User Guide for the Universal Robots OptoForce Kit

Version 1.10

March 2017

Content

1 Preface

1.1 Conventions

The following conventions are used in this document.

Table 1: Conventions

`Courier Text`	Used for file paths and file names, code, user input and computer output.
Italicized text	Used for citations and marking image callouts in text.
Bold text	Used to indicate UI elements, including text appearing on buttons and menu options.
<angle brackets>	Indicates variable names that must be substituted by real values or strings.
1. Numbered lists	Numbered list elements indicate steps of a procedure.
A. Alphabetical lists	Alphabetical list elements indicate image callout descriptions.

2 Getting Started

In the OptoForce Universal Robots (UR) Kit everything is provided that is required to connect the OptoForce force/torque sensor to your UR.

For integration, follow this process:

1. Mount the sensor to the robot. For more information, see **Sensor Mounting**.

2. Connect the 4 pin M8 cable (5m long) to the sensor and secure it to the robot with cable ties.

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

3. Place the converter somewhere near or inside the UR robot control cabinet and connect the 4 pin M8 sensor cable. The provided cable gland (size PG16) can be used to lead in/out the cable from the UR Control cabinet.

4. Connect the converter's Ethernet interface with the UR controller's Ethernet interface via the supplied UTP cable (yellow).

5. Use the 3 pin M8 cable (1m long) to power the OptoForce force/torque sensor from the UR's control box. Connect the brown cable to the 24V and the black cable to the 0V.

Power		**Configurable Inputs**				**Configurable Outputs**			
PWR	■	24V	■	24V	■	0V	■	0V	■
GND	■	CI0	■	CI4	■	CO0	■	CO4	■
24V	■	24V	■	24V	■	0V	■	0V	■
0V	■	CI1	■	CI5	■	CO1	■	CO5	■
		24V	■	24V	■	0V	■	0V	■
		CI2	■	CI6	■	CO2	■	CO6	■
		24V	■	24V	■	0V	■	0V	■
		CI3	■	CI7	■	CO3	■	CO7	■

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

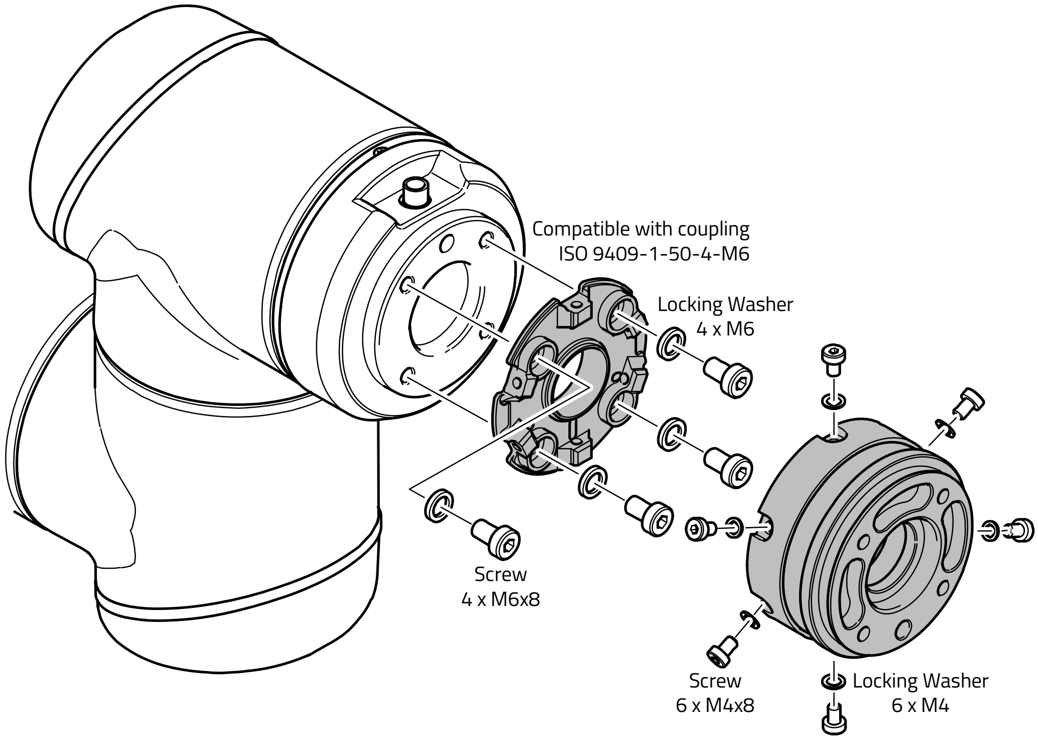
6. Apply the correct network settings to both the Ethernet converter and the UR robot. The default Ethernet converter IP address is 192.168.1.1. If you need to change the IP address of the sensor, see **Changing the IP of the Sensor**.

7. Insert the supplied USB stick into the Teach Pendant, which will automatically copy the examples and the URCaps file to the UR robot. For further information, see **Software Installation**.

3 Sensor Mounting

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

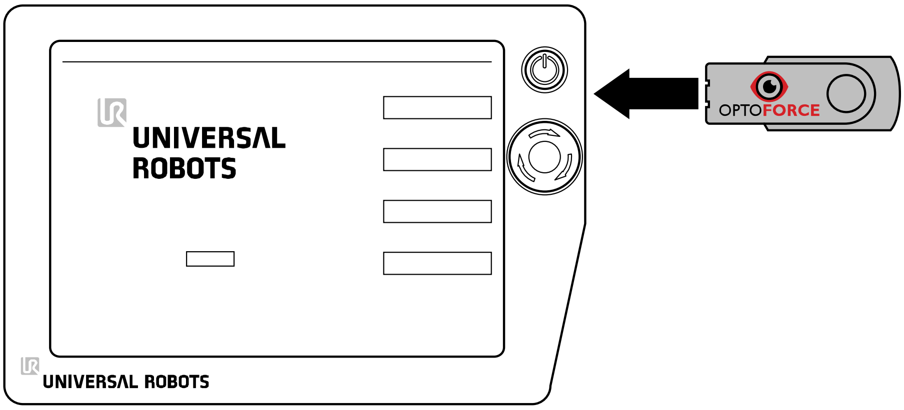
The recommended tightening torque is 2Nm.



4 Software Installation

After the mechanical and electrical installation is finished, insert the USB stick in the USB slot on the right side of the Teach Pendant.

> At least PolyScope version 3.3 is required for URCaps. Check whether the UR supports the URCaps by tapping on the **About** button on the main screen.



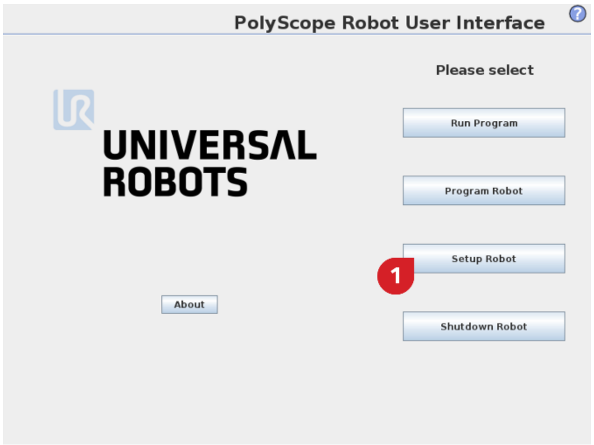
1. A red `"!USB!"` warning sign appears, indicating the upload is in progress.
2. Wait for a green `"USB"` sign to appear, indicating the upload is completed.
3. Remove USB stick. The example codes are now in the program folder of the UR robot.

The URCaps and an example is automatically copied to a new `OptoForce_UR_Programs` folder.

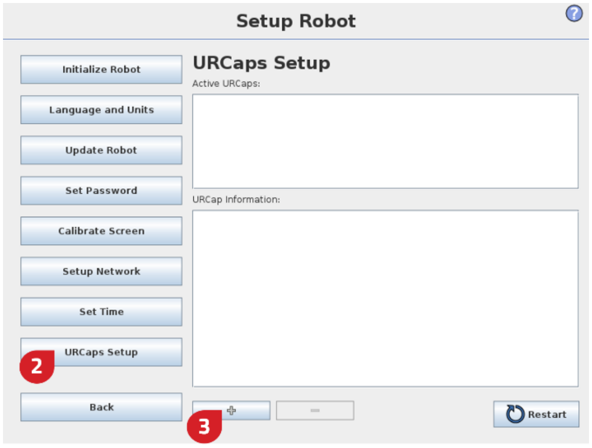
Continue with **URCaps Package Installation**.

4.1 URCaps Package Installation

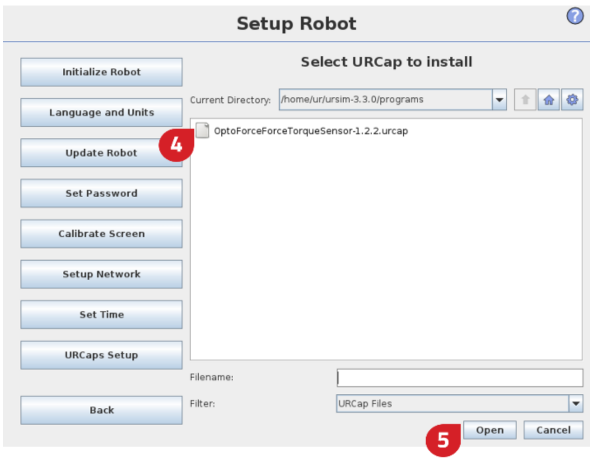
Make sure that PolyScope version of the robot is up-to-date.



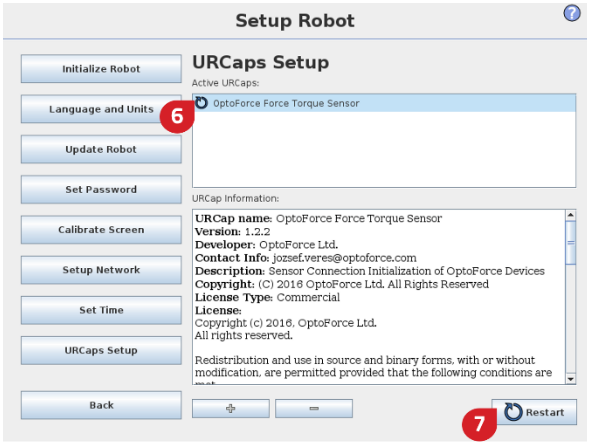
1. Select the **Setup Robot** option from the main menu.



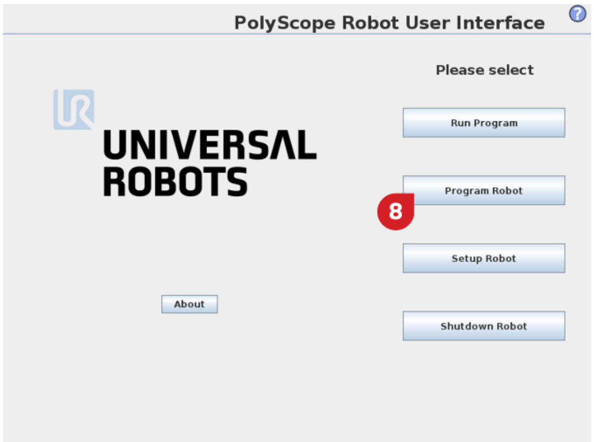
2. Press the **URCaps Setup** button. If this button is not visible, then the UR software version is lower than v3.3.
3. Press the **+** sign to load the appropriate URCAP file.



4. Select the `OptoForceForceTorqueSensor-x.x.x.urcap` file from the `OptoForce_UR_programs` folder.
5. Press the **Open** button.



6. The screen shows that the OptoForce Force Torque Sensor URCaps has been installed.
7. Press the **Restart** button to finalize installation process.



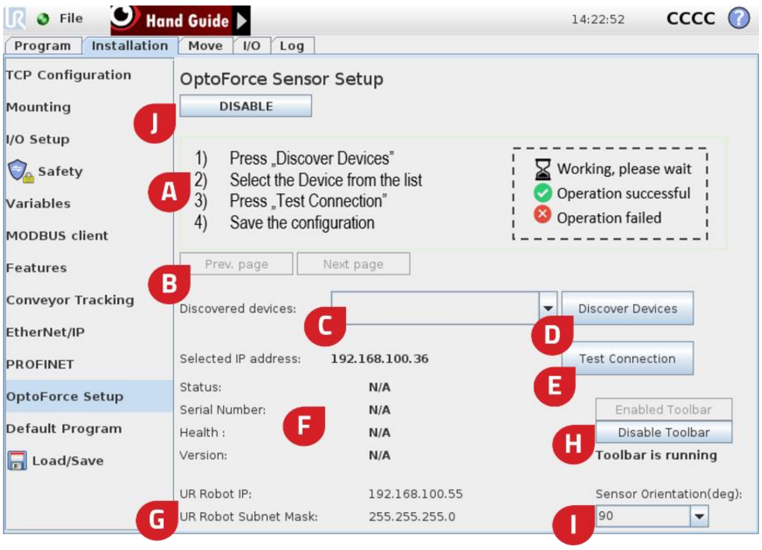
8. Click on the **Program Robot** button after the restart is finished.

9. Select the **Installation** tab.
10. Select **OptoForce Setup**.
11. The welcome screen of the newly installed OptoForce Force Torque Sensor URCaps is shown.
12. Note, that the OptoForce Hand Guide Toolbar is now also visible.

See **URCaps Package Setup**, on how to setup the OptoForce Force Torque Sensor URCaps.

4.2 URCaps Package Setup

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



A. This area shows information about the steps needed for the sensor setup.

B. Navigation buttons to move between pages of the guide.

C. 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.

D. Connected sensors can be discovered by pressing the **Discover Devices** button.

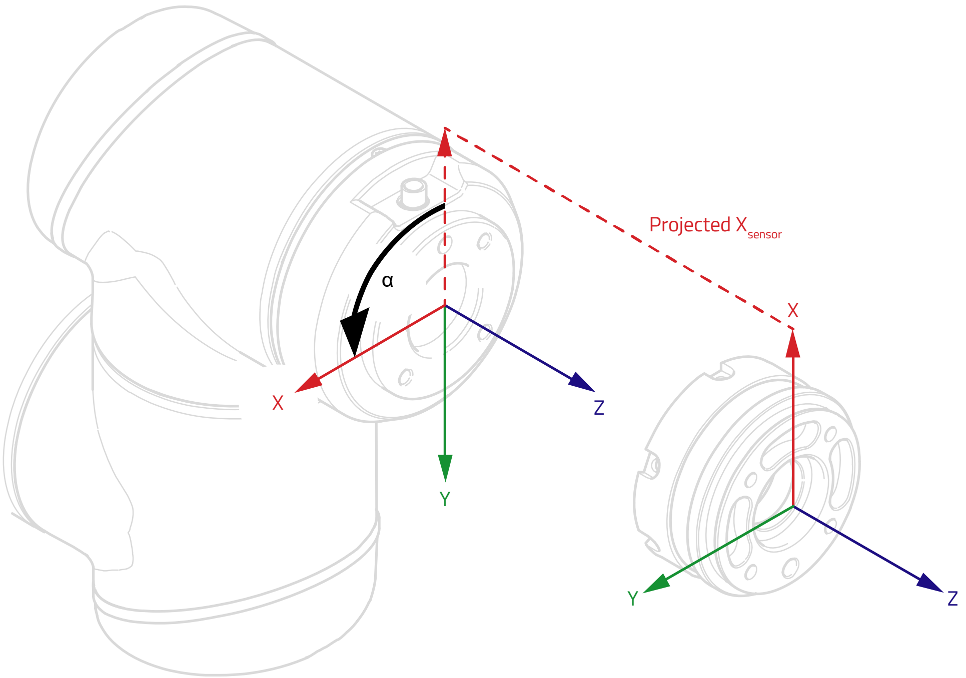
E. The selected OptoForce device must be tested first.

F. General information about the tested device.

G. IP and Subnet mask of the currently used UR robot.

H. Enable/Disable buttons for the OptoForce Hand Guide Toolbar.

I. The rotation of the mounted OptoForce Sensor (α).

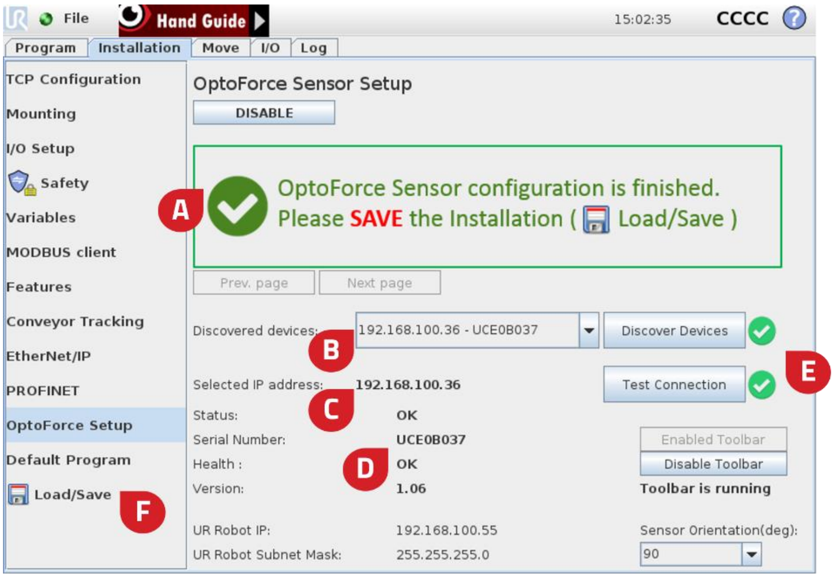


J. Enables or disables the OptoForce URCaps. If disabled, the Hand Guide Toolbar and all other functionalities are suspended.

4.2.1 Example Setup

1. Press **Discover Devices**.
2. Select the device you would like to use.
3. Press **Test Connection**.

If the setup is successful, the following screen is shown:



A. If the procedure is successful, this message is shown. It indicates that the OptoForce device can be used in your own UR programs.

B. A dropdown-list of the IP addresses and the Serial Numbers of the available OptoForce devices.

C. In this example, the Device with serial number UCE0A053 was selected as the IP address is matched to UCE0A053.

D. After pressing the **Test Connection** button, basic information is shown about the device:

   a. the status of overall process

   b. the serial number

   c. the Health String of the device

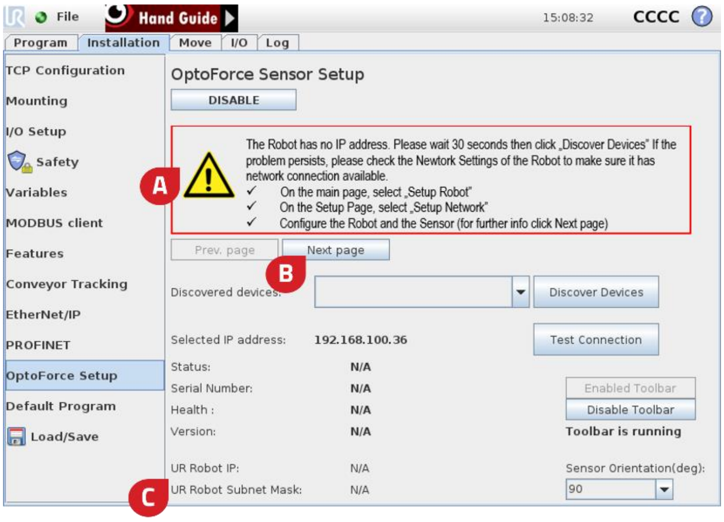
   d. the version of the converter

E. The green tick marks indicate that both the **Discover Devices** and **Test Connection** procedures were successful.

F. You can save the current configuration by pressing **Load/Save**.

4.2.2 Common Errors

4.2.2.1 "The Robot has no IP address"



A. The quick guide provides a possible solution to this problem.

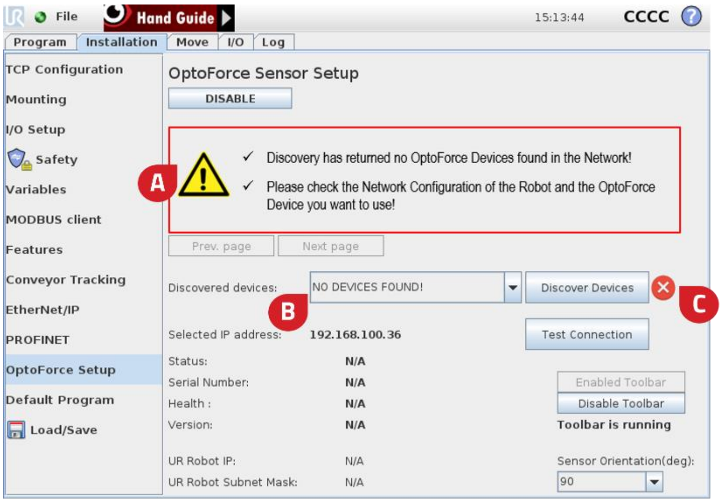
B. This guide has more than one page. Change pages with the Prev. page and Next page buttons.

C. 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 **OptoForce Setup** page just after turning on the robot or when the Network Configuration has just been modified and the **OptoForce Setup** page is selected within 30 seconds.

In these cases, wait 30 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:

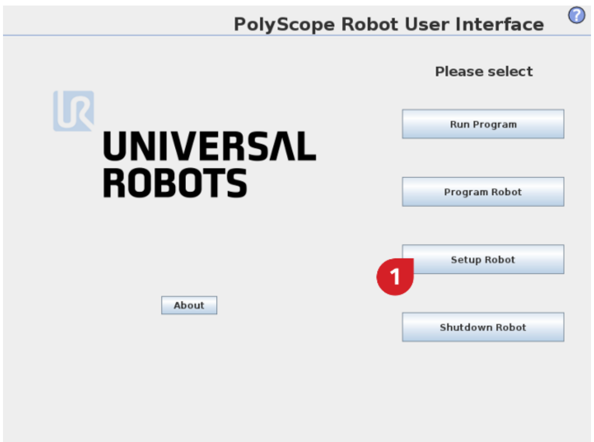


A. The error description.

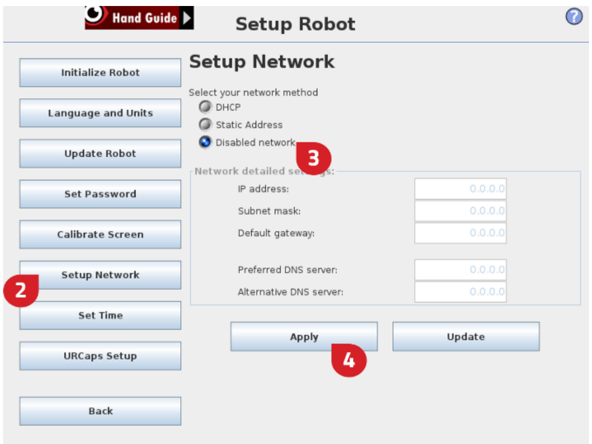
B. "`NO DEVICES FOUND!`" in the device selection list.

C. 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:



1. Press the **Setup Robot** button.



2. Press the **Setup Network** button.
3. If the network of the UR is disabled:
   a. If the OptoForce device is connected to the UR robot directly, select **DHCP**, and press the **Apply** button *(mark 4)*. The OptoForce sevice assigns an IP.
   b. If the OptoForce device is not directly connected to the UR robot, check if the OptoForce device is connected to the same network

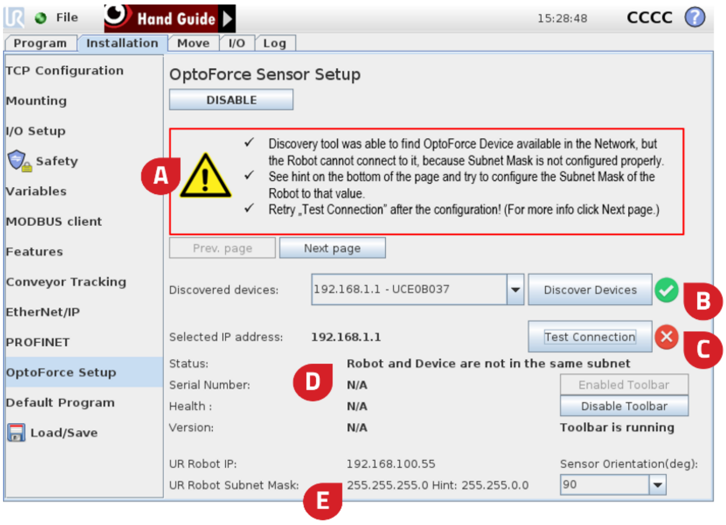
(router, switch, and so on) as the UR Robot, or consult the Network Supervisor.

If **DHCP** or **Static Address** is selected, 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.

> In some older models the DHCP IP reassignment process could happen every hour, interrupting the normal operation of the sensor.

4.2.2.2 "Discovery Tool was able to find OptoForce 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.

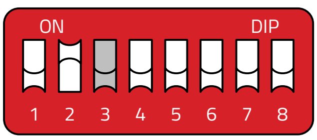
D. "`Robot and Device are not in the same subnet`" status message is shown.

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

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

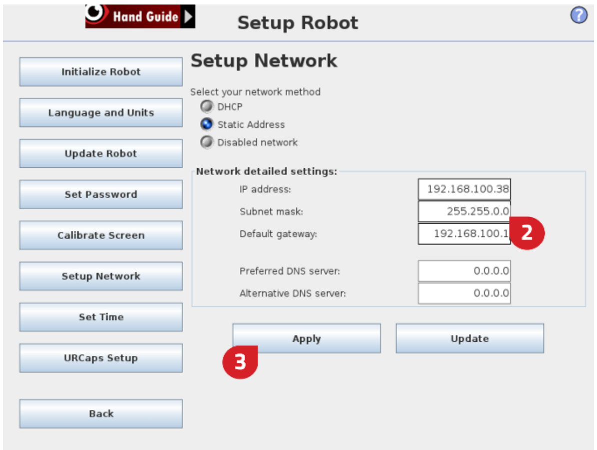
To resolve the problem, follow this procedure:

- If the OptoForce device is not directly connected to the UR Robot, check if the DIP Switch 3 is in off state on the converter, as shown on the following figure:



- If the DIP Switch is in on state, set it to off, then restart the OptoForce device 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 OptoForce Device available in the Network"**, in this example to "255.255.0.0".
3. Press the **Apply** button.

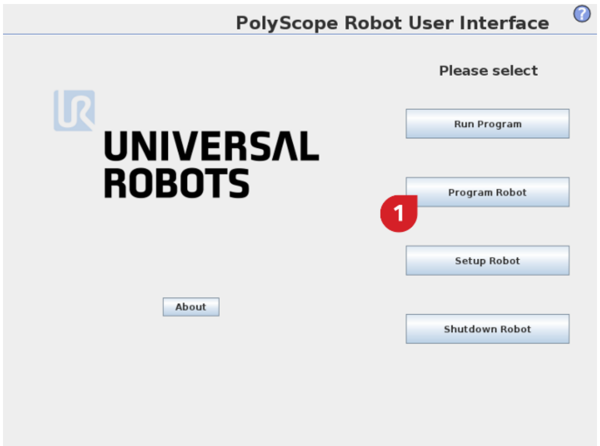
Repeat the steps in **Example Setup**.

5 Using the URCaps Package

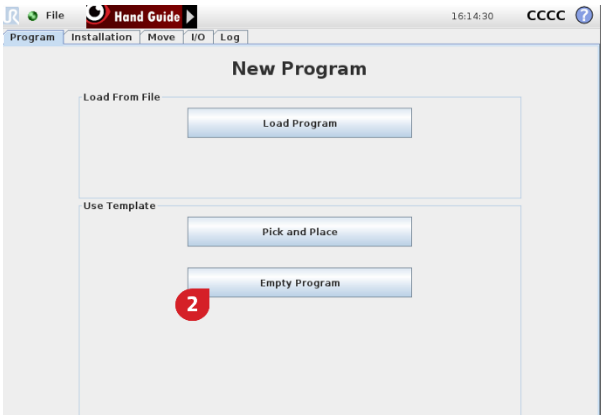
5.1 Simple features

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

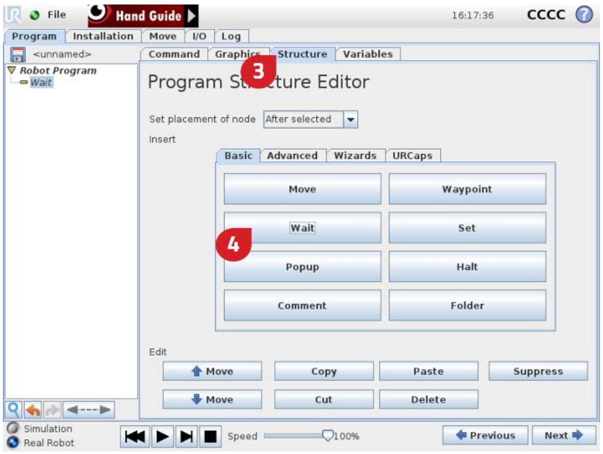
> It is assumed that the OptoForce Sensor is set up as described in **Example Setup**, and you have not disabled the package using the **Enable/Disable** button described in **URCaps Package Setup** *(mark J)*.



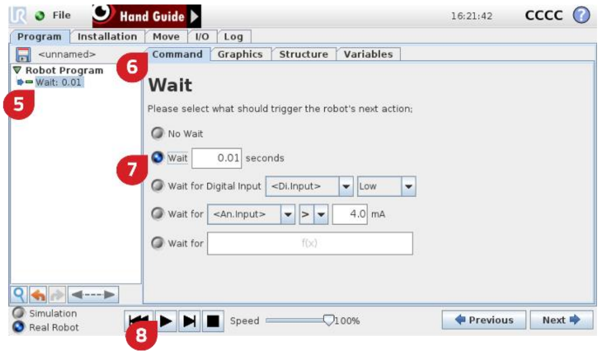
1. Click on **Program Robot**.



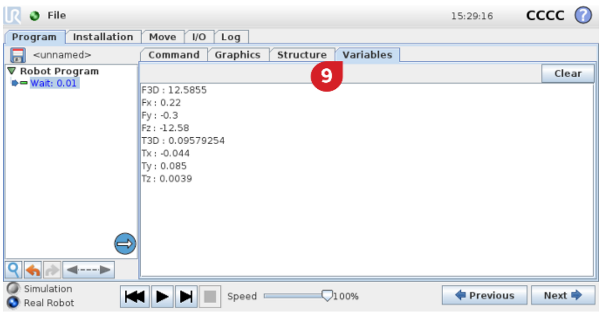
2. Click on **Empty Program**.



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



5. Select the `Wait` structure under `Robot Program` folder.
6. Select the **Command** tab of `Wait` structure.
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 access these variables in your own program.

These variables are updated automatically at a rate of approximately 125Hz (according to the sensor coordinate system):

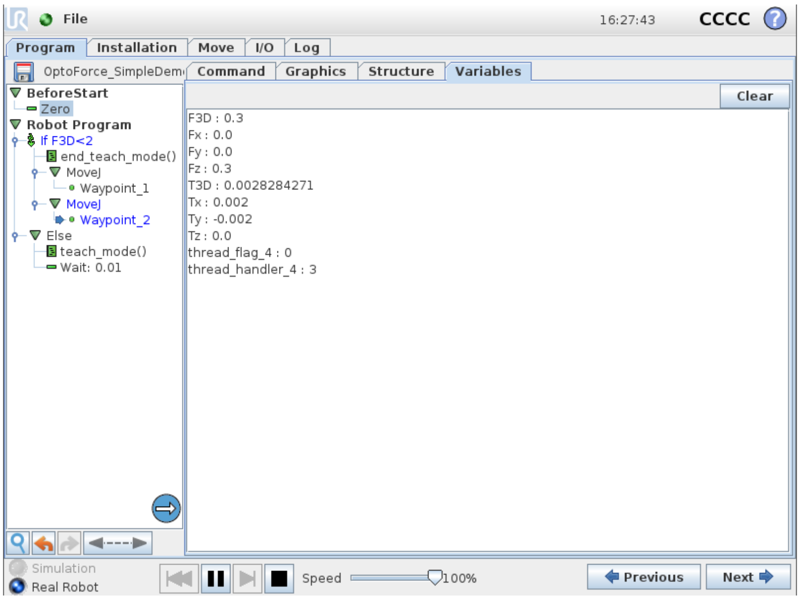
- **Fx**: Force vector in the X direction in Newton (N)
- **Fy**: Force vector in the Y direction in Newton (N)
- **Fz**: Force vector in the Z direction in Newton (N)
- **Tx**: Torque in the X direction in Newton Meter (Nm)
- **Ty**: Torque in the Y direction in Newton Meter (Nm)
- **Tz**: Torque in the Z direction in Newton Meter (Nm)
- **F3D**: Length of the 3D force vector F3D = sqrt $(Fx^2 + Fy^2 + Fz^2)$

- **T3D**: Length of the 3D torque vector T3D = sqrt (Tx^2 +Ty^2 +Tz^2)

5.1.1 Simple Demo as an Example

`OptoForce_SimpleDemo.urp`

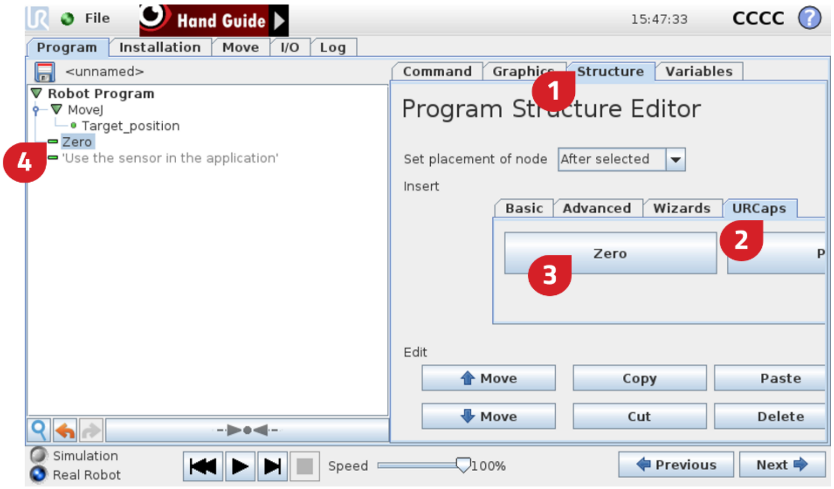
In this example the robot is navigated between two waypoints, `Waypoint_1` and `Waypoint_2`. If the force is above a threshold limit (2N), the robot motion is stopped and the robot mode is set to tech mode. The `Zero` command at the `Before Start` section is used to make the sensor reading zero.



5.1.2 Using the `Zero` Command in a Program

Expand the program created in **Simple features** with the `Zero` command of the OptoForce Force Torque Sensor URCap. This structure can zero the OptoForce Sensor's force and torque values.

Follow this procedure to add the command:



1. Select the **Structure** tab.

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

3. Press the **Zero** button.

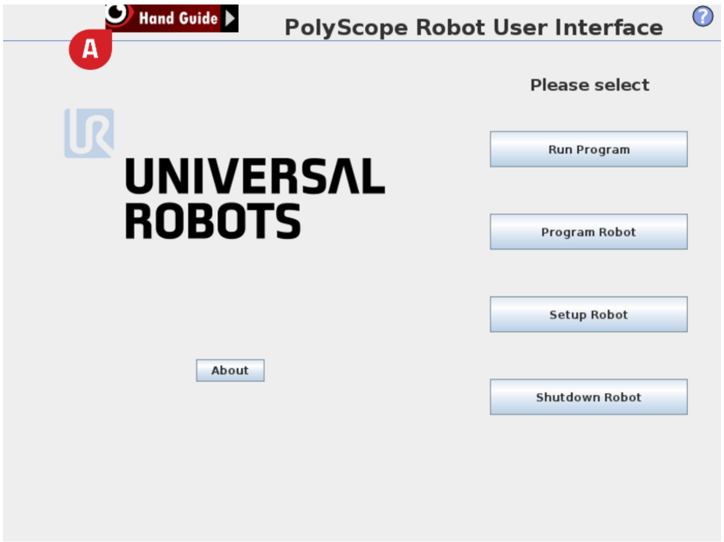
4. The `Zero` command appears under the `Robot Program` node *(mark D)*.

The `Zero` structure can only be used when the OptoForce Force Torque Sensor URCap is enabled (**URCaps Package Setup**, *mark J*) and the OptoForce sensor has been configured correctly as described in **Example Setup**.

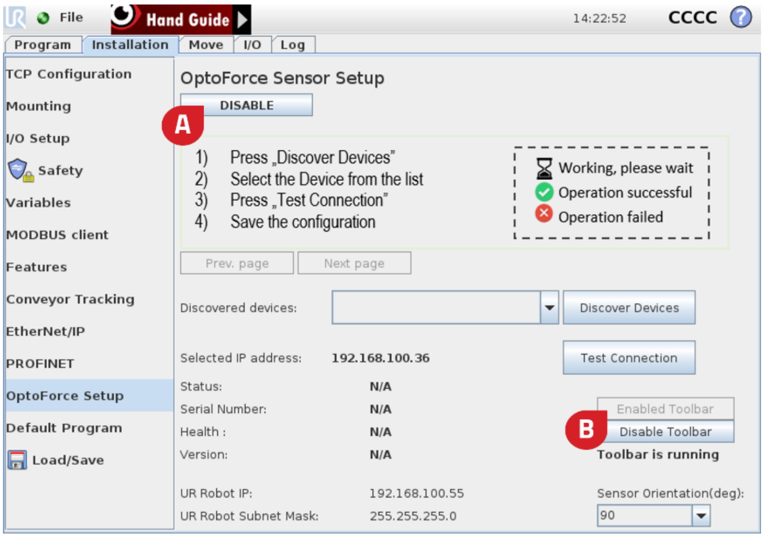
5.2 Hand Guide

> It is assumed that the OptoForce Sensor is set up as described in **Example Setup**, and you have not disabled the package using the **ENABLE/DISABLE** button described in **URCaps Package 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 OptoForce Hand Guide Toolbar appears *(mark A)*. It is normal to have a yellow warning signal during the boot-up for a few seconds.

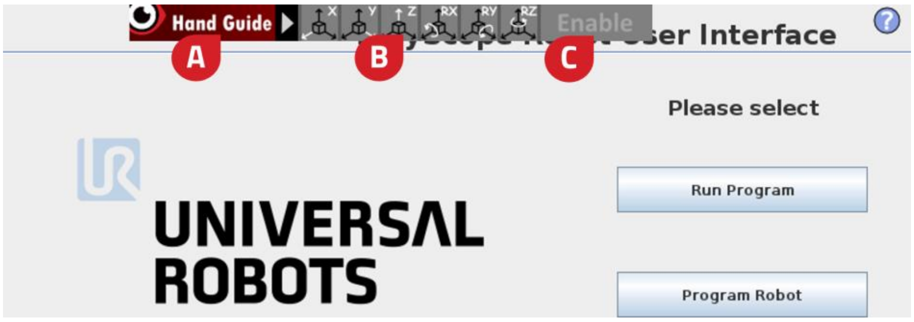


If the Toolbar does not appear, on the **Installation** page, select **OptoForce Setup** and check the following:



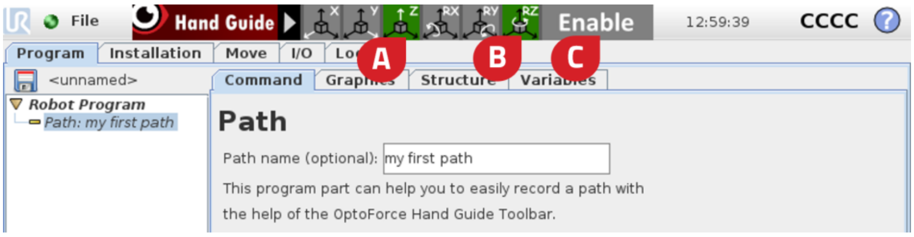
The OptoForce Force Torque Sensor URCap package must be enabled (the screen above is shown). If the package is not enabled, the button shown under *mark A* has the text **ENABLE**. Press the button, then start the toolbar with the **Enable Toolbar** button (*mark B*).

5.2.1 Functions of the Toolbar

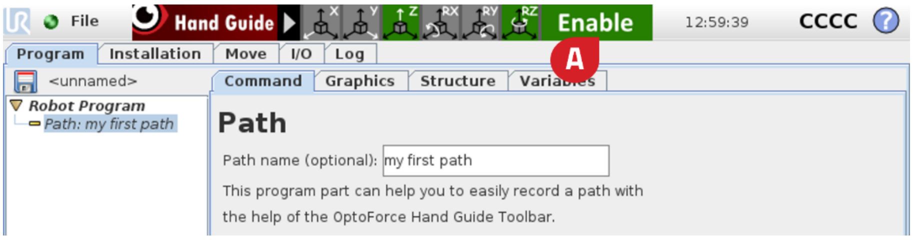


To activate the functions of the toolbar, press on any point of the surface under *mark A*. The Toolbar expands, and the available directions/rotations *(mark B)* and the **Enable** button *(mark C)* will appears as shown on the image above.

To select a rotation or a direction, press the appropriate item. In the following example the Z *(mark A)* and the RZ *(mark B)* items are selected. Note that the **Enable** button *(mark C)* is now enabled (the button text is white).



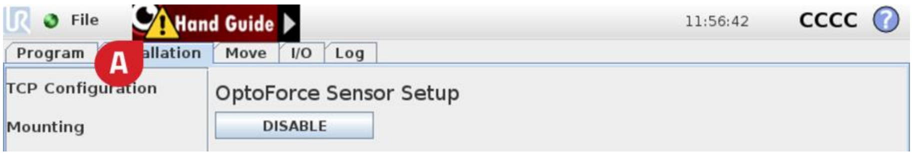
To deactivate any direction/rotation that has been selected, press them again. If no directions and rotations are selected, the **Enable** button is deactivated.



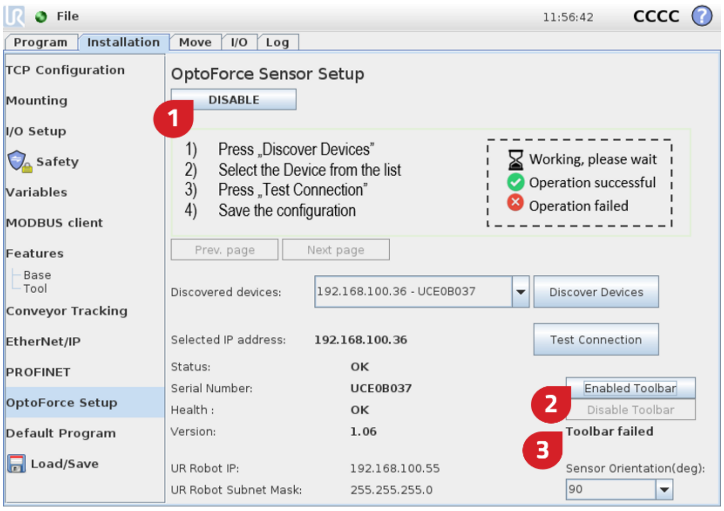
To start hand guiding the UR robot, press and hold the **Enable** button, and drive the robot by hand with the help of the OptoForce sensor. Note that the **Enable** button *(mark A*) has changed (background color is now green).

To stop hand guiding of the UR Robot, release the **Enable** button.

5.2.2 Troubleshooting



If the installed OptoForce 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 and the toolbar has failed (*mark 3*), so the OptoForce Service could not be started.

1. Press the **DISABLE** button, and then press it again.
2. Press the **Enable Toolbar** button to start OptoForce Hand Guide Toolbar.

If the error persists, uninstall the OptoForce Force Torque Sensor URCAP package, restart the UR Robot, and then reinstall the package.

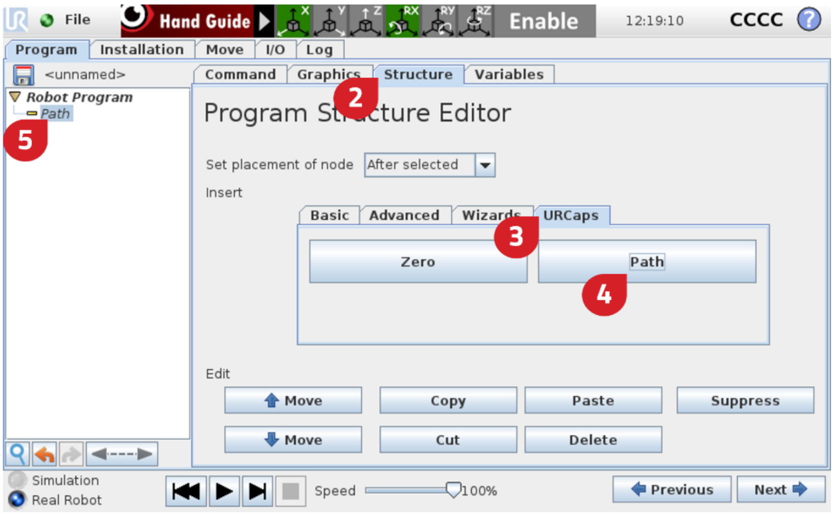
5.3 Path Recording

> It is assumed that the OptoForce Sensor is set up as described in **Example Setup**, and you have not disabled the package using the **ENABLE/DISABLE** button described in **URCaps Package Setup** *(mark J)*.

With the OptoForce Hand Guide Toolbar and OptoForce Path Recording, the path of the UR robot can be recorded, replayed, and used in your own UR robot program.

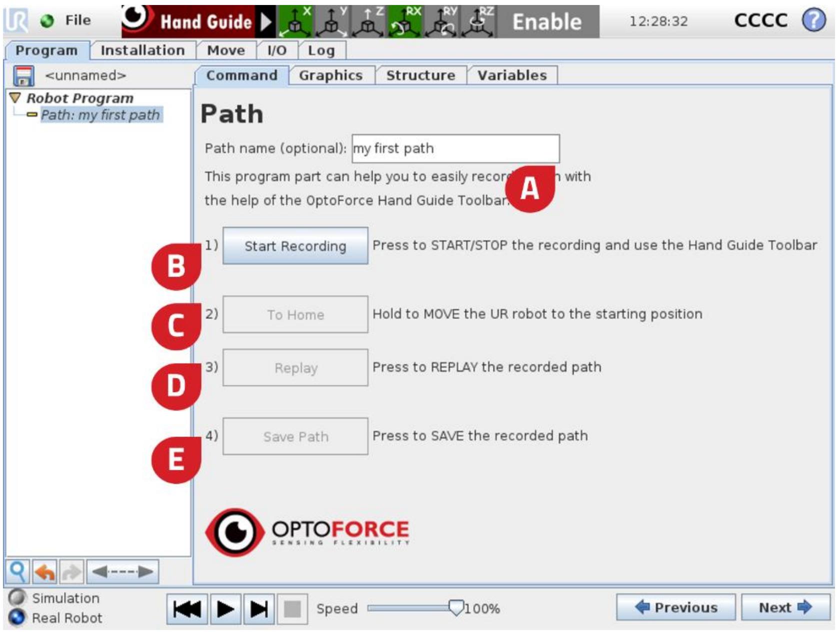
5.3.1 Creating a Path

To use the OptoForce Path Recording in your program, follow these steps:



1. Create an empty program.
2. Select the **Structure** tab.
3. Select the **URCaps** tab.
4. Select the **Path** button.
5. The `Path` command appears under the `Robot Program` node.

After the `Path` command is added to your program, go to the **Command** tab of the `Path` item. The following information is displayed:



A. The optional name of the `Path`. This is useful when you have more than one recorded path in your program. The name of the path appears on the node (*mark F).*

B. The **Start/Stop Recording** button. You can start or stop the path recoding using this button.

C. The **To Home** button. After the path recording is finished, and the starting position is not the same as the stopping position, holding this button moves the UR robot to the starting position. When you release this button, the UR robot stops.

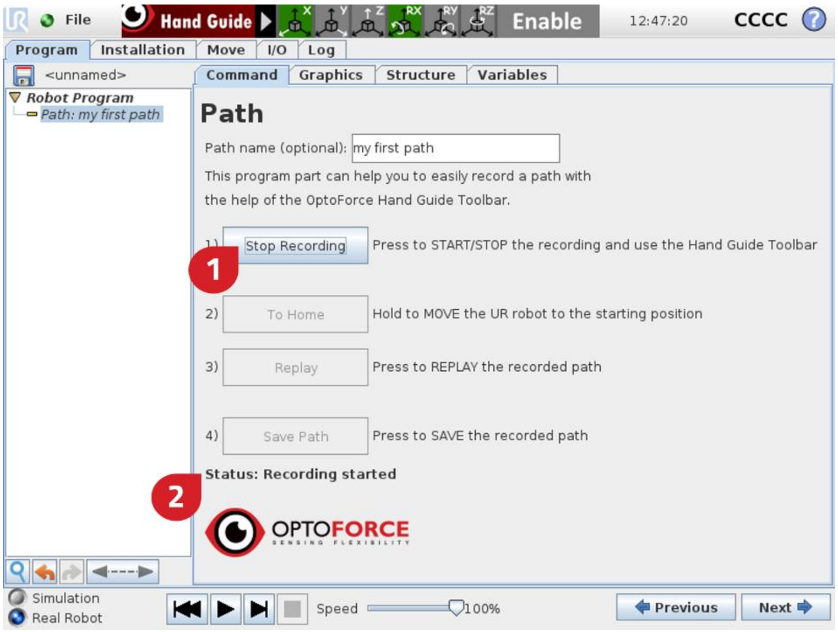
D. With the **Replay** button the recorded path replay can be started or stopped.

> The recorded path will not be saved with the use of the **Replay** button.

E. Pressing the **Save Path** button saves the recorded path into your UR program.

5.3.2 Recording a Path

To record a path, follow these steps:

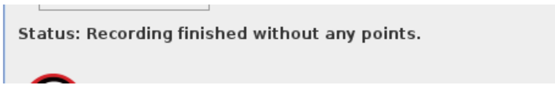


1. Press the **Start Recording** button.
   The caption of the button changes to **Stop Recording**. Pressing this button again stops the path recording.

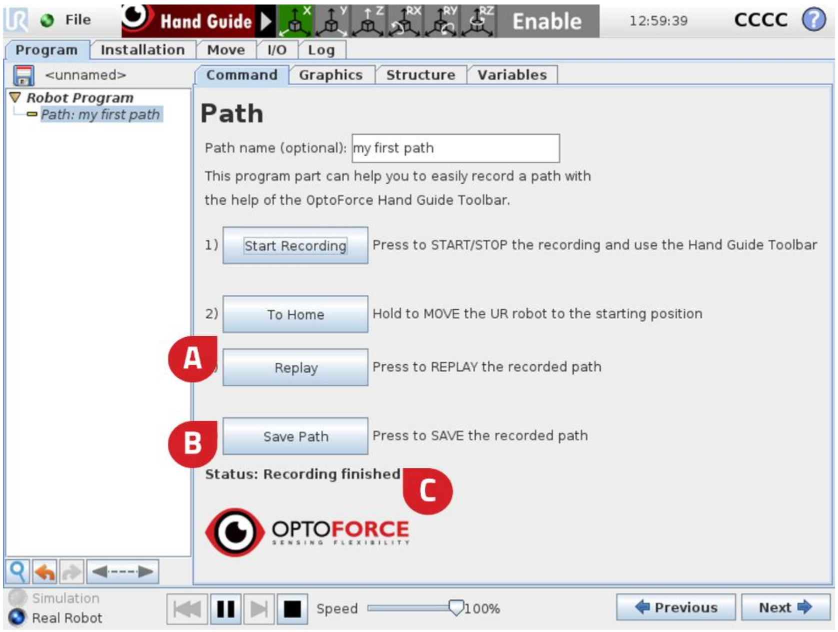
2. The status of the path recording is displayed.

3. Select the desired directions on the Hand Guide Toolbar and move the UR robot as described in **Hand Guide**.

While you are holding the **Enable** button of the Toolbar, poses will be recorded. If there are no recorded points or you did not move the robot, the status bar will warn you:



If the recording is successful, the following screen is shown:



The **To Home**, **Replay** (*mark A*) and **Save Path** (*mark B*) buttons are enabled and the status bar displays the `Recording finished` status *(mark C)*.

5.3.3 Finalize the Recorded Path

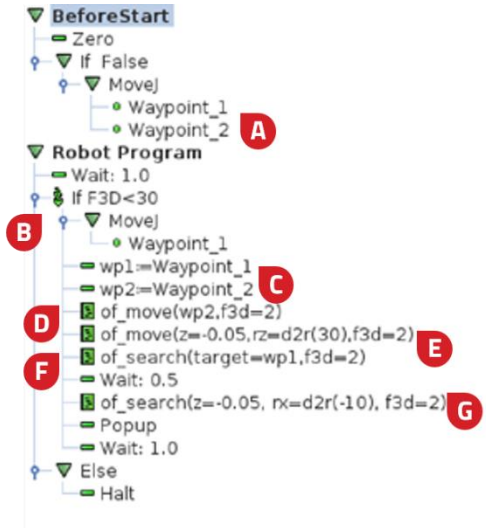
To finalize the recorded path, follow this process:

1. Press the **Save Path** button. Note that the status bar displays the `Path saved` message *(mark 3)* and the program item status changes *(mark 4)*.
2. Press the **Replay** button to run the program which will replay the recorded path.

The recorded path can then be saved as a UR program, and can be run any time as a usual UR program.

5.4 Collision Detection

The OptoForce force control library extends the standard robot movement command with collision detection features.



A. Waypoint definitions

B. Move to `wp1` position (UR)

C. Variables are created from waypoints

D. Move to `wp2` position, but stop if 1N collision force is detected

E. Relative linear movement and rotation with 2N collision force limit

F. Searching 1N contact force in the direction of `wp2`

G. Searching 2N contact force along the –Y axis with maximum 1 meter movement limit

The `of_move` function can be used to move the robot to a waypoint (as a variable), or to move the robot relative to the current pose. Collision force limits can be defined for every movement. If a collision happens, a pop-up warning is displayed and options are provided to exit the program or continue with the next command.

The `of_search` function can be used to search for collision in a certain direction. Absolute or relative coordinates can be specified. The speed of the movement and the deceleration during collision can also be specified, so the robot does not break the workpiece.

5.4.1 of_move(...)

```
of_move(target, x, y, z, rx, ry, rz, fx, fy, fz, tx, ty, tz,
f3d, t3d,
```

```
... speed=0.05, acc=1.2, brake=1.2, silent=False)
```

This function moves the robot from the actual to the desired pose. If the sensor detects an obstacle, there is larger force or torque value, then the limit values it will stop the movement.

Arguments:

- `target`: The target pose, where the robot needs to move to.
- `x,y,z,rx,ry,rz`: Distance of relative movement (from current pose). Either the `target` parameter or these relative movement commands can be used, but not both.
- `fx, fy, fz, tx, ty, tz, f3d, t3d`: The force and the torque limits set to detect collision (N, Nm).
- `speed`: Tool speed. (m/s, rad/s).
- `acc`: Tool acceleration. (m/s^2,rad/s^2).
- `brake`: Tool deceleration during collision (m/s^2,rad/s^2).
- `silent`:
    - `False` (default): Po-pup messages (blocking) will appear if exception occurs.
    - `True`: Pop-ups disabled. You can monitor the success/failure of the function from return values only. Suitable for embedded function calls.

Return value:

- 0: The move could not be started because of the force/torque limits.
- 1: Move completed without collision. (Normal exit state.)
- 2: The move has been interrupted with collision.

Examples:

- `of_move(target=waypoint1, fx=5)`

This function moves the robot from the actual position to the target pose, unless there is higher force value than the 5 Newton in the X direction.

- `of_move(target_pose, fy=5, t3d=10, speed=0.2, acc=1, brake=1.5, silent=True)`

  This function moves the robot from the actual position to the target pose, unless there is higher force value than the 5 Newton in the Y direction or bigger 3D torque value than 10 Newton-meter. The robot moves faster than the default speed with smaller acceleration, but brakes harder. The silent parameter is set to `True`, so there are no pop-up messages if there is an error during the movement.

5.4.2 of_search(...)

```
of_search(target, x, y, z, rx, ry, rz,fx, fy, fz, tx, ty, tz,
f3d, t3d, ... speed=0.03, acc=1.2, brake=1.2, silent=False)
```

This function moves the robot from the actual to the desired pose during force control mode. The robot looks for collision and it stops if there was larger force or torque value than the given limit values. If the robot reaches the desired pose without collision, the search was unsuccessful.

This function can be used to search for a contact with a tool or reference point.

Arguments:

- `target`: The target pose, where the robot needs to move to.
- `x,y,z,rx,ry,rz`: Distance of relative movement (from current pose). Either the target parameter or these relative movement commands can be used, but not both of them.
- `fx, fy, fz, tx, ty, tz, f3d, t3d`: The force and the torque limits. (N, Nm)
- `speed`: Tool speed. (m/s, rad/s)
- `acc`: Tool acceleration. (m/s^2,rad/s^2)
- `brake`: Tool deceleration during collision. (m/s^2,rad/s^2)
- `silent`:
  - `False` (default): Pop-up messages (blocking) will appear if exception occurs.

- o `True`: Pop-ups disabled. You can monitor the success/failure of the function from return values only. Suitable for embedded function calls.

Return value:

- 0: The move could not be started because of the force/torque limits.
- 1: The search was unsuccessful. Movement reached the distance limit.
- 2: The move has been interrupted with collision. (Normal exit state.)

Examples:

- `of_search(y=0.1, fz=5)`

  This function gives the direction of the search, so the robot moves 0.1 m to the Y direction. This function exits with value 2 when a higher torque value than the given 5 Newton-meter in the Z axis is reached.

- `of_search(y=0.1, fx=5, tx=10, t3d=10, speed=0.05, acc=0.1)`

  This function gives the direction of the search, so the robot moves 0.1 m to the Y direction from the actual pose. This function exits with value 2 when a higher force value than the 5 Newtons in the X direction or a higher torque value than 10 Newton-meter in the X axis is reached. The robot moves slower than the default speed with smaller acceleration.

5.5 Centerpointing

Centerpointing can be used to align the robot's TCP to some reference object.

A. Move to start position (UR)

B. Find the center point along the X axis with maximum 1 meter search and 2N contact force

C. Find the cent point along the Y axis with maximum 1 meter search and 2N contact force

D. Find the center point along RZ axis with maximum 15 degrees rotation and 0.2 Nm contact torque

With the `of_center` function the center point along one of the six base exes can be found. The axis, the distance along it, and the collision forces need to be defined. The speed and acceleration of the robot can also be defined.

5.5.1 of_center(...)

```
of_center(x, y, z, rx, ry, rz,  fx, fy, fz, tx, ty, tz, f3d,
t3d,

speed=0.03, speedCP=speed, acc=0.2, brake=1.2,
mb_distance=0.01, silent=False)
```

The function moves the robot along the given axis until it finds an obstacle. After the collision, it moves to the opposite direction until another collision happens. After that the robot calculates the center of the two boundary points and moves to it.

Arguments:

- `x,y,z,rx,ry,rz`: The axis of centerpointing. There must be only one axis defined.
- `fx, fy, fz, tx, ty, tz, f3d, t3d`: The force and the torque limits (N, Nm).
- `speed`: Tool speed (m/s, rad/s).
- `speedCP`: Tool speed when moving to the center point. (m/s, rad/s).
- `acc`: Tool acceleration. (m/s^2,rad/s^2).
- `brake`: Tool deceleration during the collision. (m/s^2,rad/s^2).
- `mb_distance`: The distance the robot needs to move back after the collision to clear the collision force (m, rad).
- `silent`:
    - `False` (default): Pop-up messages (blocking) will appear if exception occurs.

  - `True`: Pop-ups disabled. You can monitor the success/failure of the function from return values only. Suitable for embedded function calls.

Return value:

- 0: Arrived successfully to the center point.
- 1: The 1st boundary search was unsuccessful. Movement reached the distance limit.
- 2: The 2nd boundary search was unsuccessful. Movement reached the distance limit.
- 3: Could not reach the center point. The tool collided during the movement.

Examples

- `of_center(y=0.5, f3d=10)`

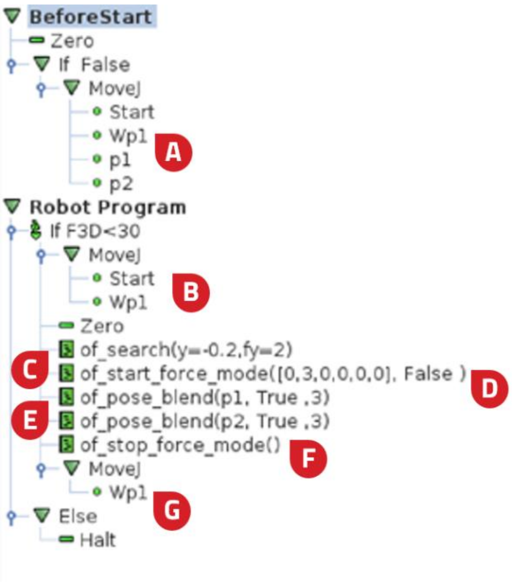
  This function moves the robot along the axis until it finds an obstacle. If the distance of the movement is larger than 0.5 meter, the robot stops. The limit of the 3D collision force is 10 Newtons.

- `of_center(x=0.7, f3d=5, t3d=5, speed=0.1, speedCP=0.3, acc=0.4, mb_distance=0.01)`

  This function moves the robot along the X axis until it finds an obstacle. If the distance of the movement larger than 0.7 meter, the robot stops with an error message. The limit of the collision detection is 5 Newtons and 5 Newton-meters. The speed when the robot searches the boundary point is slower than the default, but it is faster during movement to the center point. The acceleration is higher than the default. Because of the `mb_distance` variable, the robot moves back 0.01 meters (1 cm) after the collision to clear the collision force.

5.6 Plastic Parting Line Removal

Plastic parting line removal is a subset of polishing tasks, where only movements along lines are required. Surfaces to be polished cannot be defined in this release.



A. Waypoint definitions

B. Move to start position

C. Contact the workpiece to the polishing disk

D. Hold constant 4N force in the Y direction…

E. …while the workpiece is moved to the `p1` point

F. Stop force control mode

G. Move to start position again

The `of_start_force_mode` function initializes force mode. A 6D force/-torque vector is defined to be held until the `of_stop_force_mode` function exits force mode. During force mode, the `of_pose_blend` function can be used to move the robot's TCP linearly the same way as the `movel` function does. This function can be used to move the robot or use `Wait` commands to hold the force.

> UR movement commands cannot be executed until the `of_stop_force_mode` function is used.

5.6.1 Holding Constant Force and Torque

The main purpose of the force control script is to provide easy-to-use functions to application programmers, who want to develop force-controlled applications such as assembly, polishing/sanding. A large subset of these applications may require holding constant force to a defined direction during movements. Algorithms dealing with these type of problems are described in this section.

5.6.1.1 Coordinate Systems

Most of the force-controlled applications have to hold a constant force toward some well-defined direction in reference to the robot's global coordinate system.

The robot can be controlled in two coordinate systems:

- Tool coordinate system: The tool coordinate system is referenced to the robot's TCP (Tool Center Point). When the robot moves, this coordinate system moves and rotates with it.

  Holding forces to a direction defined in this coordinate system is error prone, because if the TCP is rotated, the forces and movements represented in this coordinate system would change direction. The user has to continuously observe the TCP's orientation as opposed to the application's global coordinates.

- Base coordinate system: The base coordinate system is fixed. It is referenced to the robot's mounting base. If the robot is mounted on a horizontal surface, the Z axis points upwards and the Y axis points to the direction of the robot's power cable. The coordinate system is based on the right-hand rule.

  This coordinate system is more suitable for force-controlled application programming, because it does not move or rotate during the robot's movement.

> The scripts use the Base coordinate system as default.

5.6.1.2 The PID controller

A standard PID (Proportional-Integral-Derivative) controller loop is implemented in this script. The controller coordinates all the axes of the robot during movement. Each degree of freedom can be force-controlled or position-controlled. Force-controlled axes will hold the defined force, position-controlled axes will hold (or move to) a defined position.

The PID controller moves the robot from a separate thread, the `of_pid_thread()`. Due to the control method of UR robots, only one controlling thread can exist at a time. Concurrency is not allowed. So while the `of_pid_thread()` is running, no UR-based control scripts (`movel`, `movej`, `speedl`, and so on) can be executed.

Only the movement functions defined in this section can be executed during force control.

Global parameters of the controller:

- `of_pid_thrd`: The thread ID of the running `of_pid_thread()`.
- `of_kill_pid`: The `of_pid_thread()` can be shut down using this variable. This parameter cannot be used directly. The `of_start_pid()` and `of_stop_pid()` handles this.
- `of_disorient`: Indicates whether the orientation calculation is enabled for the PID controller. If this parameter is set to `false`, orientation calculation is enabled. Calculating the orientation error is time consuming. The `of_POS_ERROR` is updated at a 25Hz rate only. If this parameter is true, the orientation calculation is disabled, the `of_POS_ERROR` carries only position errors (first 3 values) and the orientation error is always 0.

  This parameter cannot be used directly. The `of_start_pid()` sets this variable when you start the force control.
- `of_PID_TARGET`: The force target for the controller. It is first defined with the first parameter of `of_start_pid()`.
- `of_POS_TARGET`: The position target for the controller. The position-controlled axes will be set to this position continuously by the PID controller.

  This parameter cannot be used directly. OptoForce movement commands defined command the robot.
- `of_POS_ERROR`: The position error of the position-controlled axes. This is the difference between the `of_POS_TARGET` and the robot's current position returned by `get_actual_tcp_pose()`. The `of_pos_error_calc_thread()` updates this variable continuously.
- `of_FPID_P`, `of_FPID_I`, `of_FPID_D`, `of_FPID_ILength`: The PID parameters for the force-controlled axes.
- `of_PPID_P`, `of_PPID_I`, `of_PPID_D`, `of_PPID_ILength`: The PID parameters for the position-controlled axes.

5.6.1.3 of_start_pid(force_target,orient_calc)

Initiates force control mode. The robot instantly starts to hold the defined forces.

Parameters:

- `force_target`: This 6 element vector is the force-target to hold during force mode. The elements are the desired [`Fx Fy Fz Tx Ty Tz`] values. Note, that these values are Base coordinate-system values.
- `orient_calc`: This parameter is `True`, if orientation calculation is required during force-mode, and `False` otherwise.

The control system for the UR robot runs at a 125Hz speed. If orientation calculation is not needed (the `orient_calc` parameter is `False`), the position errors in `of_POS_ERROR` are calculated at the normal 125Hz speed. This is desired if the robot only moves, but does not need to rotate during force mode. This way more responsive movement is achieved.

If orientation calculation is needed (the `orient_calc` parameter is `True`), position and orientation differences are both calculated, but only at 25Hz speed. It is advised to design the application this way if rotations are not required during constant force mode. This restriction only applies to the script implementation of the force control library. URCaps force control modules do not have this restriction (and this parameter).

Note, that by executing this function, the `of_pid_thread()` will send movement commands to the robot.

> Only one controller should be active in the robot control environment. General movement commands cannot be executed while this controller is active. Position-controlled axes can be modified while holding constant force, but only OptoForce script functions defined in this section can be used.

5.6.1.4 of_stop_pid()

This function stops the `of_pid_thread()`, thus exits force mode. You might use general UR movement functions from this point.

5.6.1.5 Movement functions under force mode

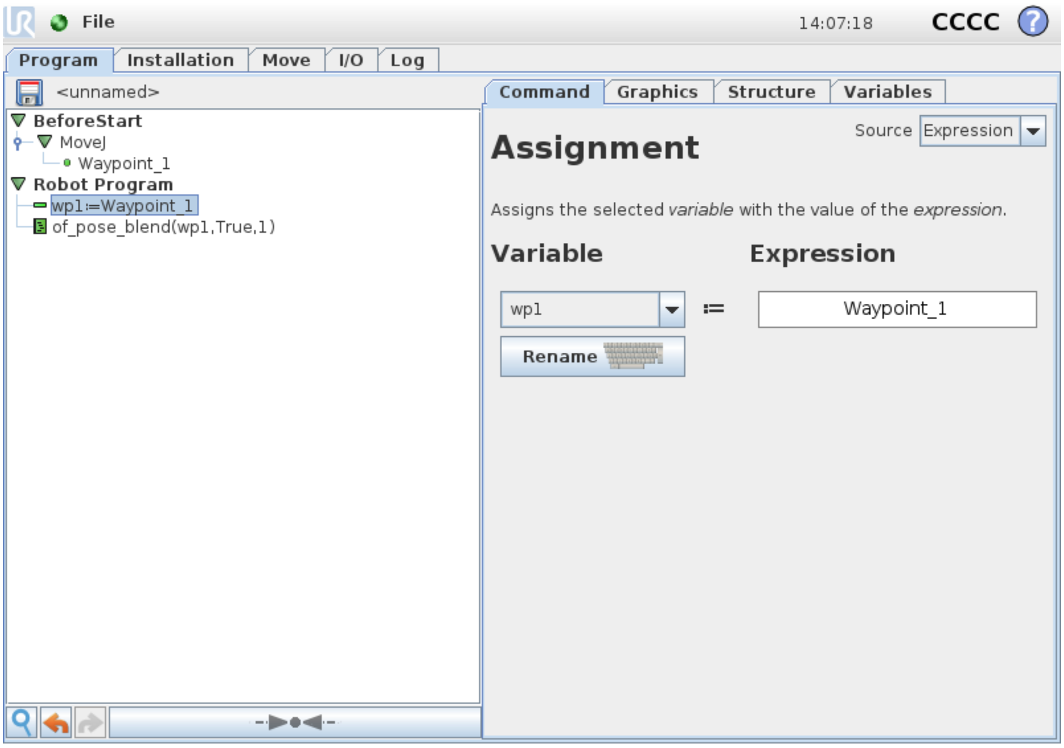
`of_pose_blend(pTo,open_loop_mode,resolution)`

This function will move the robot to another pose during force control mode. This function updates the `of_POS_TARGET` parameter of the PID controller.

Parameters:

- `pTo`: The pose to move to.

  Waypoints defined in the UR Polyscope cannot be inserted here directly, only variables can be supplied. To use a Polyscope waypoint here, copy the value to a simple variable before applying this function.

- `open_loop_mode`: This function implements two modes of operation, an open-loop and a closed-loop control mode.

  If this parameter is `False`, a closed-loop control is executed. The robot divides the trajectory to intermediate waypoints, and moves to them one-by-one. In this closed loop mode, the robot is guaranteed to execute the whole moving sequence, but due to the intermediate stops, the movement can twitch.

If this parameter is `True`, open-loop movement is initiated. The robot does not monitor whether it reached the defined waypoint. With this mode non-twitching movement is possible, but the robot is not guaranteed to reach its commanded target position. Correct movement along the trajectory can be set by changing the robot's movement speed and adjusting the last resolution parameter.

- `resolution`: In closed-loop mode, this parameter specifies how many intermediate points are calculated during the movement. By setting it to 0, the robot moves directly to the target position.

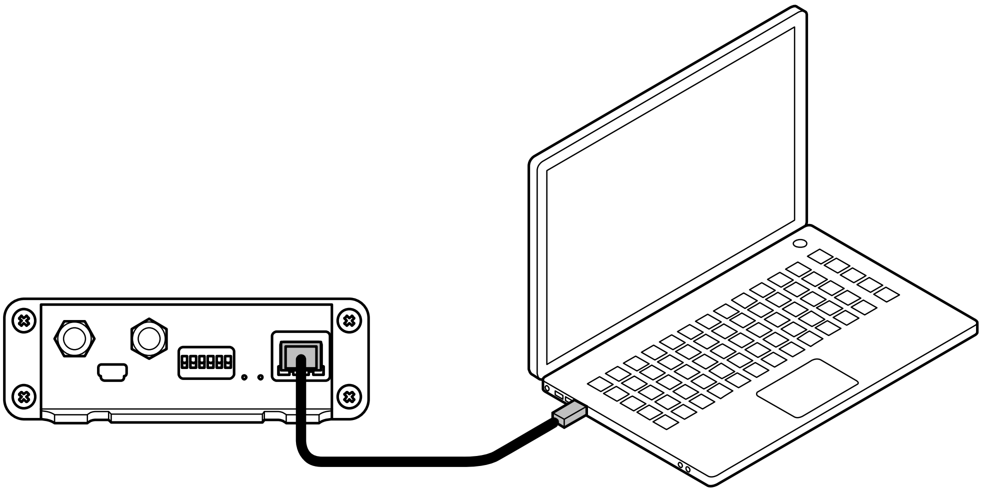
  In open-loop mode, this parameter specifies the desired time span of the movement in seconds.

  In some circumstances the desired result is not guaranteed since, due to the slow monitoring and control capabilities from inside the UR script environment, low-level robot control is not possible. The force control library in OptoForce URCaps has more advanced movement methods.

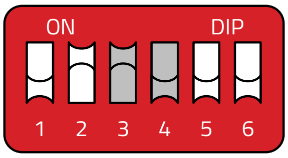
6 Changing the IP of the Sensor

To change the IP address of the sensor, connect your laptop or an external PC to the OptoForce sensor device.

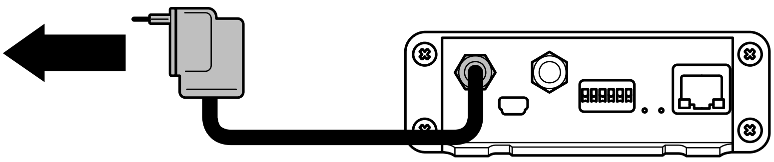
1. Make sure that the device is not powered. Connect the device and the computer with the provided Ethernet cable.



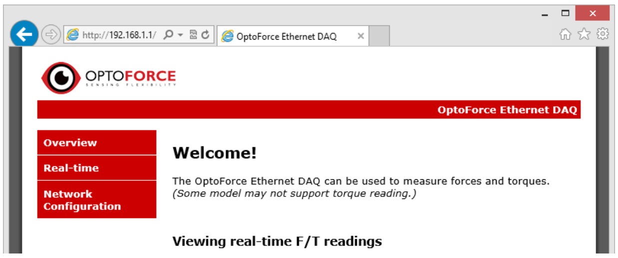
2. If your device is in the factory default settings, proceed to step 3. Otherwise, make sure to switch the DIP switch 3 to the ON position (up) and the DIP switch 4 to the OFF position (down).



3. Power the device from the provided power supply and wait 5 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 displays real-time sensor data.



5. If you need to change the IP address of the device, use the Network configuration menu.

6.1 Quick Troubleshooting

In case the webpage cannot be accessed, check the following:

- Check your computer's IP settings. The default setting on most operating system is the automatic IP mode. If you have other than the default setting, save your current setting and then set it to automatic mode or to a fixed IP: 192.168.1.2 (sub-net mask: 255.255.255.0). Then repeat the **Changing the IP of the Sensor** process.
- Close the web browser and reopen it (it might have cached a previous webpage).
- Make sure that no hardware/software firewall (or router) blocks the connection between the computer and the device.
- Your device might not be in the factory default state. Repeat **Changing the IP of the Sensor** from step 2.

The web access can be used with any web browser by entering the device IP address or the device hostname.

For example: **http://192.168.1.1** or **http://OPTODAQ/** if the factory default values are used.

The following welcome page opens is all setting are correct:

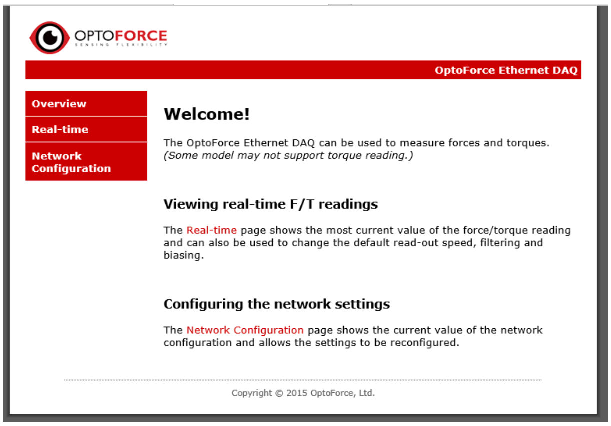


Figure 1: Welcome page

If you cannot access the welcome page by entering the device IP address, the network settings can be restored to the default values by switching the DIP switch 3 ON. The default values are IP: 192.168.1.1 (subnet mask: 255.255.255.0) with DHCP client off.

If the device cannot be reached by its hostname, clear the address caches in the web browser and OS. Enter `nbtstat -R` to the command prompt in Windows, to clear the hostname cache, close your current web browser, open a new web browser, and then try to access the web address above.

The **Network Configuration** menu on the top left can be used to check or change the network configuration of the device.

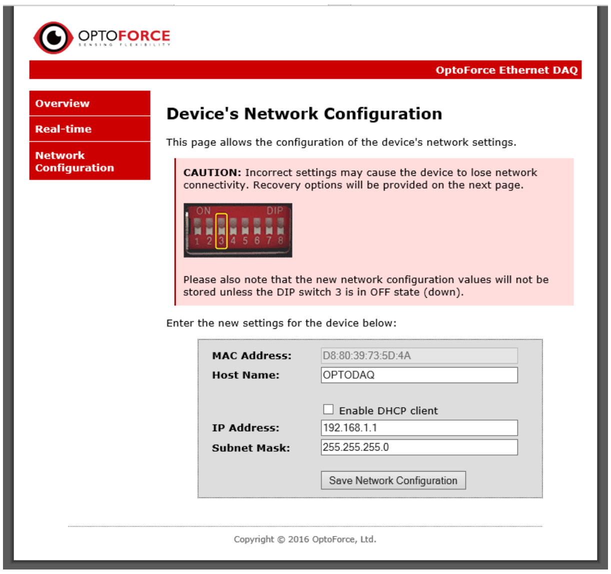


Figure 2: Network Configuration page

A. The MAC address is the word-wide unique ID that is fixed for the device.

B. The **Host name** can be maximum 16 character long.

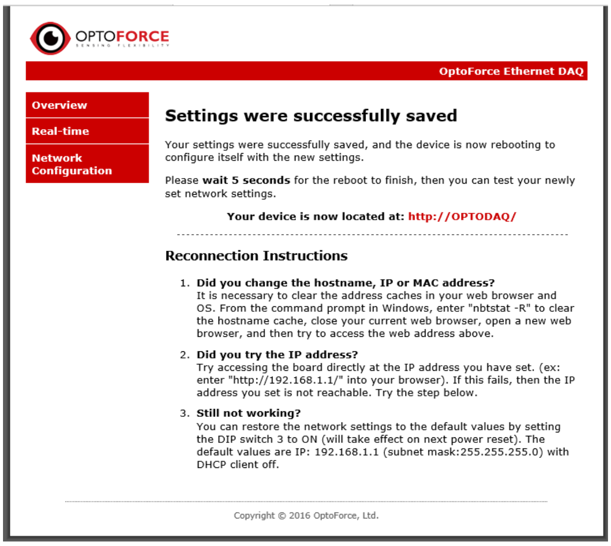
C. The **Enable DHCP client** checkbox can be used to enable or disable the automatic IP addressing.

> With DHCP client enabled, if the network that you connected to has no DHCP server, then the fixed 192.168.1.1 IP will be used instead for the device.

If DHCP client is disabled the static IP address can be set.

If you are using the device within a company network contact your IT department for the correct IP and subnet mask to be assigned.

After all parameters are set, click on the **Save Network Configuration** button to store the new values permanently. After the new parameter are stored the next page describes how to proceed:



For further information, refer to the *Installation and Operation Manual for Ethernet Converter* document provided on the USB pendrive.

7 Software Uninstallation

1. To uninstall (remove) the previously copied OptoForce UR program files, choose from the following options:

    a. Remove the files and folder, using the **Delete** option of the Tech Pendant during file operation (for example, Load Program, Save Program)

    b. Copy the `uninstall.sh` file from the USB pendrive to a new pendrive, rename it to `urmagic_optoforce_uninstall.sh` and plug it into the Tech Pendant. The file creates a backup copy to the pendrive, and then it permanently deletes the `OptoForce_UR_Programs` folder.

2. Uninstall the URCaps installation.

    a. Go to the Main screen of the PolyScope.

    b. Click **Setup Robot**.

    c. Click on **URCaps Setup** and locate the `OptoForce Force Torque Sensor` in the list of active URCaps.

    d. Click on the **-** sign at the bottom to uninstall it.

    e. Restart the robot.