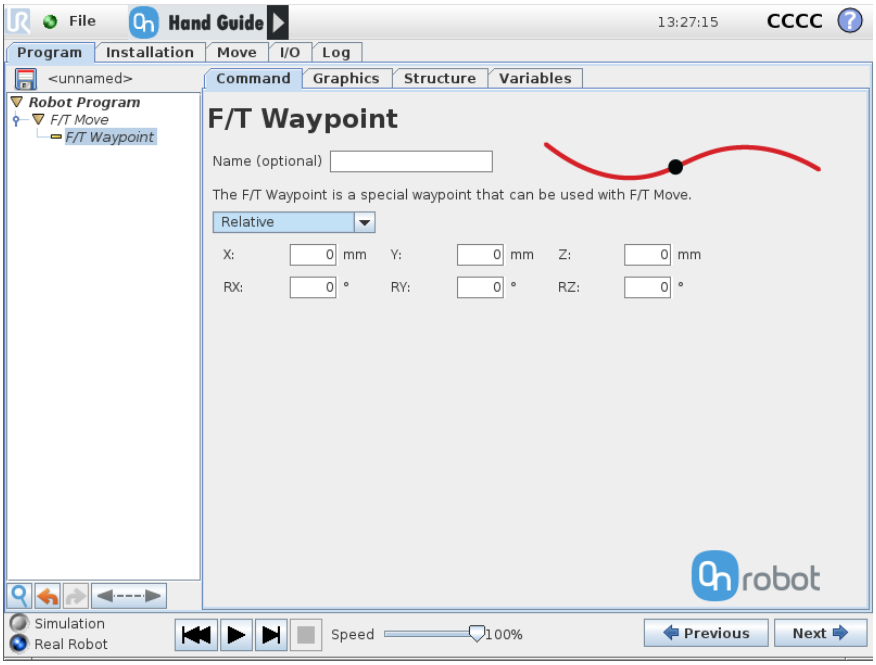
Clear button**:** deletes the contents of the **Target TCP pose** field.

Get current TCP pose button: inserts the current TCP coordinates into the **Target TCP pose** field.

Move robot to pose [HOLD] button: moves the robot to the pose set in the **Target TCP pose** field.



Variable: The position represented by the waypoint in the robot route. The target pose can be defined by a variable.



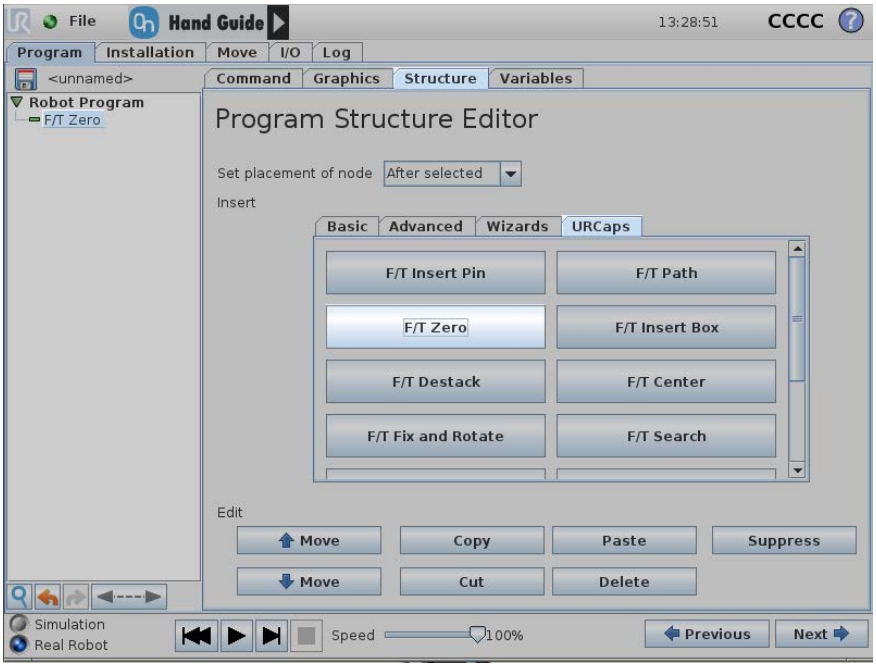
Relative X, Y, Z, RX, RY, RZ: the distances and rotations this waypoint represents, compared to the previous robot position.

3.3.13 F/T Zero Command

This command can bias the force and torque values of the sensor.

> Make sure that the tool is not in contact with any object.

Follow this procedure to add the command:



1. Select the **Structure** tab.
2. Select the **URCaps** tab within the **Structure** page.

3. Press the **F/T Zero** button.
4. The `F/T Zero` command appears under the `Robot Program` node.

This command has no return value.

3.4 TCP Offset Propagation

Each time the TCP offset is changed within any program, the following script needs to be run, to propagate this change to the Compute Box: of_send_tcp_offset()

3.5 Application Examples

3.5.1 Collision Detection

Collision detection can be implemented by the following commands:

1. `F/T Search`: It can be used for Presence detection. It searches for an object and stops once it is found. If the object could not be found it gives a warning message. If the position of an object is varying it can also be used to easily determine its exact location.
2. `F/T Move`: It can be used for Force/torque limited movements. It is similar to the UR's Move command but with built-in force/torque limitation and supports relative offset type parameters (e.g.: move 1 cm or 1 inch along the Z axis).
3. `F/T Guard`: It can be used in combination with any UR command to limit the exerted force/torque. It monitors the set limits in parallel to your code and once the set limits are reached it stops the robot.

The `programs/OnRobot_UR_Programs` folder contains a collision detection example UR program, called *OnRobot_Collision_Detection_Example.urp*.

3.5.2 Center Point Detection

With the help of gentle contacts the robot can be positioned to the geometrical center point of a hole. It also works with shiny metal objects that are usually impossible with camera based solutions.

The `programs/OnRobot_UR_Programs` folder contains a collision detection example UR program, called *OnRobot_Centerpoint_Detection_Example.urp*.

3.5.3 Polishing and Sanding

For any polishing or sanding task, it is very important to keep the predefined force value constant. This task can be achieved with our force/torque control functions, that requires the following two commands to be used:

1. `F/T Control`: This command is similar to the UR's built-in Force command, but uses the OnRobot more precise force/torque sensor as an input to achieve an excellent result even with low forces. The force/torque control tries to keep the defined force/torque constant on the axes that are set to be compliant. The non-compliant axes are position controlled (only with the `F/T Move` command).

2. F/T Move: It can be used to position control (move) the robot along/about the non-compliant axis in the F/T Control.

The programs/OnRobot_UR_Programs folder contains a collision detection example UR program, called *OnRobot_Plastic_Partingline_Removal_Example.urp*.

3.5.4 Stacking and Destacking

The task of stacking or unstacking items are commonly used in application such as machine tending or collaborative assembly. Working with fragile or flexible parts it is very important to have a precise sensor that makes sure during the seek phase the items are not harmed. In case of rigid parts, positioning inaccuracy or parts variance can make the task hard to achieve. With the following two commands, anyone can easily solve these challenges quickly and easily:

1. F/T Stack: This command is similar to the UR's built-in Stacking (Seek) command, but uses the OnRobot precise force/torque sensor as an input to achieve an excellent result even with low forces. The items are moved to the next stack position and also monitors whether the stack is full or not.

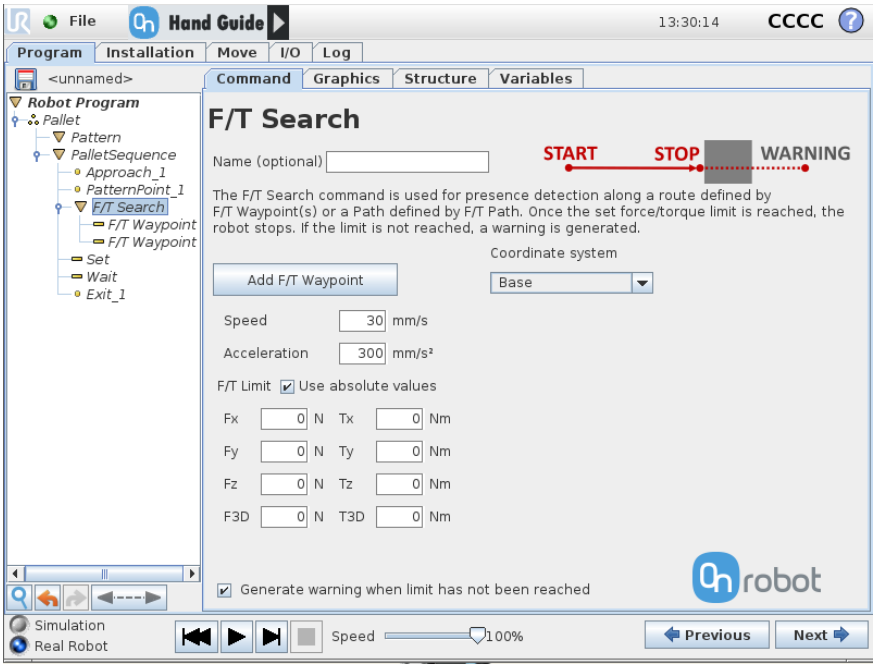
2. F/T Destack: This command is also similar to the UR's built-in Destacking (Seek) command, but uses the OnRobot precise force/torque sensor as an input to achieve an excellent result even with low forces. The items at the next destack position is found (and moved to the users's target) and also monitors whether the stack is empty or not.

The programs/OnRobot_UR_Programs folder contains a collision detection example UR program, called *OnRobot_Stacking_Example.urp*.

3.5.5 Palletizing

Palletizing objects that need to be handled with care can be a challenging task. Placing flexible cardboard boxes next to each other requires more than just a simple positioning in a fixed pattern. Using the UR's built-in palletizing command in combination with our F/T Search command anyone can easily solve these challenging tasks.

First set up the UR's built-in Pallet command, to achieve the required pattern. Make sure that the positions are a little bit further than what is going to be the final position. That allows the F/T Search command to find the neighboring item by a gentle touch to adopt to any positioning errors.



If necessary, more than one F/T Search can be used to align the item horizontally and vertically.

> Make sure to use only the relative offset type of input parameters of the F/T Search command to be always relative to the pattern.

For more information see **F/T Search command**.

The `programs/OnRobot_UR_Programs` folder contains a collision detection example UR program, called *OnRobot_Palletizing_Example.urp*.

3.5.6 Pin Insertion

Inserting pins or pegs into tight holes cannot be achieved with traditional position based solutions. Even with cameras one cannot have a robust solution.

With the help of the precise OnRobot F/T sensor and the `F/T Insert Pin` command anyone can easily and robustly solve tasks that requires precision fitting.

The `programs/OnRobot_UR_Programs` folder contains a collision detection example UR program, called *OnRobot_Pin_Insertion_Example.urp*.

3.5.7 Box Insertion

Inserting a rectangular object into a rectangular hole is a common task, such as inserting a car radio assembly into the radio bracket or inserting a battery into a phone.

With the help of the `F/T Insert Box` command anyone can easily solve these tasks.

The `programs/OnRobot_UR_Programs` folder contains a collision detection example UR program, called *OnRobot_Box_Insertion_Example.urp*.

3.5.8 Fix and Rotate

With the help of the precise OnRobot F/T sensor and the `F/T Fix and Rotate` command anyone can easily and robustly solve tasks that requires using any Bayonet type of mounting.

4 Glossary of Terms

Term	Description
Compute Box	A unit provided by OnRobot along with the sensor. It performs the calculations needed to use the commands and applications implemented by OnRobot. It needs to be connected to the sensor and the robot controller.
OnRobot Data Visualization	Data visualization software created by OnRobot, to visualize the data provided by the sensor. Can be installed on Windows operating system.

5 List of Acronyms

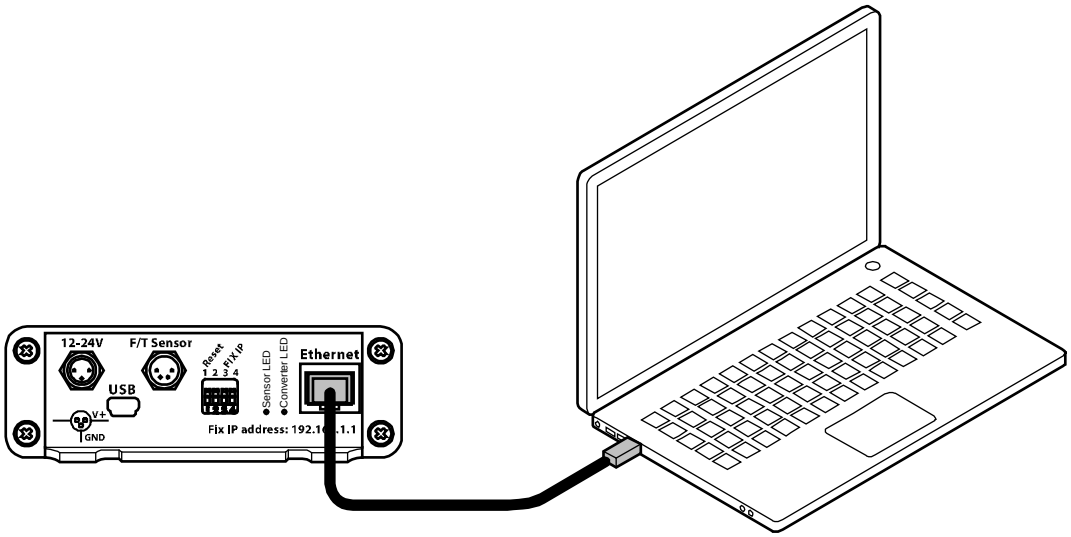
Acronym	Expansion
DHCP	Dynamic Host Configuration Protocol
DIP	dual in-line package
F/T	Force/Torque
ID	Identifier
IP	Internet Protocol
IT	Information technology
MAC	media access control
PC	Personal Computer
RPY	Roll-Pitch-Yaw
SP	Starting Position
SW	software
TCP	Tool Center Point
UR	Universal Robots
URCap	Universal Robots Capabilities
USB	Universal Serial Bus
UTP	unshielded twisted pair

6 Appendix

6.1 Changing the IP of the Compute Box

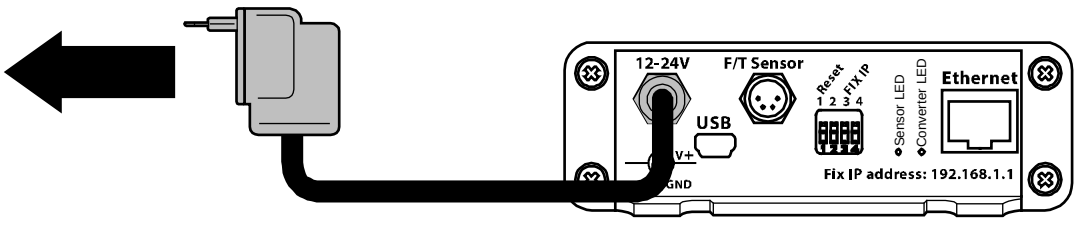
To change the IP address of the sensor, connect your laptop or an external PC to the OnRobot Compute Box.

1. Make sure that the device is not powered. Connect the device and the computer with the provided Ethernet cable.

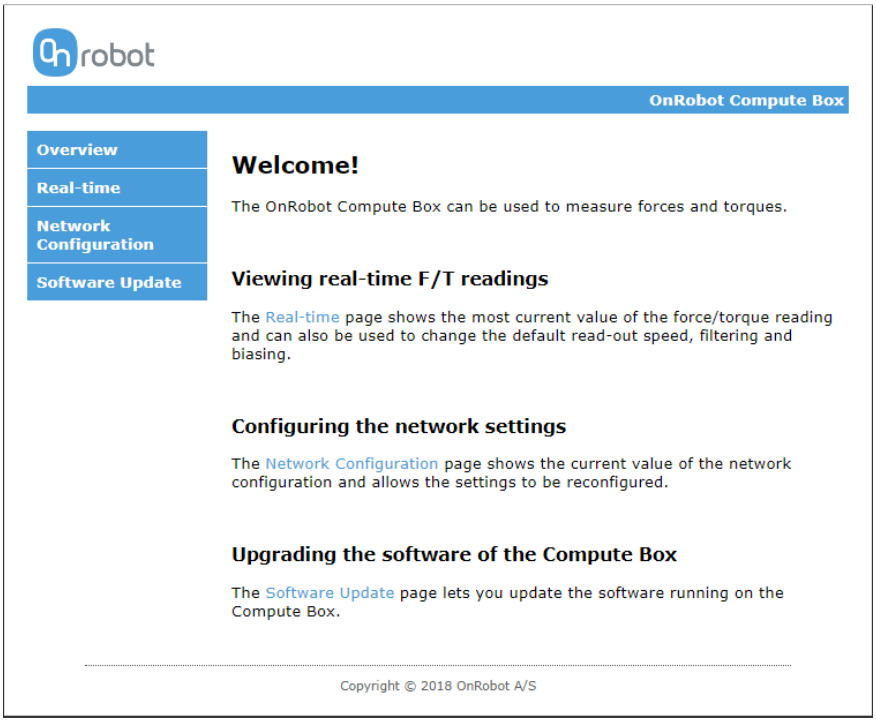


2. If your device is in the factory default settings, proceed to step 3. Otherwise, make sure to switch the DIP switch 3 to the ON position (up) and the DIP switch 4 to the OFF position (down).

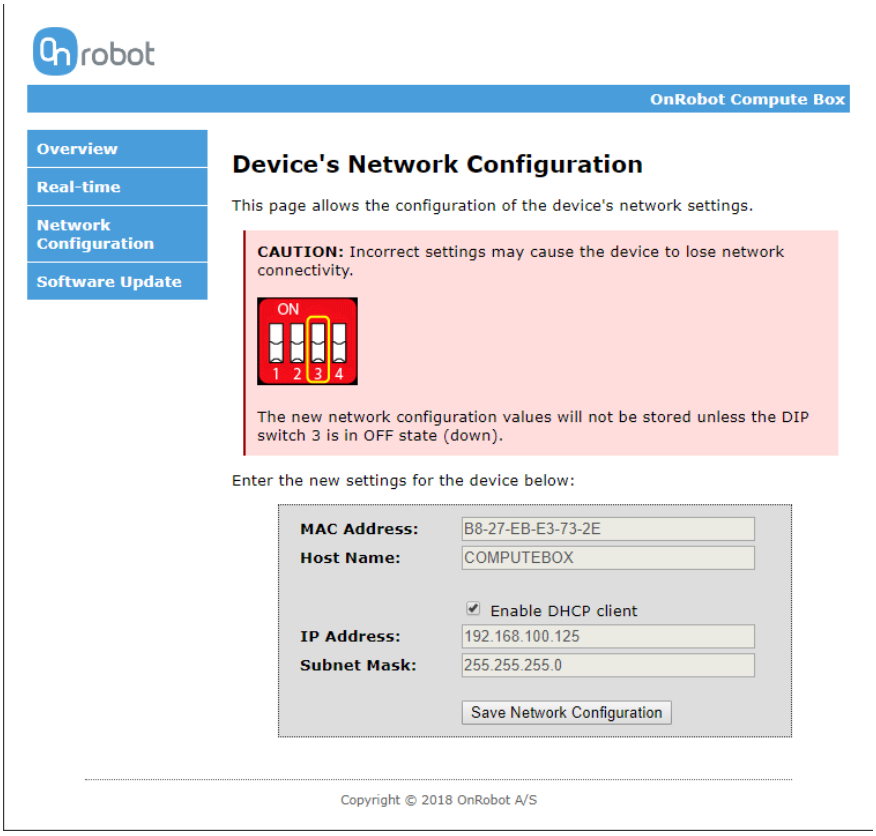
3. Power the device from the provided power supply and wait 30 seconds for the device to boot up.



4. Open a web browser (Internet Explorer is recommended) and navigate to **http://192.168.1.1**. The welcome screen is displayed.



5. Click on **Network Configuration** in the left-side menu. The following screen is displayed:



6. Uncheck the Enable DHCP client checkbox, if it is checked.
7. Edit the IP Address.
8. Set DIP switch 3 to off position.

9. Click the Save Network Configuration button
10. Open a web browser (Internet Explorer is recommended) and navigate to the IP Address set in step 7.

6.2 Updating the software on the Compute Box

Refer to the Compute Box Description document.

6.3 Software Uninstallation

1. To uninstall (remove) the previously copied OnRobot UR program files, choose from the following options:
    a. Remove the files and folder, using the **Delete** option of the Teach Pendant during file operation (for example, Load Program, Save Program)
    b. Copy the `uninstall.sh` file from the USB drive to a new USB drive, rename it to `urmagic_OnRobot_uninstall.sh` and plug it into the Teach Pendant. The file creates a backup copy to the USB drive, and then it permanently deletes the `OnRobot_UR_Programs` folder from the UR.
2. Uninstall the URCap plugin.
    a. Go to the Welcome screen of the PolyScope.
    b. Click **Setup Robot**.
    c. Click on **URCaps Setup** and locate the `OnRobot Force Torque Sensor` in the list of active URCaps.
    d. Click on the **-** sign at the bottom to uninstall it.
    e. Restart the robot.

6.4 Return Values

6.4.1 `F/T Center` Command Return Values

0 Arrived successfully to the center point.

1 The first boundary search was unsuccessful. Movement reached the distance limit.

2 The second boundary search was unsuccessful. Movement reached the distance limit.

3 Could not reach the center point. The tool collided during the movement.

4 The search has not been started because of the conditions.

5 The second search has not been started because of the conditions.

99 Do not define more than one directional parameter.

6.4.2 `F/T Control` Command Return Values

This command has no return value.

6.4.3 F/T Destack Command Return Values

0	One iteration of the destacking is complete.
1	The iteration counter is over the maximum: the stack is empty.
2	Destacking is unsuccessful. Next item not found.
3	Destacking cannot start due to a force or torque exceeding the set limit.
4	The movement to the next element was unsuccessful, a collision occurred.
5	The movement to the starting point was unsuccessful, a collision occurred.

6.4.4 F/T Fix and Rotate Command Return Values

0	The Fix and Rotate finished with no error.
11	The Orientation centerpoint search of Ry was unsuccessful.
12	The Orientation centerpoint search of Ry was unsuccessful.
21	The rotation was unsuccessful, collision occurred.
22	The rotation ended without contact.
99	Parameter error.

6.4.5 F/T Guard Command Return Values

This command has no return value, and halts the program when limits are reached.

6.4.6 F/T Insert Box Command Return Values

0 The Box Insertion finished with no error.

1 The first direction search was unsuccessful. Movement reached the distance limit.

2 The second direction search was unsuccessful. Movement reached the distance limit.

3 The tilt back movement was unsuccessful. Collision occurred.

4 The tilt movement was unsuccessful. Collision occurred.

5 The box stuck during the insertion state while center pointing of the X axis! Please check the position and orientation.

6 The box stuck during the insertion state while center pointing of the Y axis! Please check the position and orientation.

7 The box stuck during the insertion state while center pointing of the Z axis! Please check the position and orientation.

8 The box cannot be inserted to position, too many collisions occurred. Please check the position and orientation.

6.4.7 F/T Insert Pin Command Return Values

0 The Insert Pin command reached the maximal distance.

1 The Insert Pin command exited at a bump after the minimal insertion depth.

2 The Insert Pin command got stuck after the minimal insertion depth. Insertion is slower than required.

3 The Insert Pin command got stuck before the minimal insertion depth. Insertion is slower than required.

4 The Insert Pin command exited with timeout after the minimal insertion depth.

5 The Insert Pin command exited with timeout before the minimal insertion depth.

6 The Insert Pin command exited due to too high side-forces/-torques at the non-compliant axes after the minimal insertion depth.

7 The Insert Pin command exited due to too high side-forces/-torques at the non-compliant axes before the minimal insertion depth.

8 The Insert Pin command has a parameter error.

6.4.8 F/T Move Command Return Values

0 The move ended without detecting a force or torque greater than the set limit.

1 The move ended, because a force or torque greater than the set limit was detected.

3 The move cannot start due to a force or torque exceeding the set limit.

11 The move cannot start, because there is no recorded Path on the Compute Box with the selected.

12 The move cannot start, because there are no recorded points in this Path.

13 The move cannot start, because the Path file found at this Path ID is empty.

14 The move cannot start, because the Path file is corrupted.

6.4.9 F/T Path Command Return Values

This command has no return value.

6.4.10 F/T Search Command Return Values

0 The search ended successfully, because a force or torque greater than the set limit was detected.

1 The search ended without detecting a force or torque greater than the set limit.

3 The search cannot start due to a force or torque exceeding the set limit.

11 The search cannot start, because there is no recorded Path on the Compute Box with the selected.

12 The search cannot start, because there are no recorded points in this Path.

13 The search cannot start, because the Path file found at this Path ID is empty.

14 The search cannot start, because the Path file is corrupted.

6.4.11 F/T Stack Command Return Values

Return values:

0 One iteration of the stacking is complete.

1 The iteration counter is over the maximum: the stack is full.

2 Stacking is unsuccessful. Next item not found.

3 Stacking cannot start due to a force or torque exceeding the set limit.

4 The movement to the next element was unsuccessful, a collision occurred.

5 The movement to the starting point was unsuccessful, a collision occurred.

6.4.12 F/T Waypoint Command Return Values

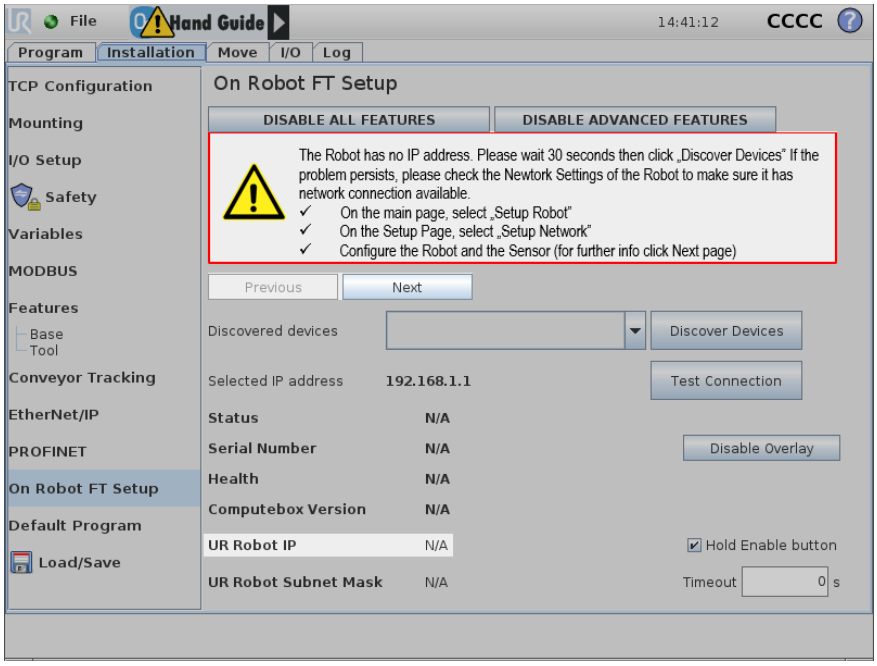
This command has no return value.

6.4.13 F/T Zero Command Return Values

This command has no return value.

6.5 Troubleshooting

6.5.1 “The Robot has no IP address”

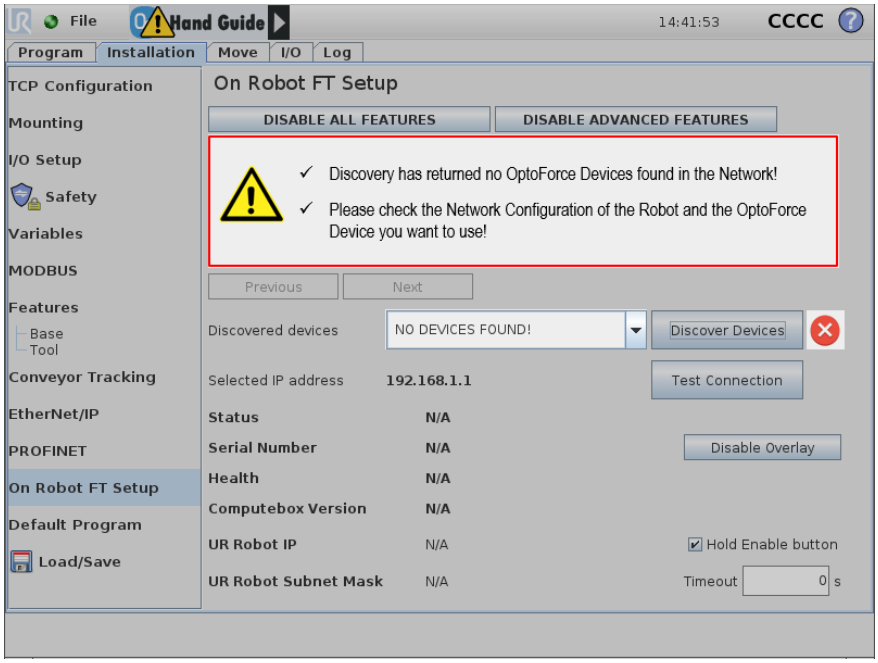


- The quick guide provides a possible solution to this problem.
- This guide has more than one page. Change pages with the Prev. page and Next page buttons.
- As shown in the example, the UR Robot has no IP address (“N/A” is shown). This indicates an error in the Network Configuration of the UR robot.

This error can happen, when the Network Configuration of the UR Robot has not been finished. The Network Configuration might not be finished when you select the **OnRobot Setup** page just after turning on the robot or when the Network Configuration has just been modified and the **OnRobot Setup** page is selected within 60 seconds.

In these cases, wait 60 seconds and use the **Discover Devices** function. If the operation of **Discovering Devices** is successful, move on to the next steps as described in **Example Setup**.

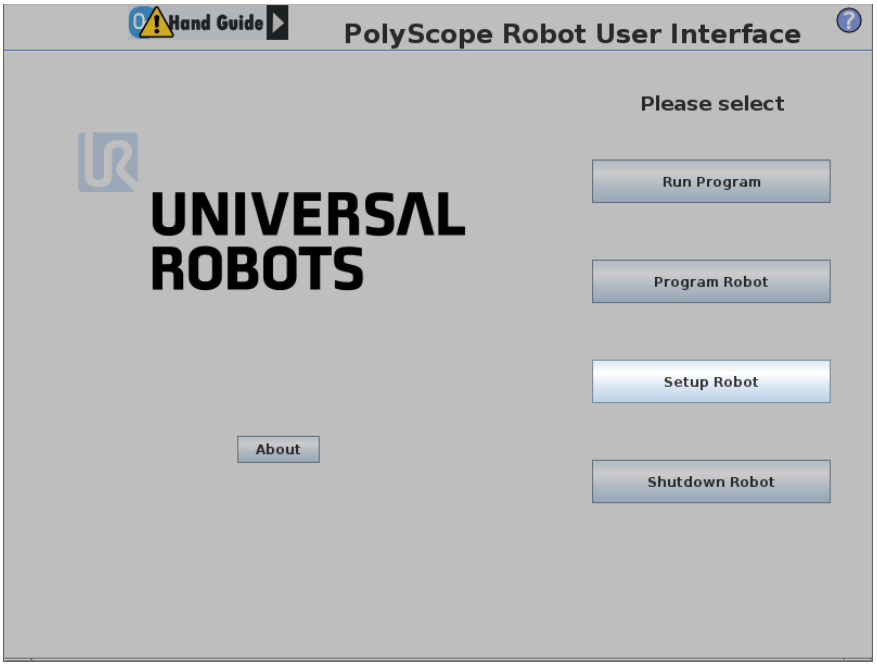
If the error is still present after pressing the **Discover Devices** button, the following screen is shown:



- The error description.
- "NO DEVICES FOUND!" in the device selection list.
- The red "X" indicates that the **Discover Devices** operation failed.

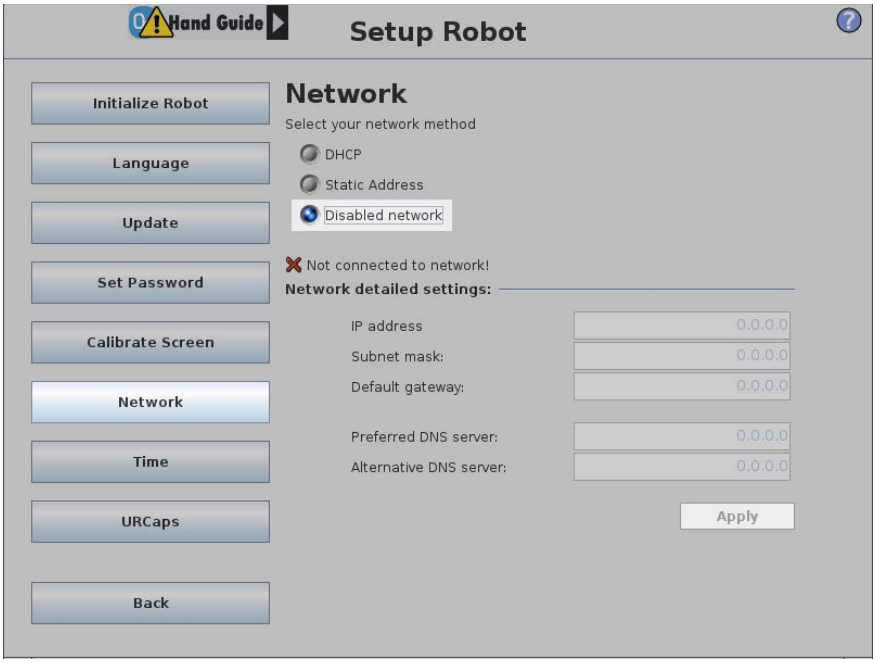
To resolve the problem, check the Network Configuration of the UR Robot, by doing the following:

10. Press the Setup Robot button.



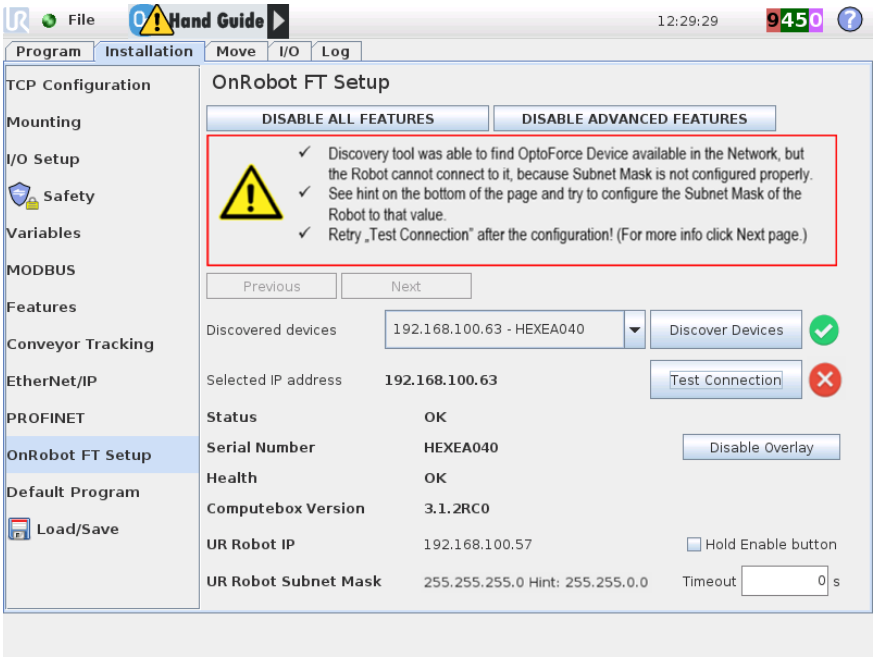
11. Press the Setup Network button.
12. If the network of the UR is disabled:
13. If the OnRobot device is connected to the UR robot directly, select DHCP, and press the Apply button (mark 4). The OnRobot service assigns an IP.

14. If the OnRobot device is not directly connected to the UR robot, check if the OnRobot device is connected to the same network (router, switch, and so on) as the UR Robot, or consult the Network Supervisor.
15. If DHCP or Static Address is selected, and the problem remains consult your Network Supervisor.



In case of a DHCP, after the proper IP address is assigned to the UR robot switch to Static address mode (the IP address of the UR robot should remain the same) and press the **Apply** button. The IP address is now fixed and does not change later.

6.5.2 “Discovery Tool was able to find OnRobot Device available in the Network”



A. The guide page of the error.

B. The **Discover Devices** operation was successful (the green tick icon can be seen).

C. Test Connection failed.

D. “`Robot and Device are not in the same subnet`” status message is shown.

E. UR Robot Subnet Mask is extended with “`Hint: 255.255.0.0`”.– (HINT).

This error occurs when the OnRobot Device and the UR Robot are not in the same Subnet.

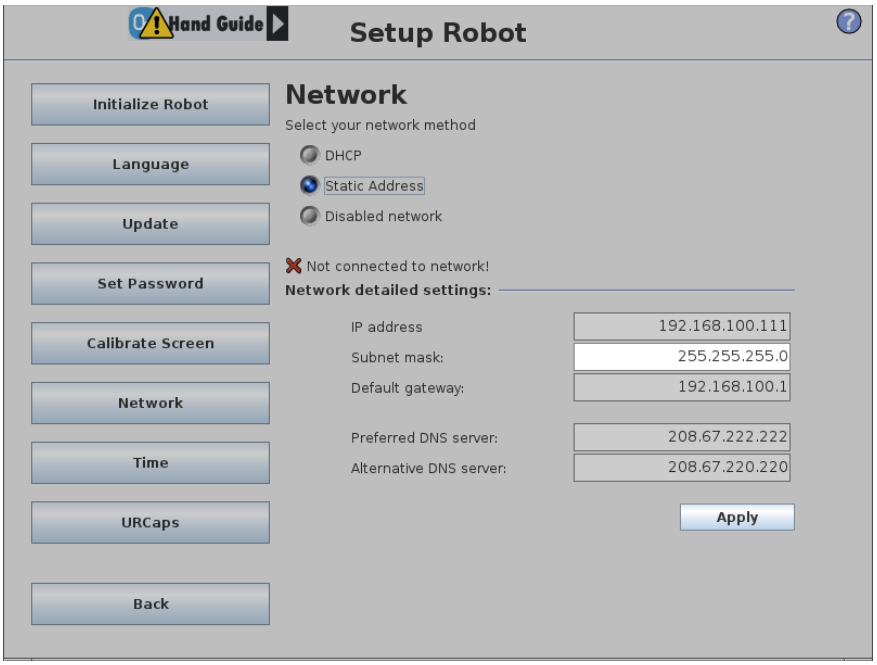
To resolve the problem, follow this procedure:

1. If the OnRobot device is not directly connected to the UR Robot, check if the DIP Switch 3 is in off state on the Compute Box, as shown on the following figure:

2. If the DIP Switch is in on state, set it to off, then restart the OnRobot device (by disconnecting the power) and repeat the steps in Example Setup section.

If the problem is still present, follow this procedure:

1. Open the Setup Network page of the UR robot explained in “The Robot has no IP address”.
2. Modify the Subnet mask to the Hint value provided in the beginning of “Discovery Tool was able to find OnRobot Device available in the Network”, in this example to “255.255.0.0”.
3. Press the Apply button.



Repeat the steps in Example Setup Example Setup.

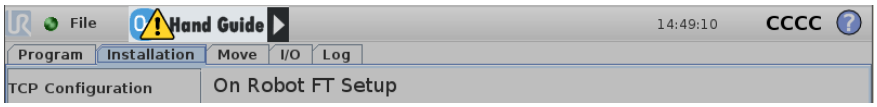
6.5.3 Too Close to Singularity

During hand guiding, if the tool is guided too close to the cylindrical volume directly above or below the robot base, a warning message is displayed.

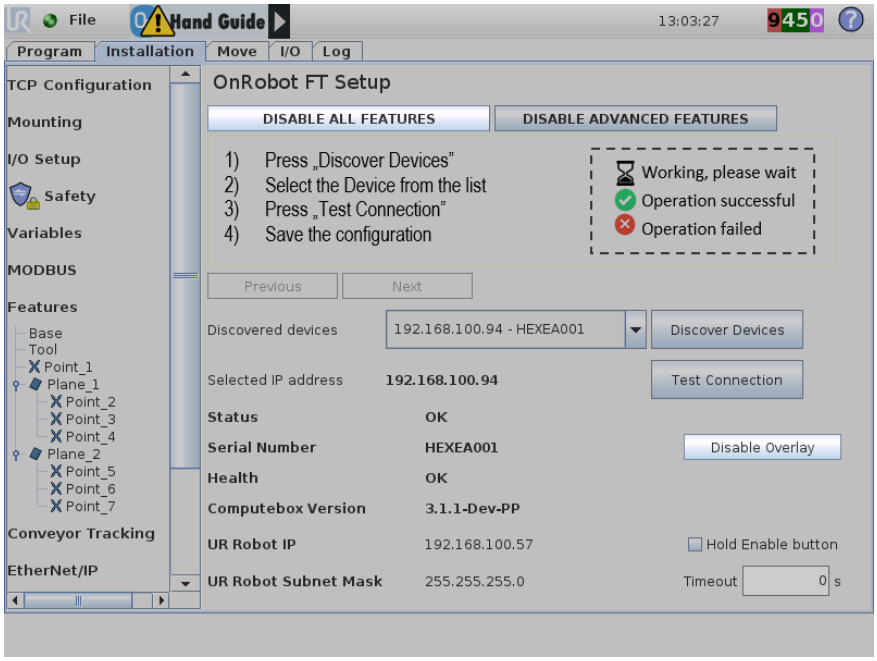
Pressing the **Stop Program** button will disable the Hand Guide function. Pressing the **Continue** button will switch to Safe Mode, which prevents the tool flange from moving into the cylindrical volume directly above or below the robot base by the Hand Guide function. Moving 10 mm away from that volume turns Safe Mode off, enabling movement in all directions again.

> For safety and accuracy, the Hand Guide mode keeps the tool flange at a greater distance from the cylindrical volume than the physical possibility of the UR robot. Moving the tool flange closer is possible, by using the PolyScope Move tab, or move commands.

6.5.4 Warning sign on Hand Guide Bar



If the installed OnRobot Hand Guide Service is not running properly, a warning sign appears (*mark A*). In case this warning appears check the following:



The toolbar warning is visible (*mark 3*), so the OnRobot Service could not be started.

1. Press the **DISABLE ALL FEATURES** button, it changes to **ENABLE ALL FEATURES** and then press it again.
2. Press **Disable Overlay** the button, it changes to **Enable Overlay** button and then press it again to start the OnRobot Hand Guide Toolbar.

If the error persists, uninstall the OnRobot Force Torque Sensor URCap package as described in **Software Uninstallation**, restart the UR Robot, and then reinstall the package, as described in **URCap Package Installation**.

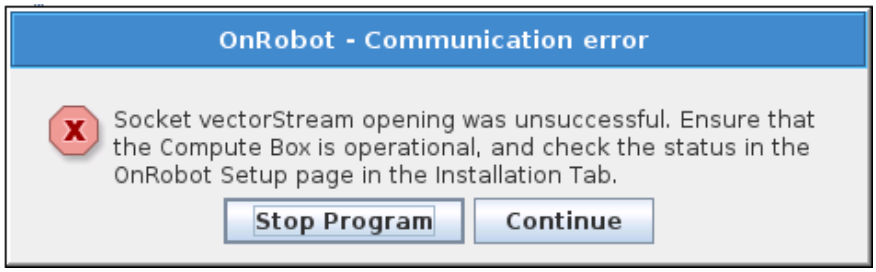
6.5.5 "socket_read_binary_integer: timeout"

If any command is running for more than 2 seconds, a **socket_read_binary_integer: timeout** entry appears in the **Log**.

This has no impact on the program execution by the robot.

6.5.6 "Socket vectorStream opening was unsuccessful."

If the Robot Controller cannot connect to the Compute Box, the "Socket vectorStream opening was unsuccessful." error message is displayed.



In this case, ensure that the Compute Box is connected to the Robot Controller, and powered on.

6.5.7 Path Replay is Slower than Expected

When using the F/T Path command, it is possible that the recorded Path is not smooth due to the limits of human dexterity. In these cases, the robot can only replay the Path at a very slow speed. To avoid this problem, try recording the Path again, with confident smooth movements with as little variations in translational and rotational speeds, as possible. Also, try to avoid recording Paths that contain rotations without translational elements.

6.5.8 "Error number -2" on Path Saving

If an empty Path is recorded, when trying to save the path, the "Error number: -2" error message is displayed.

In this case, ensure that the robot is moved between starting and stopping the path recording function.

6.5.9 "Error number -3" on Path Saving

If a Path cannot be saved due to not enough storage space on the Compute Box, the "Error number -3" error message is displayed.

In this case, delete previously recorded paths that are not used anymore.

6.6 Editions

Edition	Comment
Edition 2	Document restructured. Glossary of Terms added. List of Acronyms added. Appendix added. Target audience added. Intended use added. Copyright, Trademark, contact information, original language information added. Behavior of F/T Move, F/T Search, F/T Insert Pin and F/T Control commands changed. F/T Waypoint command added. F/T Move (Ctrl) command removed. Application example references added to example UR programs.
Edition 3	Hand Guide Toolbar coordinate system corrected to Tool. Note added about TCP Orientation limitation. Hand guiding axis activation limit removed. Clarification about waypoint type usage added.
Edition 4	TCP Orientation limitation removed.
Edition 5	F/T Search and F/T Move command return values updated. Path Recording section removed. F/T Path Command section added. F/T Insert Connector section removed. F/T Insert Connector Return Values section removed. F/T Move Command and F/T Search Command sections updated with constant replay speed information, and new command screenshots. F/T Control Command section updated with directional force control limitation. Editorial changes.
Edition 6	Path replay accuracy added. Section “An error occurred in the running program” on Program Continue changed to “An error occurred in the running program” on Program Stop, pausing and continuing program no longer causes alarm. Section Effects of the TCP Position added. socket_read_byte_list(): timeout log item changed to socket_read_binary_integer: timeout, behavior changed. Section “Socket vectorStream opening was unsuccessful.” added to Troubleshooting.

	Section Connector Insertion removed. Section Path Replay is Slower than Expected added. Limitations added for rotation only waypoints.
Edition 7	Editorial changes.
Edition 8	Path recording maximum rotation per translation limit added to section F/T Path Command. Section “Error number -2” on Path Saving, and “Error number -3” on Path Saving added. Editorial changes.
Edition 9	Important Safety Notice added. Warning Symbols added. Screenshots updated. Note added to warn against rotating the sensor cable in section Cable Connections.
Edition 10	Hex v2 information added.