



Gecko Gripper Universal Robots Appendix



Table of Contents

Table of Contents.....	2
1. Mounting and Wiring the Gripper on a Universal Robot	3
2. Using the URCap Plugin.....	3
2.1 URcap Plugin Installation.....	4
2.2 Programming URcap.....	7
2.3 URcap Feedback Variables	10
2.4 Error Handling.....	10
2.5 Adjusting Safety Limits to Pick Objects.....	12
3 Sample UR Programs.....	13
3.1 UR Program with Gecko Gripper URcap (gg-pick-eth)	13
3.2 UR Program with Gecko Gripper URcap Feedback Variables (gg-urcap-force).....	14
3.3 UR Program with Gecko Gripper I/O Control (gg-pick-io)	15

Find the most up to date user manual and additional documentation on our website:

<https://onrobot.com/products/gecko-gripper/>

1. Mounting and Wiring the Gripper on a Universal Robot

The Gecko Gripper is designed to be compatible with Universal Robot models. The gripper can be mounted directly to a Universal robot arm and does not require an adaptor plate.



Figure 1: The Gecko Gripper mounts directly to the Universal Robot arm.

Additionally, the gripper can be powered by directly connecting the power cable to the robot's integrated 24V DC power supply. The Ethernet cable can be connected directly to the robot's controller or through a network.

2. Using the URCap Plugin

The Gecko Gripper teach pendant GUI allows you to enter or adjust the gripper state using the simple touch interface and to display gripper data directly on the robot teach pendant.

2.1 URCap Plugin Installation

Plug in the OnRobot USB flash drive to the robot. The user will install it by navigating to Setup Robot → URCaps Setup.

(This flash drive is included with the Gripper in delivery.)

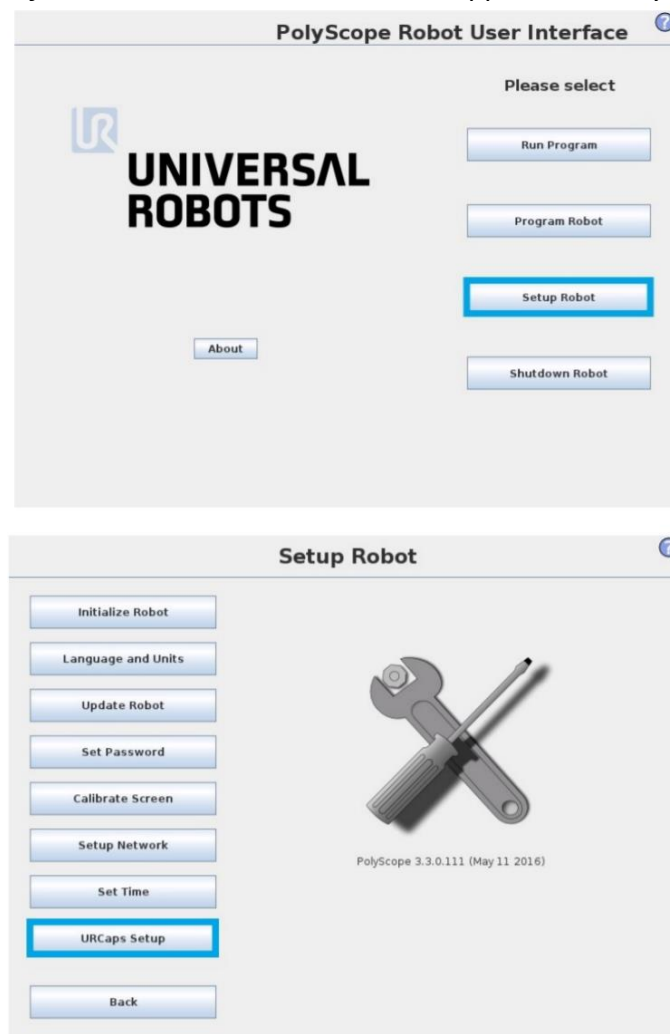


Figure 2: How to install the gecko gripper URCaps to the Universal Robot.

Step 2: Navigate to the “Installation tab.” Here, you may enter the Gecko Gripper’s IP address and port to enable communication to the Gecko Gripper. The Gecko Gripper’s default IP address is 192.168.0.170 and the default port number is 30000. Click “Connect” to connect to the gripper.

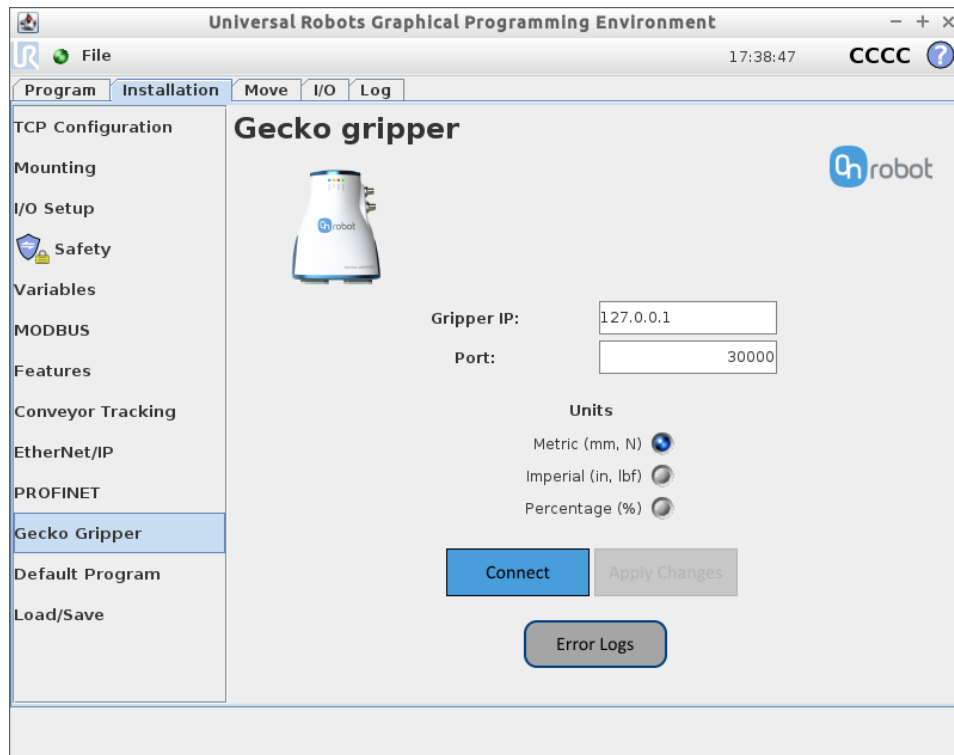


Figure 3: Gecko Gripper installation screen in the Universal Robots graphical programming environment.

Step 3: You may also change the desired units used by the Gecko Gripper program, selecting between Metric (*default units*), Imperial, or Percentage. Once you have selected your desired units, click “Apply Changes” to save these settings.

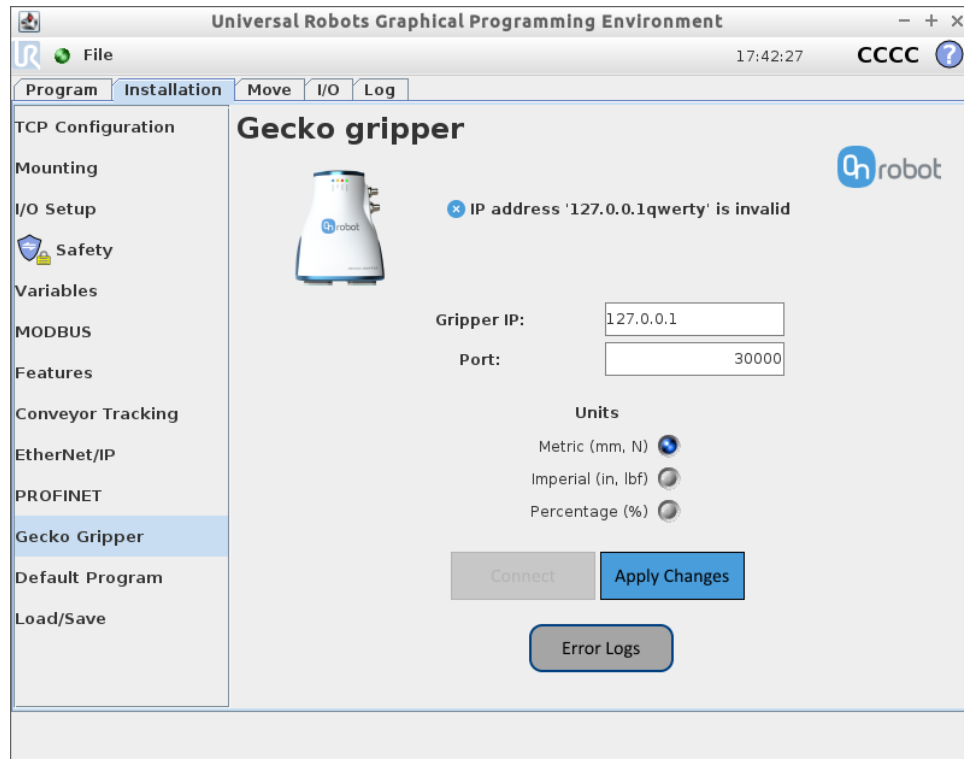


Figure 4: Units can be changed under the Installation tab by toggling between the Units radio buttons. Select Apply Changes to save.

Step 4: Error logs can be retrieved by clicking the “Error Logs” button on the Installation tab, then selecting “Load Logs.”
For more information on the error handling, see Section 2.6.

You are now ready to Verify Gripper Functionality and Configure the Gripper from the robot teach pendant.

2.2 Programming URCap

Step 1: The Gecko Gripper program node can be found by first clicking on the main “Program” tab, then clicking on the “Structure” tab, and finally selecting the “URCaps” tab within the “Program Structure Editor” window.

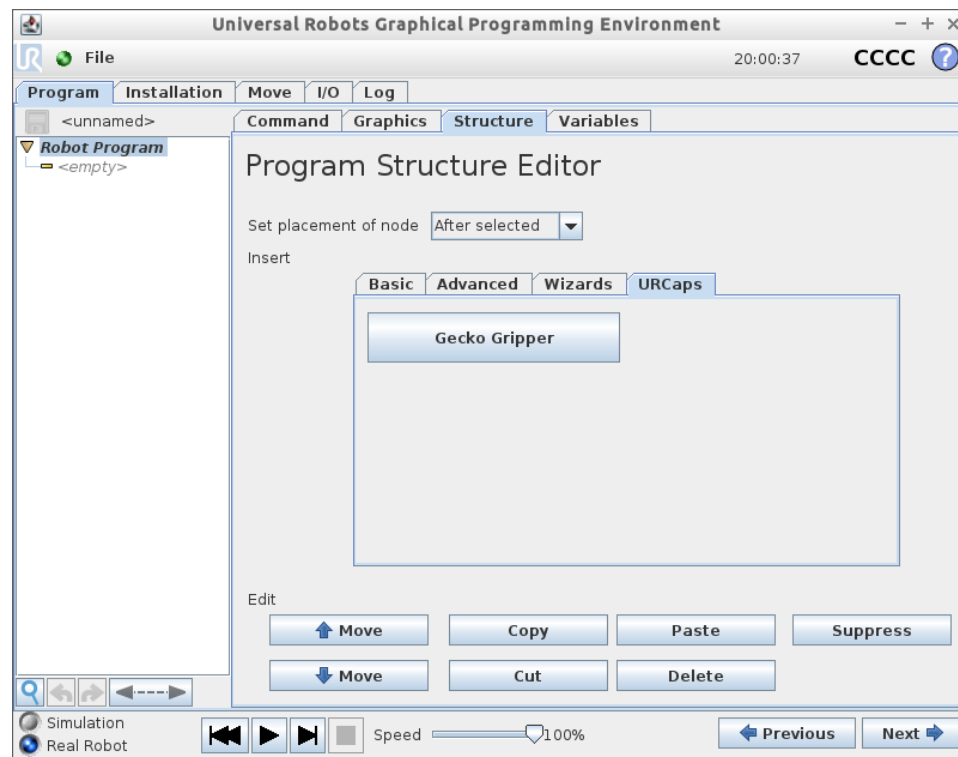


Figure 5: Location of the Gecko Gripper program node within the UR environment.

Step 2: Navigate to the main UR Cap screen under the “Program” tab → “Command” tab in the UR GUI. If the gripper power and Ethernet cables are properly connected to the UR, then the URCap will display the message “Gecko gripper connected!” (Figure 6). If there is a problem with either of these connections, the URCap will display the message “Failed to connect to the gecko gripper”.

The main “Training Mode” screen from which the operator specifies several gripper parameters that collectively define a single gripper state.

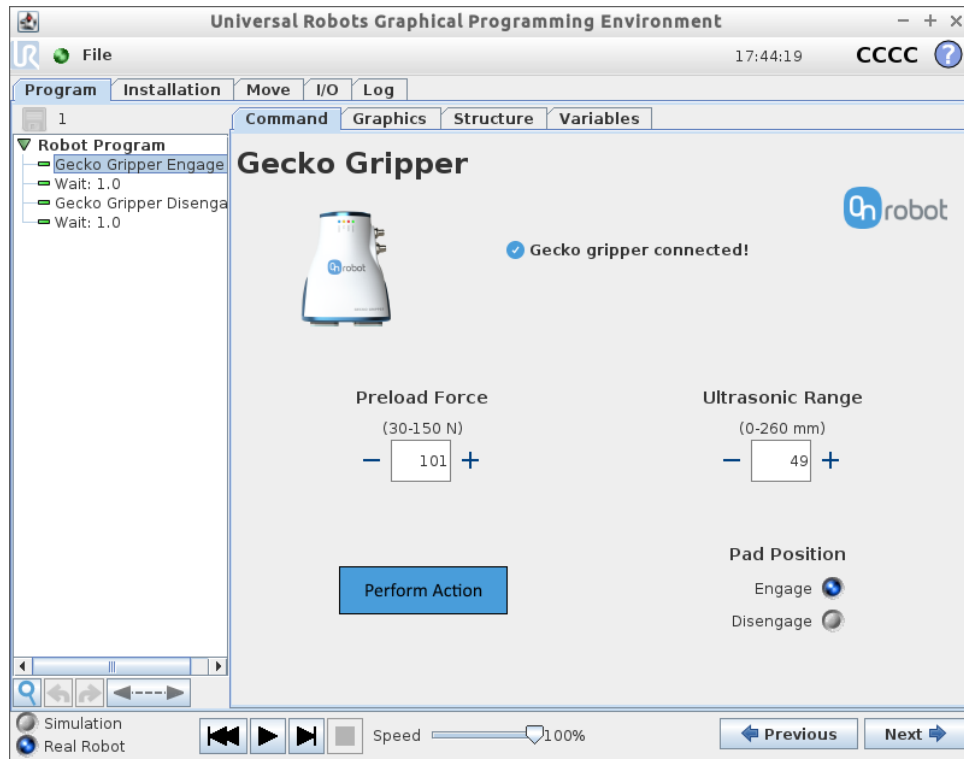


Figure 6: Gecko Gripper URCap screen

- Step 3:** Set the desired Preload Force.
 This setting modifies at what force level the gripper notifies the robot that it has reached a certain load. For example, in picking a large piece of glass where 100 N preload force is required, when 100N is reached in I/O mode, pin 5 is set HIGH; in Ethernet mode, packet index 9 is set from 0 to 1.
For more information on selecting an appropriate preload force for your task and material, see Gecko Gripper User Guide, Section 9.4.

NOTE: The gecko grippers pre-load sensing range is 40 to 140N, it **CANNOT sense below 40N**

- Step 4:** Set Ultrasonic Range.
 Like setting the Preload Force, this setting notifies the robot at which *range* the designated Preload Force is reached. This feature is useful for picking flat objects out of a pile as it allows the robot programmer to run the robot at maximum speed until the gripper detects that it is approaching a pickup point. An example of this use case is described in the Gecko Gripper User Guide, Section 8.1, Step 2.

Default ultrasonic range trip-wire setpoint is 125.0mm.
The ultrasonic sensor has a 500 msec delay in response, please account for that in robot programming.

- Step 4: Select Pad Position.
To test basic gripper functionality, the user can try to perform an action with each pad position (“Engage” and “Disengage”). When picking up a part, the user should select “Engage”, and when releasing a part, the user should select “Disengage”.
Default Pad Position is “Engage.”
- Step 5: To view real-time gripper data, including the Parts Presence, Wear, Preload Force, and Pad Position, navigate to the “Variables” tab when the robot program is running.
- Step 6: When you have completed setting up the new state, select “Perform Action” to set the gripper to the state that matches the selected parameters. When the gripper has completed the action, the URCap will display the message “Gripper action completed successfully!” (Figure 7). These parameters are written to the gripper’s memory. If the gripper is run in I/O configuration, it will reference those parameters to set the gripper’s state. If the gripper used in Ethernet mode, it will reference those parameters as initial state, but they can be dynamically modified.

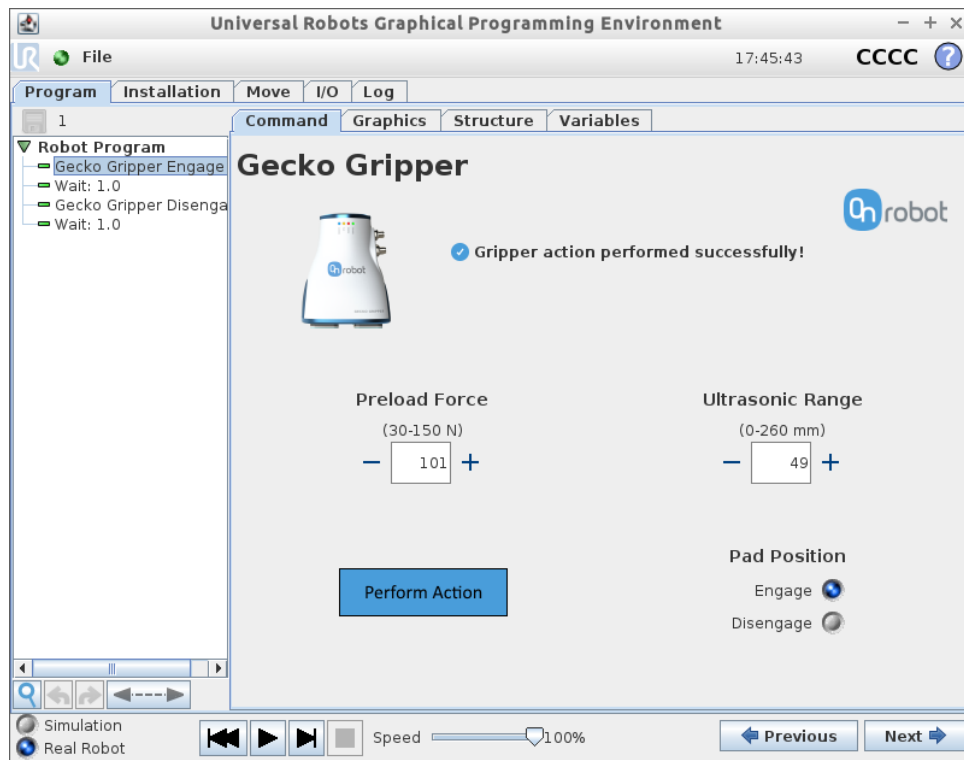


Figure 7: Gecko Gripper URCap has completed the gripper action.

2.3 URcap Feedback Variables

The robot programmer can use the Gecko Gripper URcap feedback variables to access real-time gripper sensor data. The units for these variables is metric by default, but can be changed in the installation tab (Section 2.1, Figure 4). The sample program gg-urcap-force.urp is an example of how to use these feedback variables in a UR robot program (see Section 3.2).

Feedback Variable	Description
gg_force	Normal force exerted by gripper on the part for pickup
gg_force_deriv	Force derivative
gg_ultrasonic	Distance of gripper to part as measured by ultrasonic range sensor
gg_parts_pres	True if gripper has a part
gg_wear	True if gripper has dropped a part

Figure 8: UR Gecko Gripper variables. These variables will update automatically during UR program execution.

2.4 Error Handling

The Gecko Gripper GUI saves detailed information about unexpected events or errors during program execution. These error logs are can be retrieved by navigating to the Gecko Gripper Installation tab and clicking “Error Logs”.

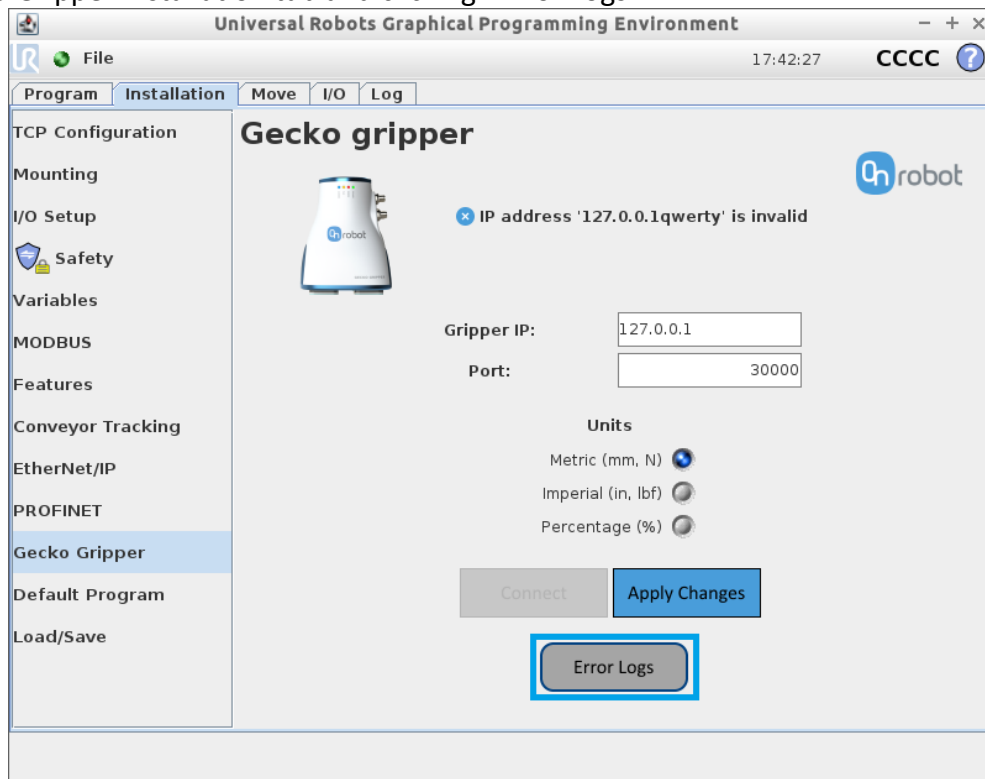


Figure 9: Gecko Gripper URcap has completed the gripper action.

To load the error logs and details, click “Load Logs”. To clear all logs on the screen, click “Clear All.”

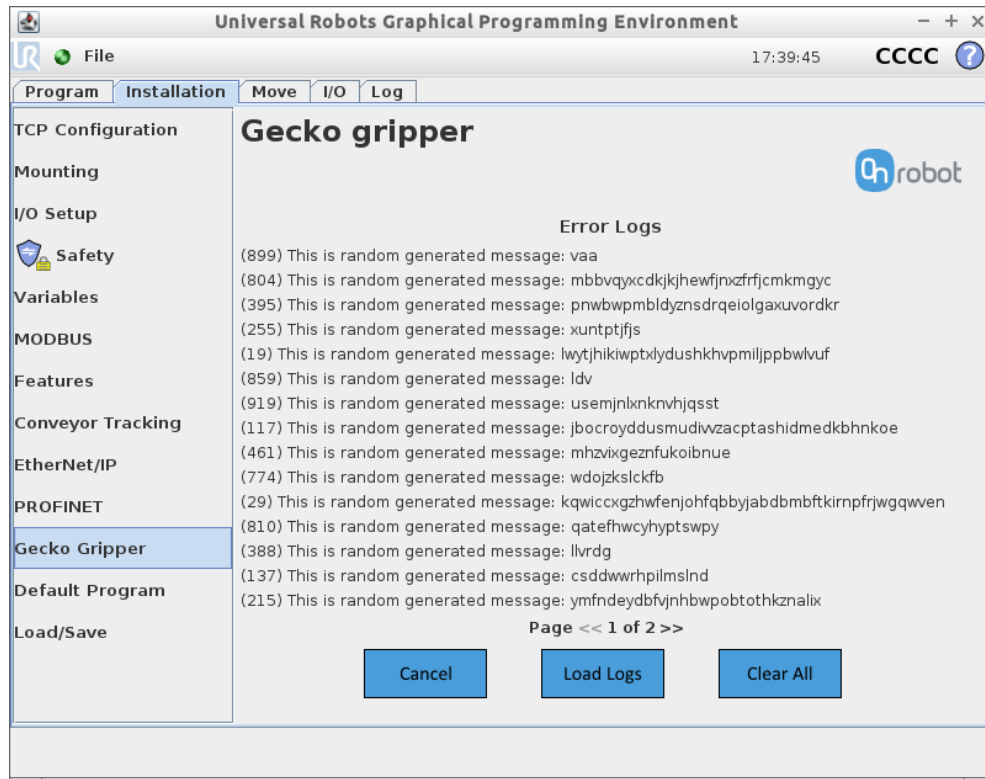
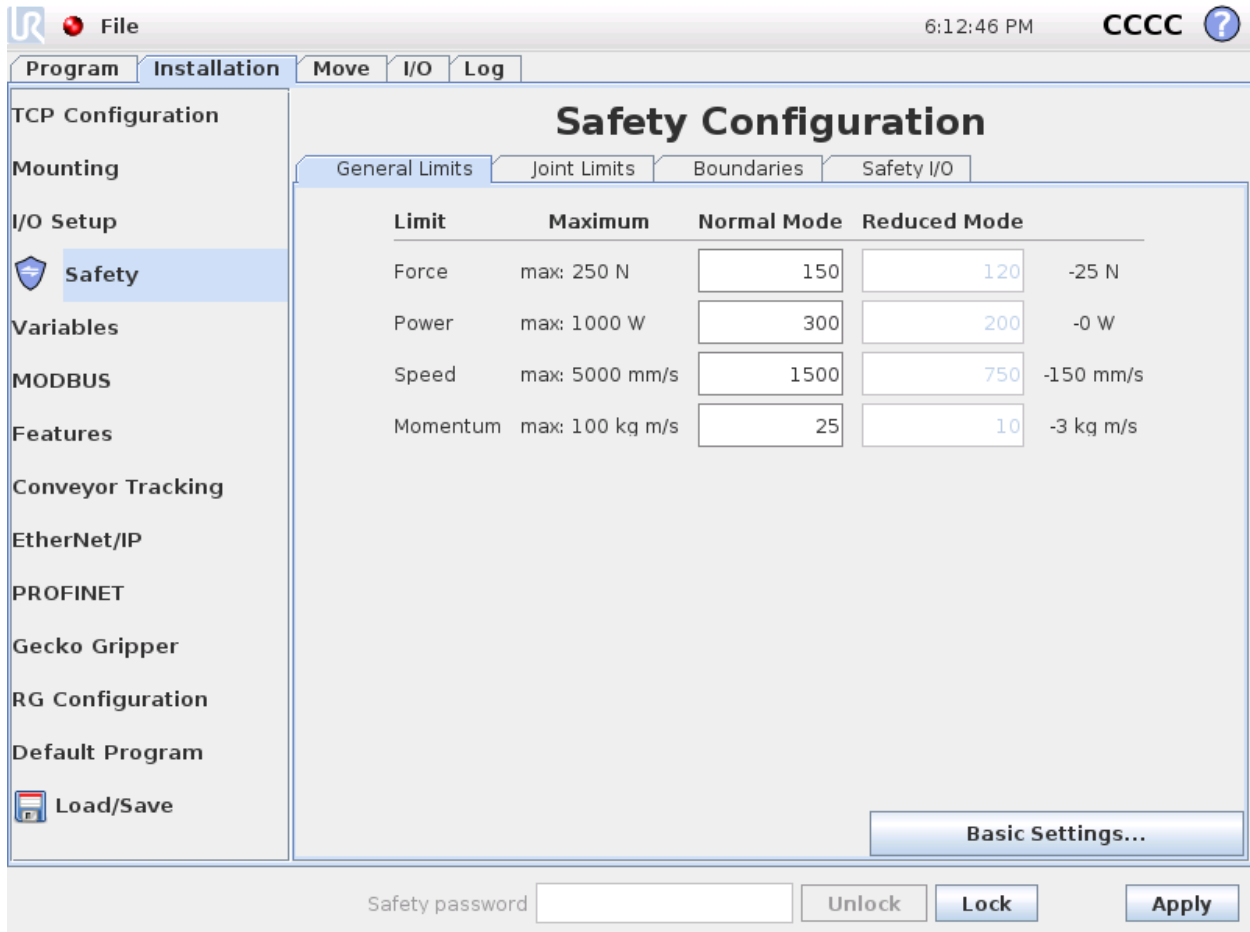


Figure 10: Error codes and error logs.

2.5 Adjusting Safety Limits to Pick Objects

Based on the pick speed of the robot and the required pre-load force, it may be necessary to adjust the UR's default 150N force safety value higher to pick properly. It is not likely (but not guaranteed) that this value exceeds 200N. This can be done via:

Installation → Safety → Advanced Settings (see Figure 11).



The screenshot shows the UR Safety Configuration interface. The left sidebar contains various configuration options, with 'Safety' highlighted. The main area is titled 'Safety Configuration' and has four tabs: 'General Limits', 'Joint Limits', 'Boundaries', and 'Safety I/O'. The 'General Limits' tab is active, displaying a table of safety parameters. The table has columns for 'Limit', 'Maximum', 'Normal Mode', 'Reduced Mode', and an additional column for a secondary value. The values for 'Normal Mode' and 'Reduced Mode' are currently set to 150 and 120 for Force, 300 and 200 for Power, 1500 and 750 for Speed, and 25 and 10 for Momentum. At the bottom of the screen, there is a 'Safety password' field, 'Unlock', 'Lock', and 'Apply' buttons, along with a 'Basic Settings...' button.

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

Figure 11: Safety configuration screen

3 Sample UR Programs

The Gecko Gripper packaging box comes with an OnRobot USB that includes two UR sample programs in the UR/Sample Programs directory. The gg-pick-urcap.urp program uses the URcap/Ethernet to control the gripper, while the gg-pick-io.urp program uses robot I/O to control the gripper.

3.1 UR Program with Gecko Gripper URcap (gg-pick-eth)

The sample program that demonstrates how to use the URcap is called gg-pick-eth.urp. It is a simple pick-and-place with gecko gripper URcap. It's recommended to use subroutines for Engage and Disengage commands, to simplify the program tree.

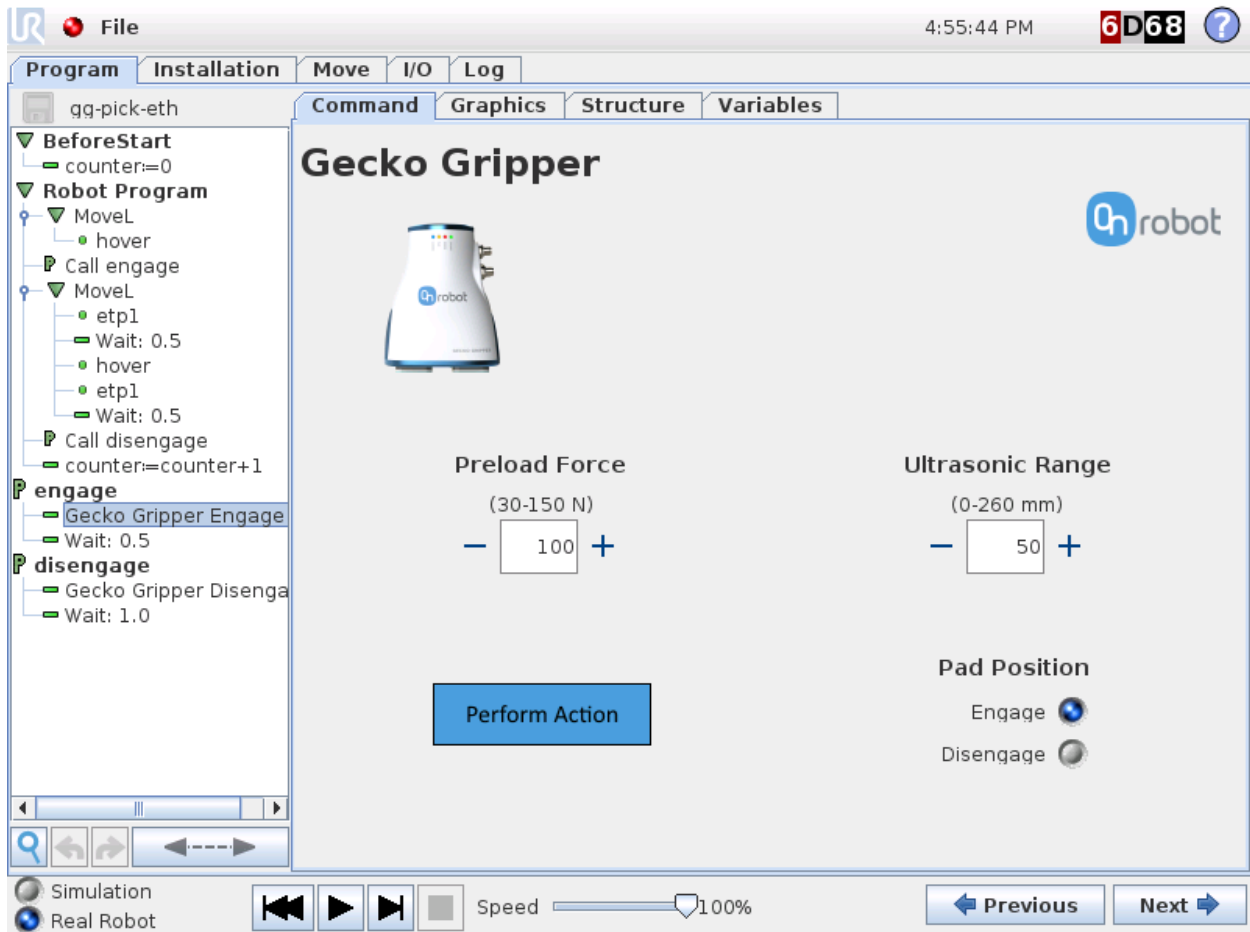


Figure 12: Screenshot of Gecko Gripper URcap in gg-pick-eth

In this program, when the Gecko Gripper does an object pick, the **engage** subroutine is called, and when the object is released the **disengage** subroutine is called. The robot picks up the object located at the fixed waypoint *etp1*.

3.2 UR Program with Gecko Gripper URCap Feedback Variables (gg-urcap-force)

This sample program shows how to use the URCap feedback variables. In this example, the preload force sensor variable (`gg_force`) is used so the robot can pick objects out of a pile.

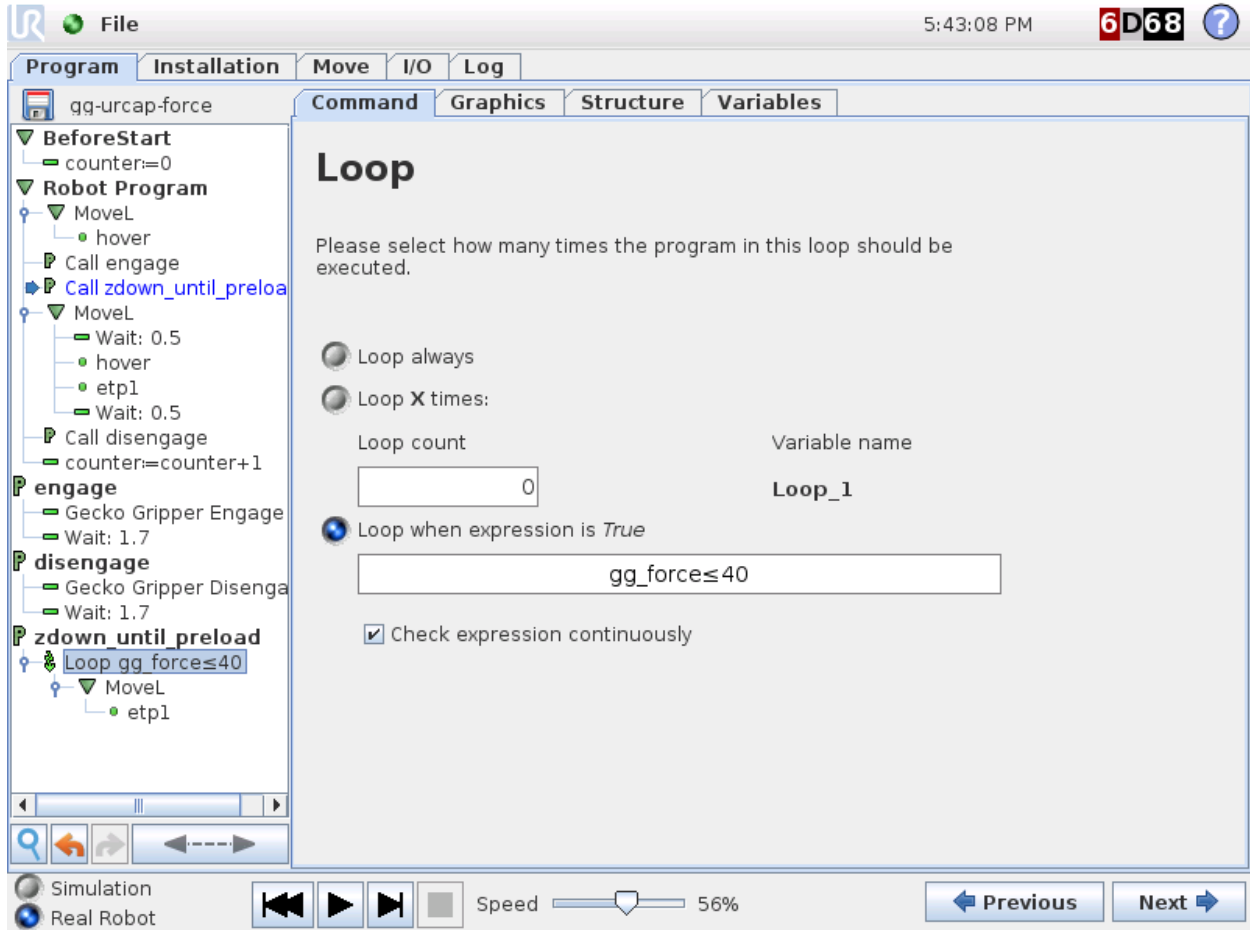


Figure 13: Screenshot of gg-urcap-force program tree

In this program, the robot first moves to the “hover” point above the pile, and then calls the subroutine `zdown_until_preload`. In this subroutine, there is a conditional while loop that moves the robot to the bottom of the pile, which is denoted as `etp1`. This robot motion gets interrupted when the preload force variable (`gg_force`) exceeds 40N. This allows the gecko gripper to pick objects out of a pile without having to create a new waypoint for each successive piece.

NOTE: The MoveL in the `zdown_until_preload` subroutine has a tool speed of 200 mm/s. If the user wishes to move the robot at a faster speed, the gripper’s preload force sensor is not guaranteed to respond quickly enough before the robot hits the part, which could result in a robot stopping due to the UR’s safety setting.

3.3 UR Program with Gecko Gripper I/O Control (gg-pick-io)

The sample program that demonstrates how to control the gripper from robot I/O is called gg-pick-io.urp. It's recommended to use subroutines for Engage and Disengage commands, to simplify the program tree.

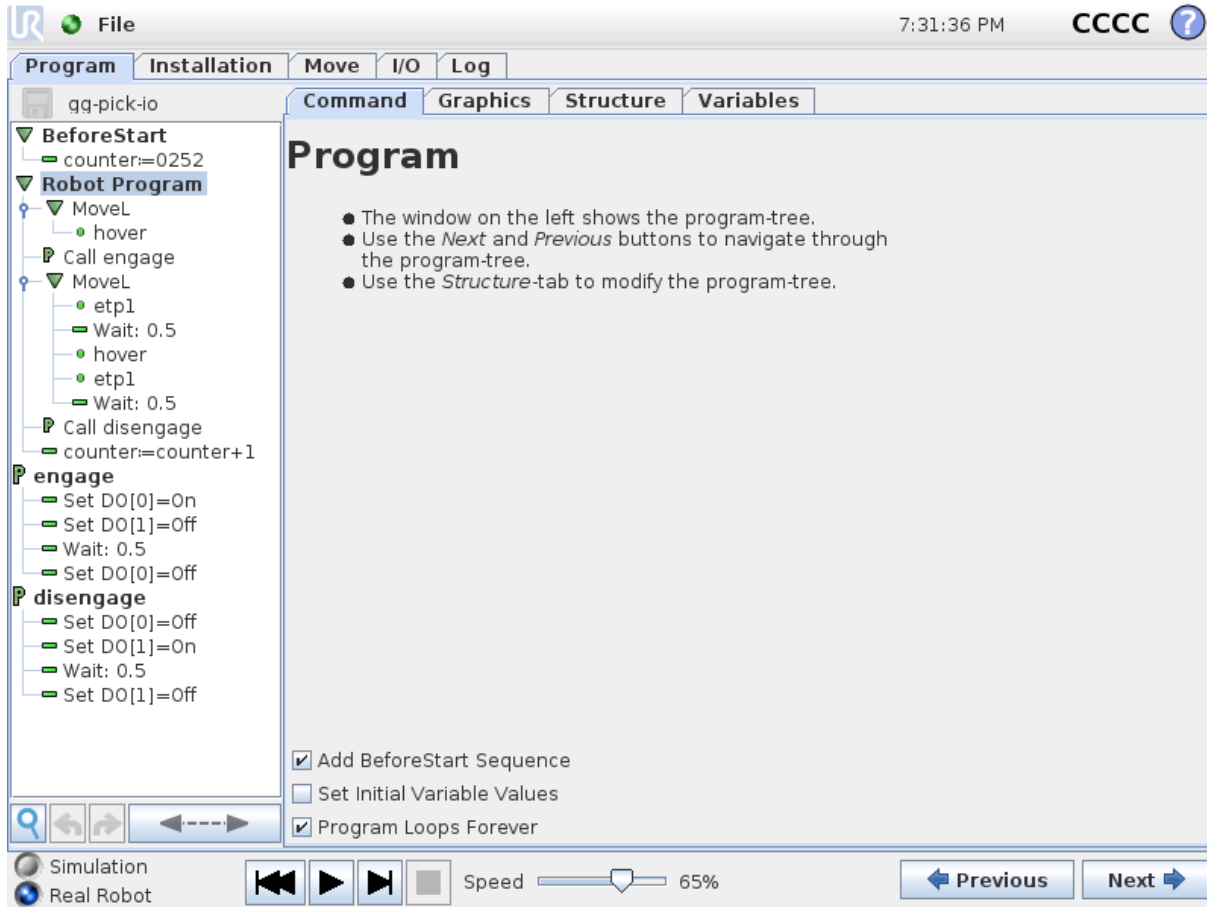


Figure 14: Screenshot of gg-pick-io program tree

In this program, when the Gecko Gripper does an object pick, the **engage** subroutine is called, and when the object is released the **disengage** subroutine is called. The white engage wire has been wired to robot digital output 0, and the brown disengage wire has been wired to robot digital output 1 (see *Gecko Gripper User Manual, Section 7.1, Figure 37*). The gripper outputs (ultrasonic, parts presence, preload), pad service) can also be wired to robot digital inputs, to create a more robust program.